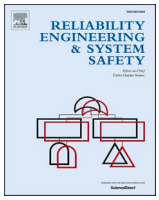
Reliability Engineering and System Safety 236 (2023) 109290

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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



Joint optimization of maintenance and quality inspection for manufacturing networks based on deep reinforcement learning

Zhenggeng Ye [a], Zhiqiang Cai [b,*], Hui Yang [c], Shubin Si [b], Fuli Zhou [d]

[a] Department of Industrial Engineering, School of Management, Zhengzhou University, Zhengzhou 450001, China
[b] Department of Industrial Engineering & Ministry of Industry and Information Technology Key Laboratory of Industrial Engineering and Intelligent Manufacturing, Northwestern Polytechnical University, Xi'an 710072, China
[c] Harold and Inge Marcus Department of Industrial and Manufacturing Engineering, Pennsylvania State University, University Park, PA 16802-1401, USA
[d] Department of Management Science and Engineering, School of Economics and Management, Zhengzhou University of Light Industry, Zhengzhou 450002, China

ARTICLE INFO

Keywords:
Manufacturing network
Reliability model
Maintenance
Quality inspection
Optimization
Deep reinforcement learning

ABSTRACT

Most existing studies on joint optimization of manufacturing systems (MS) focus on small-scale systems with simple structures, such as the single-machine, simple serial, or parallel MS. Simultaneously, traditional algorithms utilized in small-scale MS always show an insufficiency in solving large-scale dynamic MS with complex structures, such as manufacturing networks. Therefore, considering the effectiveness of reinforcement learning on the infinite-horizon Markov Decision Process (MDP), this paper presents a joint optimization problem of preventive maintenance and work-in-process quality inspection for manufacturing networks with reliability-quality interactions. First, dynamic reliability and quality models are proposed at the machine level to cope with complex interactions in manufacturing networks. Second, based on the MDP-based optimization model, the proposed Deep Deterministic Policy Gradient (DDPG) algorithm realizes the optimal reliability-quality joint control in manufacturing networks. Besides, it also offers a novel mixed action space containing discrete maintenance and continuous quality inspection, which could satisfy the action diversity in actual production. At last, training and experiments imply our algorithm is more adaptable to diverse manufacturing scenarios than traditional ones. Also, it is proved that more-frequent state observations for learning cannot help the constructed reinforcement learning model get a better control policy because of the information redundancy.

1. Introduction

Mass customization provides higher production flexibility but increases MS complexity [1]. In this context, high-technology manufacturing equipment with flexible capabilities is bred, and process routes of final products are greatly enriched, making the MS show the characteristic of complex networks when considering machines as nodes and work-in-process (WIP) flows as edges. The enhancement of both machine flexibility and structural complexity strengthens the nonlinearity of MS. This will increase the difficulty in operational control of networked MS and weaken the profit brought by flexible machines.

Operational control of MS refers to optimizing system performance through production management methods, where maintenance of machines and quality inspection of WIPs have been two significant actions [2,3]. Integrated optimization through joint control of production scheduling, WIP quality, machine reliability, etc., has been a prior choice in improving MS performance. Different integration forms have been tried in current studies, such as single preventive maintenance (PM), integration of production scheduling and maintenance, or the integration of production scheduling, maintenance, and quality inspection, as shown in Table 1. It can be concluded that current studies of MS joint optimization mainly focus on simulation-based methods. Besides, traditional dynamic programming, integer programming, and heuristic algorithms are also important methods in solving this topic. These traditional optimization methods have great potential for optimizing small-scale systems, such as the single-machine, simple serial, or cellular MS, as shown in Table 1. However, they show insufficient in optimizing large-scale MS with complex system structures. Although the heuristic algorithm can effectively optimize multi-stage serial or parallel MS, such as the genetic algorithm [4,5], their effectiveness in large-scale MS with complex system structures is not validated.

Diversified process routes make MS show large-scale and structural

* Corresponding author.
E-mail address: caizhiqiang@nwpu.edu.cn (Z. Cai).

https://doi.org/10.1016/j.ress.2023.109290
Received 20 August 2022; Received in revised form 24 March 2023; Accepted 2 April 2023
Available online 6 April 2023

Acronyms

MS	manufacturing system
WIP	work-in-process
RL	reinforcement learning
DRL	deep reinforcement learning
MDP	markov decision process
DDPG	deep deterministic policy gradient algorithm
DQN	deep Q-Network algorithm
DDQN	deep deterministic policy gradient algorithm
CM	corrective maintenance
PM	preventive maintenance
NDM	neighboring-downstream machines
NUM	neighboring-upstream machines
NHPP	non-homogeneous Poisson process
SD	standard deviations
CV	coefficient of variation
FFT	fast fourier transform

Notations

$G(V, E)$	directed acyclic graph of manufacturing network
V	set of nodes representing manufacturing machines
W^+	matrix of the forward weight w^+_{ij} of directed edges, representing the probability that each WIP in machine i will flow into machine j
W^-	matrix of the inverse weight w^-_{ij} of directed edges, representing the probability that each WIP in machine j comes from machine i
D_i	set of neighboring-downstream machines of the machine i
U_j	set of neighboring-upstream machines of the machine j
P_{rmi}	maximum production velocity of the machine i
P_{rai}	actual production velocity of the machine i
$r_b(t)$	failure rate of a machine with dynamic production velocity
t_r	relative operating time of a machine under dynamic production velocity
$r(t)$	the actual failure rate of a machine
$N(t)$	number of low-quality feedstocks processed by a machine during $[0, t)$
Δr	failure-rate increment caused by one processed low-quality feedstock in a machine
t_{cm}	time of one CM in a machine
t_{pm}	time of one PM in a machine
c_{cm}	costs per unit time of CM in a machine
c_{pm}	costs per unit time of PM in a machine
$N_{cm}(t)$	frequencies of CM during $[0, t)$ in a machine
$N_{pm}(t)$	frequencies of PM during $[0, t)$ in a machine
$M(t)$	number of defectives produced by a machine during $[0, t)$
$\lambda(t)$	intensity function of defectives produced by a machine at time t
q	defective proportion obtained by the inspection activity in a machine during $[t, t+\Delta t)$
c_{ins}	inspection cost per WIP
c_I	total inspection cost in a machine during $[t, t+\Delta t)$
v_{ai}	average value increment brought by processing one WIP in machine i
v_{net_i}	net value increment from processing WIPs during $[0, t)$ in machine i
S^k	state of a manufacturing network at time $t = k\Delta t$
Δt	period length of one quality statistic
r_k	step reward of a manufacturing network during $[t, t+\Delta t)$
G_k	long-term reward of the manufacturing network during an epoch $[k\Delta t, k'\Delta t)$
A^k	actions for a manufacturing network at time $t = k\Delta t$
S^k_a	vector of sampling fractions in quality inspection actions for all machines
P^k_m	vector of PM actions for all machines
c_d	criterion for executing PM

Table 1
Integration forms in joint optimization of MS performance.

System structure	Optimizing actions	Optimizing methods
A single machine	Maintenance	Iterative optimization [6].
	Production & Maintenance	Integer programming [7], Gradient-based algorithm [8]
	Production & Maintenance & 100% Inspection	Simulation [9]
	Production & Maintenance & Sampling inspection	Simulation [10], Genetic algorithm [11]
Serial MS	Maintenance	Simulation [12,13], Dynamic search algorithm [14]
	Production & Maintenance	Genetic algorithm [4]
	Production & Maintenance & Sampling inspection	Simulation [15]
Parallel MS	Production & Maintenance	Genetic algorithm [5]
Five-machines cellular MS	Production & Maintenance	Nonlinear programming [16]

complexity characteristics. Combining interactions between WIP quality and machine reliability, the research on manufacturing networks is becoming a challenging issue. Many researchers have tried to solve the interaction [13,17] or large-scale problems [18–20]. However, they only independently studied one aspect of interaction and structural complexity, which cannot fully grasp the feature of manufacturing networks. As stated above, joint optimization of manufacturing networks with structural complexity and interacting behaviors is an intractable issue, especially when we seek to solve it by traditional methods [21]. Traditional methods are highly effective for a specific manufacturing scenario but may be ineffective in controlling or optimizing dynamic and diverse manufacturing scenarios. However, artificial intelligence (AI) development gives hope to solving this real-time control and optimization issue effectively. And AI-based methods have been widely used in related fields, such as risk assessment [22], intelligent maintenance [23], and quality control [24]. RL has demonstrated attractive achievements in various control tasks of highly nonlinear, high-dimensional, and dynamical systems [25]. Given its effectiveness, many RL-based control topics have been explored to optimize concerned dynamic systems, as shown in Table 2.

In research on MS control, the structural complexity does not get more attention, and discrete maintenance or inspection activities have been a mainstream selection in action settings. In actual production, however, the quality inspection has a continuous action space that can sample WIP by any fraction in [0,1]. Therefore, simplifying system states to onefold continuous or discrete type also deviates from the actual production status. By utilizing advanced RL algorithms, the study in control of non-MS provides a good example for large-scale problems with state-action diversity [23,35], and inspection activities are considered in some action settings [32,37]. However, this discrete maintenance inspection is very different from the quality inspection activities in MS. Furthermore, the study of non-MS in Table 2 also did not consider interacting behaviors between components. Therefore, it needs to be further explored whether or not those advanced algorithms are competent for large-scale manufacturing networks with reliability-quality interactions.

This paper presents new research on maintenance-inspection joint

Table 2
RL-based control in MS and non-MS.

System types	System structure	State types	Actions and types	Algorithms
MS	A single machine MS [26]	Discrete	Production/ Maintenance/ Discrete	Q-learning
	A manufacturing facility with a buffer [27]	Discrete	Production/ Maintenance/ Discrete	Q-learning
	2-serial-machines MS [28]	Discrete	Maintenance/ Discrete	Multi-agent Q-learning
	6-machine-5-buffer serial production line [29]	Continuous	Maintenance/ Discrete	DDQN
	7-stage manufacturing line [30]	Discrete	Maintenance/ Discrete	DQN
Non-MS	A single-unit system [31]	Discrete	Replacing/ Discrete	DQN
	A pump system in nuclear power plants [32]	Discrete	Repair/ Inspection/ Discrete	Iteration algorithm
	3-component system [33]	Continuous	Maintenance/ Replacing/ Discrete	Q-learning
	A batch of aircrafts [34]	Continuous	Maintenance/ Discrete	Q-learning
	13-component system [35]	Mixed integer-discrete-continuous	Maintenance/ Discrete	Actor-critic
	K-component system [36]	Mixed discrete-continuous	Replacing/ Discrete	DQN
	10-nodes multiple routes network [37]	Discrete	Maintenance/ Maintenance inspection/ Discrete	Multi-agent actor-critic
	14-components parallel-serial system [23]	Mixed integer-discrete-continuous	Performance rate after maintenance/ Continuous	DDPG
	15-components multi-state serial-parallel systems [38]	Discrete	Maintenance/ Replacing/ Discrete	Multi-agent DDQN

optimization of manufacturing networks by deep reinforcement learning (DRL). Through the RL-based reliability-quality joint control, the proposed method optimized maintenance and quality inspection in a large-scale manufacturing network with reliability-quality interacting behaviors. A novel model depicting dynamic machine reliability and processing quality is offered by considering reliability-quality interacting behaviors. Further, a dynamic operating environment of manufacturing networks is constructed, where network topology, machine reliability, product quality, maintenance activities, quality inspection activities, and dynamic production velocity are modeled. Further, to realize the reliability-quality joint control of dynamical manufacturing networks, a deep neural network-driven RL model is constructed based on the DDPG algorithm, where a mixed action space of discrete maintenance and continuous quality inspection is modeled. The designed training experiment and comparative analysis prove the effectiveness and progressiveness of the proposed methods.

The rest of this paper is organized as follows. Section 2 details the studied problem and research framework. Section 3 provides detailed mathematical models of the dynamic manufacturing network environment. Section 4 proposes the joint optimization model of maintenance and quality inspection based on MDP. Section 5 details the DDPG algorithm in the DRL agent. Section 6 demonstrates the efficacy of the proposed models and presents the experimental results. Section 7 concludes this research.

2. Problem description

WIPs will have multidirectional flows in a flexible manufacturing environment due to the diversity of process routes, leading to the networked structure of MS. As shown in Fig. 1, nodes represent machines, and the edges represent flows of WIPs. In this context, for a given machine, its upstream machine refers to all machines before it in concerned process routes, and its downstream machine refers to all ones behind it. In this MS, the flow of WIPs brings feedstocks to each machine in different stages, simultaneously leading to interacting behaviors between machine reliability and WIP quality. In detail, low-quality feedstocks will raise the failure risk of a machine; conversely, the deteriorating machine will have a higher probability to produce defectives that will become low-quality feedstocks of its neighboring downstream machines (NDM). When no intervention is provided, these interacting behaviors between machine reliability and WIP quality will be propagated along process routes, bringing a higher failure risk for machines at the posterior end.

In MS, machine failures have two modes [39,40]: failures making the machine stop functioning as soon as they occur (hard failure) and failures that are usually not selfannouncing and are rectified only at inspections (soft failure). The hard failure will lead to machine breakdown and zero production velocity. Soft failure is caused by component degradation, which is induced when accumulative degradation first passes the threshold [41]. Before soft failure is induced, the degradation can only impact the process quality of a machine and does not impact the production velocity. In industrial practice, machine maintenance and quality inspection are both important activities to improve MS performance, where the former can recover the machine from failure status, and the latter can cut the loss timely by avoiding the flow of defectives among machines. Corrective maintenance (CM) can be implemented to repair the breakdown machine when a hard failure or detected soft failure happens. Also, the PM can be implemented to restore the degraded machine when the degradation has not induced soft failure [42]. On the other hand, PM can decrease the probability of hard failures [5,43]. Furthermore, processing quality of the machine can also be improved.

During MS operation, if one machine fails and restoration is not completed in time, the impact of this breakdown machine will be propagated to its upstream and downstream machines. Then, the production velocities of these machines will be decreased. Moreover, idle machines may emerge because of the block or starvation. For example, the breakdown machine M5 in node 5 shown in Fig. 2 will lead to idle machine M3 because M5 is its unique NDM. On the other hand, the production velocities of machines M4 and M7 will decrease since machine M6 can sustain their operations with reduced production velocities. Further, farther downstream and upstream machines will be impacted before the breakdown machine is repaired. When the breakdown machine is repaired, production velocities of itself and impacted machines will also be recovered. However, the oscillation of production

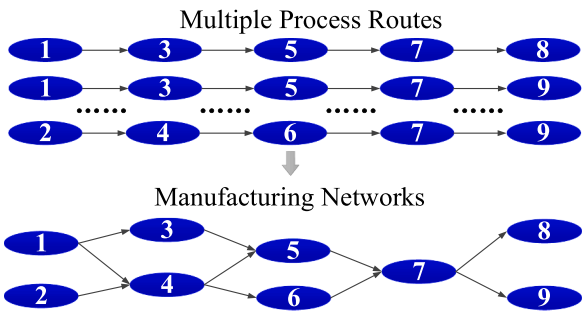


Fig. 1. Manufacturing networks caused by multiple process routes.

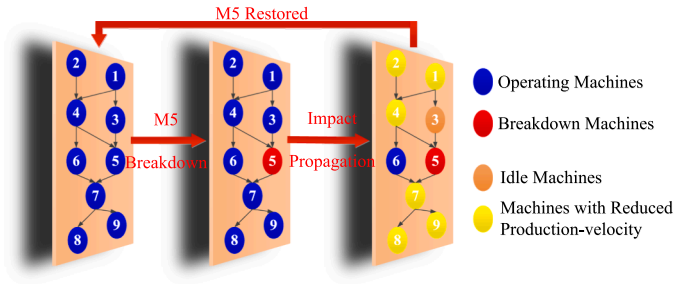


Fig. 2. Cascading impacts of breakdown machines in a manufacturing network.

velocities will lead to more uncertainties for machine degradation, weakening the effectiveness of pre-made maintenance and quality inspection plans.

In this context, dynamically optimizing machine maintenance and quality inspection activities is necessary to operate manufacturing networks efficiently. Therefore, a double-level study is proposed to explore manufacturing networks' optimal reliability and quality control policy, as shown in Fig. 3. At the machine level, dynamic production velocity caused by machine breakdowns, machine reliability considering the impact of feedstock quality (Q-R Impact), and processing quality considering the effects of machine reliability (R-Q Impact) are constructed. Therefore, machine status and manufacturing network states are evaluated systematically. At the system level, a DRL-based optimization model is proposed to learn optimal policies about quality inspection and maintenance under given manufacturing network states, where the economic operation of manufacturing networks is used as the criteria for policy evaluation.

Also, some assumptions should be considered in this study as follows:

- Discrete-part manufacturing process is considered.
- Each machine has a quality inspection activity, which can sample WIPs for inspection with any fraction in [0%, 100%].
- Feedstocks (WIP) have two quality statuses: high-quality (non-defective) and low-quality (defective), which are classified by the given quality specification.
- The machine has two failure modes: soft and hard failure.

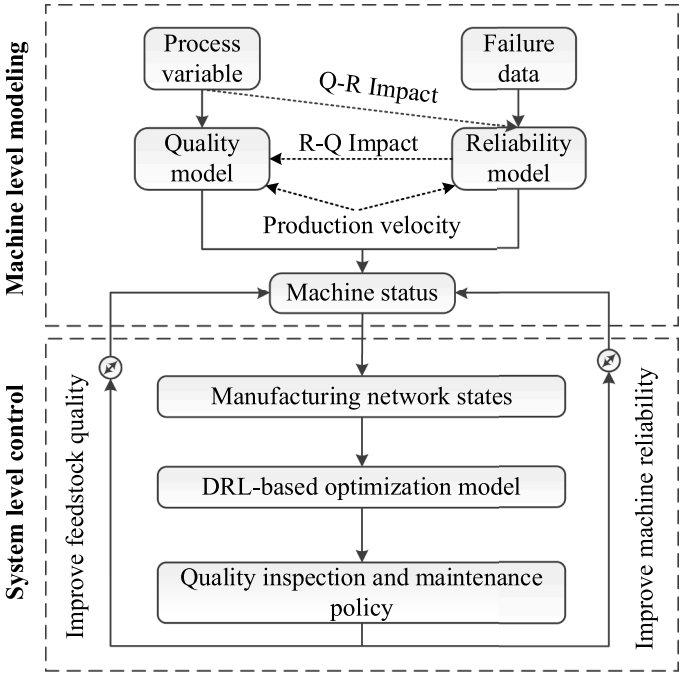


Fig. 3. Research framework.

- Two kinds of maintenance activities will be performed: corrective maintenance and preventive maintenance, which can restore machines as good as new.

3. Dynamic reliability and quality models of machine

3.1. Dynamic production velocity

In this study, an n-nodes acyclic manufacturing network is modeled by a directed acyclic graph $G(V, E)$, where $V=\{v_1, v_2,\ldots,v_n\}$ is the set of nodes representing manufacturing machines and $E\subseteq V \times V=\{(i, j) \mid v_i, v_j \in V \text{ and } i \neq j\}$ is the set of directed edges representing WIP flows. The matrix W^+ represents the forward weights of edges,

$$W^+ = \begin{bmatrix} w_{11}^+ & w_{12}^+ & \cdots & w_{1n}^+ \\ w_{21}^+ & w_{22}^+ & \cdots & w_{2n}^+ \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1}^+ & w_{n2}^+ & \cdots & w_{nn}^+ \end{bmatrix}, \tag{1}$$

where $w_{ij}^+ \in [0,1]$ is the forward weight of directed edges (i, j), representing the probability that a WIP in machine i flows into machine j. Furthermore, the inverse weight W^- of directed edges can be calculated by W^+, where $w_{ij}^- \in [0,1]$ represents the probability that a WIP in machine j comes from machine i. We define $D_i=\{v_j \mid \text{if } w_{ij}^+ \neq 0 \text{ for } j\in[1, n]\}$ as the set of NDM of machine i, and $U_j=\{v_i \mid \text{if } w_{ij}^+ \neq 0 \text{ for } i\in[1, n]\}$ as the set of neighboring-upstream machines (NUM) of machine j.

The production velocity of machines when no machine breakdown occurs in a manufacturing network is defined as the maximum production velocity, denoted as $\boldsymbol{P_{rm}}= [P_{rm1}, P_{rm2}, \ldots, P_{rmn}]$. Denote the actual production velocity at time t as $\boldsymbol{P_{ra}}= [P_{ra1}(t), P_{ra2}(t), \ldots, P_{ran}(t)]$, where $\boldsymbol{P_{ra}} \leq \boldsymbol{P_{rm}}$. During operation, the actual production velocity of a machine closely depends on its NDM and NUM so that a balanced production can be maintained. When the production velocity of one machine i has a change $\Delta P_{rai}(t)$, the production velocity of its NUM and NDM will also have corresponding changes $\Delta P_{rak}(t)$ and $\Delta P_{raj}(t)$, respectively,

$$\Delta P_{rak}(t) = \Delta P_{rai}(t)\cdot w_{ki}^-, \ (k \in U_i), \tag{2}$$

$$\Delta P_{raj}(t) = \Delta P_{rai}(t)\cdot w_{ij}^+, \ (j \in D_i) \tag{3}$$

Then, the impact of a change $\Delta P_{rai}(t)$ will propagate along its process routes to the source and sink nodes, and corresponding change will be induced in each machine on related process routes.

3.2. Dynamic machine reliability and maintenance activities

During work, a machine will suffer from dynamic failure probability. This dynamic can be attributed to three aspects: the uncertainty of the failure pattern of machines, the dynamic of production velocity, and the instability of feedstock quality. Fortunately, Weibull distribution can fit different failure patterns of machines, such as decreasing, constant, and increasing failure rates. Therefore, the failure rate of Weibull distribution is suitable for modeling the dynamic failure risk of a machine. Define the failure rate when only processing high-quality feedstocks as the basic failure rate $r_b(t)$. When considering the dynamic of production velocity, the basic failure rate is denoted as,

$$r_b(t) = (\beta / \alpha)\cdot(t_r/\alpha)^{\beta-1}, \tag{4}$$

where α and β are scale and shape parameters. t_r is the relative operating time of the machine when considering the maximum production velocity as the criterion, which is calculated by

$$t_r = \left(\int_0^t P_{ra}(u)du \right) \Big/ P_{rm}, \tag{5}$$

where t is the actual operating time of a machine. Also, the relative time t_r can be used as an index to measure machine degradation. Considering possible shocks caused by low-quality feedstocks, the actual failure rate is defined as

$$r(t) = r_b(t) + \sum_{i=1}^{N(t)} \Delta r_i, \tag{6}$$

where Δr_i is the failure-rate increment caused by one processed low-quality feedstock, obeying a Beta distribution $Beta(a, b)$. $N(t)$ is the number of low-quality feedstocks a machine processes during [0, t). Then, based on the equation between failure rate and failure probability, the failure probability function can be derived as

$$F(t) = 1 - \exp\left(-(t_r/\alpha)^{\beta} - \int \Delta r(t)dt \right), \tag{7}$$

where $\Delta r(t) = \sum_{i=1}^{N(t)} \Delta r_i$ is the accumulative increment of failure rates caused by all processed low-quality feedstocks until time t, whose integration can be derived as

$$\int_0^t \Delta r(s)ds = \int_0^t \left(\sum_{i=1}^{N(s)} \Delta r_i \right) = \sum_{i=1}^{N(t)} \left(\Delta r_i \cdot \frac{1}{P_{rm}} \int_{t_i}^{t} P_{ra}(u)du \right), \tag{8}$$

where t_i is the occurring moment of the ith failure-rate increment.

For maintenance, this study assumes both CM and PM will recover the failure rate of a machine to its initial level at time $t = 0$. Also, assume that the time of CM and PM both are random variables following normal distributions, denoted as $t_{cm} \sim N(\mu_{cm}, \sigma_{cm}^2)$ and $t_{pm} \sim N(\mu_{pm}, \sigma_{pm}^2)$, respectively. Generally, CM needs to recover sudden breakdown machines, which will be less prepared than PM. Therefore, it is reasonable to assume CM will need more time to be finished, and we set that $\mu_{cm} \geq \mu_{pm}$, $\sigma_{cm}^2 \geq \sigma_{pm}^2$. Also, we denote the costs per unit time of CM and PM are c_{cm} and c_{pm}, respectively. Then, the total maintenance cost during [0, t) in a machine is

$$c_m(t) = \sum_{i=1}^{N_{cm}(t)} c_{cm} \cdot t_{cm\_i} + \sum_{j=1}^{N_{pm}(t)} c_{pm} \cdot t_{pm\_j}, \tag{9}$$

where $N_{cm}(t)$ and $N_{pm}(t)$ are frequencies of CM and PM during [0, t), and t_{cm_i} and t_{pm_j} are times spent in ith CM and jth PM.

3.3. Dynamic processing quality and inspection activities

The processing quality is another important criterion of machine reliability, which can be depicted by the number $M(t)$ of defectives produced during a particular time [0, t). A defective refers to a WIP that falls short of quality specifications. Due to the instability of machine reliability, the processing quality also has a time-varying feature. Therefore, the random variable $M(t)$ is modeled by a non-homogeneous Poisson process (NHPP) with the following time-varying intensity function $\lambda(t)$,

$$\lambda(t) = \omega - \varepsilon \cdot e^{-\delta \cdot r(t)}, \tag{10}$$

where $t \geq 0$, $\omega > 0$, $\varepsilon > 0$, $\delta > 0$, and $r(t)$ is the actual failure rate of a machine. Simultaneously, this function satisfies $\omega - \varepsilon < \lambda(t) < \omega \leq P_{ra}(t)$, where $P_{ra}(t)$ is the actual production velocity of the machine at time t. When we define $\omega = P_{ra}(t)$, it can be demonstrated that $\varepsilon = g \times \omega$, where $g \in [0,1]$ is the initial percentage of non-defectives produced by a machine [44]. Then, we define n_d as the number of defectives produced by the machine during [t, $t+\Delta t$), and its probability can be calculated by,

$$P\{M(t+\Delta t) - M(t) = n_d\} = e^{(-m(t+\Delta t)+m(t))} \frac{[m(t+\Delta t) - m(t)]^{n_d}}{n_d!}, \tag{11}$$

where $m(t) = \int_0^t \lambda(s)ds$ is the expected function of defectives during [0, t). We define n_q as the total number of non-defectives produced by a machine during [t, $t+\Delta t$), which is obtained by

$$n_q = \int_t^{t+\Delta t} P_{ra}(u)du - n_d. \tag{12}$$

In a manufacturing network, inspection activities follow after processing activities of machines to ensure defective WIPs can be timely detected. Generally, Type I (false rejection) and Type II (false acceptance) errors may happen in an inspection activity, with probabilities denoted as p_{I} and p_{II}, respectively. Correspondingly, correct judgments of non-detectives and defectives, called the correct acceptance and correct rejection, have the probabilities 1-p_{I} and 1-p_{II}, respectively. If no inspection activity is implemented in a machine, assume p_{I}=0 and p_{II}=1.

Denote sampling fraction (randomly sampling) in an inspection activity as s_a ($s_a \in [0, 1]$) and assume that both sampling and inspection activities are independent. Then, the probabilities of type I error for each non-defective and Type II error for each defective are $s_a \bullet p_{\mathrm{I}}$ and $s_a \bullet p_{\mathrm{II}}$. We define the numbers of Type I and Type II errors are n_{fr} and n_{fa}, which obeys Binomial distributions $B(n_q, s_a \bullet p_{\mathrm{I}})$ and $B(n_d, s_a \bullet p_{\mathrm{II}})$, respectively. Therefore, the number $M'(t)$ of defectives flowing out of a machine can be considered as a compound of NHPP $M(t)$ and Binomial distribution $B(n_d, s_a \bullet p_{\mathrm{II}})$, which will become low-quality feedstocks of NDM. Then, the probability that n_{fa} defectives flow out from a machine during [t, $t+\Delta t$) can be obtained by Eq. (13).

$$P\{M'(t+\Delta t) - M'(t) = n_{fa}\} = e^{[s_a \cdot p_{II} \cdot (-m(t+\Delta t)+m(t))]} \frac{[s_a \cdot p_{II} \cdot (-m(t+\Delta t)+m(t))]^{n_{fa}}}{n_{fa}!}, \tag{13}$$

where $m(t)$ is the average defective flowing out from a machine during [0, t]. Further, the probability that n_{fr} non-defectives are rejected during [t, $t+\Delta t$) in a machine can be obtained by Eq. (14).

$$P\{D(t+\Delta t) - D(t) = n_{fr}\} = C_{n_q}^{n_{fr}} \cdot (s_a p_I)^{n_{fr}} \cdot (1 - s_a p_I)^{n_q - n_{fr}}. \tag{14}$$

where $D(t)$ is the accumulative rejected non-defectives until time t. Then, we define n_{cr} and n_{ca} as the number of correctly rejected defects and the number of correctly accepted non-defectives during [t, $t+\Delta t$), which can be calculated by

$$\begin{cases} n_{cr} = n_d - n_{fa} \\ n_{ca} = n_q - n_{fr} \end{cases}. \tag{15}$$

Also, the defective proportion judged by the inspection activity in a machine during [t, $t+\Delta t$) is

$$q = \frac{n_{cr} + n_{fr}}{n_d + n_q} = \frac{n_{cr} + n_{fr}}{\int_t^{t+\Delta t} P_{ra}(u)du} \tag{16}$$

Denote the inspection cost per WIP as c_{ins}. Then, the total inspection cost in a machine during [t, $t+\Delta t$) can be calculated by

$$c_I = c_{ins} \cdot s_a \cdot \int_t^{t+\Delta t} P_{ra}(u)du \tag{17}$$

Assume the accumulative value increment of a WIP from a source machine to machine i (including machine i) is v_{si}, which is the value loss caused by rejecting a WIP in machine i. The average value increment v_{ai} brought by processing one WIP in a machine i is defined as

$$v_{ai} = \sum_{k \in U_i} \bar{w_{ki}} \cdot (v_{si} - v_{sk}) \tag{18}$$

where U_i is the set of NUM of machine i. Further, when one defective is falsely accepted, it will flow into the NDM, and more resources will be costed, so the induced value loss will be greater than the one correctly rejected. And this value loss is denoted as $\varphi \bullet v_{si}$ (φ>1). Then, the net value increment from processing WIPs during [0, t) in machine i is the sum of three parts: the value increment of all correctly accepted WIPs, the value loss of falsely accepted WIPs, and the value loss of all rejected WIPs, as shown in Eq. (19),

$$v_{net\_i} = v_{ai} \cdot n_{ca\_i} - \varphi \cdot v_{si} \cdot n_{fa\_i} - v_{si} \cdot \left(n_{cr\_i} + n_{fr\_i}\right), \tag{19}$$

where n_{ca_i}, n_{fa_i}, n_{cr_i}, and n_{fr_i} respectively are numbers of WIPs that are correctly accepted, falsely accepted, correctly rejected, and falsely rejected in machine i.

4. Joint optimization model of maintenance and quality inspection in manufacturing networks

4.1. Evaluating manufacturing network states

As described in Section 2, machine performance has different representations, such as breakdown caused by hard or soft failure, idle caused by starvation or block, changes in processing quality, and the measured degradation level, all of which will impact the performance of a manufacturing network. For a manufacturing network with n machines, the following vector is constructed to represent its state at time t,

$$S^k = \left[\, Q^k;\quad D^k;\quad H^k;\quad O^k \,\right], \tag{20}$$

where $t = k\Delta t$, and Δt is the period length of quality statistic. Here $Q^k = [q_1^k, q_2^k, .., q_n^k]$ is the defective proportion during [t-Δt, t) in each machine, representing machines' quality status, which is calculated by Eq. (16). D^k=[t_{r1}, t_{r2}, …,t_{rn}] is machines' degradation status at time t, represented by the relative time of each machine from the last maintenance, as shown in Eq. (5). H^k=[h_1, h_2, …,h_n] is machines' health status at time t and $h_i \in \{0, 1\}$, where 1 represents the breakdown state of a machine and 0 represents its breakdown-free state. O^k=[o_1, o_2, …,o_n] is machines' idle status at time t, and $o_i \in \{0, 1\}$, where 1 represents the idle state and 0 represents its working state.

4.2. Evaluating cumulative performance of a manufacturing network

A cumulative performance evaluation is to judge whether the maintenance and quality inspection scheme for all machines is cost-efficient. From the economic viewpoint, the cumulative performance of a manufacturing network can be represented by the net profit. The step reward r_k is defined as the net profit brought by the operation of a manufacturing network during [t, t+Δt) (when $t = k\Delta t$), where the state of a manufacturing network transits to S^{k+1} from S^k. Here, Δt is the period length of quality statistic, which is also the step size to measure state transition. Therefore, the step reward equals cumulative net value increments after deducting the costs of maintenance, inspection, and decision, as shown in Eq. (21).

$$r_k = \sum_{i=1}^{n} \left(v_{net\_i} - c_{I\_i} - c_{m\_i}\right) - c_D, \tag{21}$$

where v_{net_i}, c_{I_i}, and c_{m_i} are net value increments of WIPs processed, total inspection cost, and total maintenance cost in machine i, respectively. c_D is the decision cost of maintenance and inspection actions. Then, to evaluate the cumulative performance during one epoch [$k\Delta t$, $k'\Delta t$) where the state transit to $S^{k'}$ from S^k, the return G_k is defined as the long-term reward of a manufacturing network, which is calculated by the cumulative step rewards as follows,

$$G_k = r_k + r_{k+1} + r_{k+2} + \ldots + r_{k'} = \sum_{i=k}^{k'} r_i. \tag{22}$$

where $k' > k$, both of which are the serial number of steps.

4.3. Markov decision process-based joint optimization model

The constructed dynamic reliability and quality models in Section 3 provide a state transition model of manufacturing networks. When quality inspection and maintenance are considered as actions, a typical MDP-based control model can be constructed, where the states, actions, return function, and state transition model are known. This model is devoted to seeking the optimal policy function of PM and quality inspection so that the manufacturing network will get the optimal long-term reward. In practice, CM will be implemented autonomously when machine breakdown occurs, which does not need policy. Therefore, the policy only refers to actions of PM and quality inspection at time $t = k\Delta t$, which is denoted as $A^k = [S_a^k, P_m^k] = [s_{a1}^k, s_{a2}^k, \cdots, s_{an}^k, p_{m1}^k, p_{m2}^k, \cdots, p_{mn}^k]$, where S_a^k are quality inspection actions and P_m^k are PM actions. Here, the step size Δt is also the interval to make new decisions about actions. Also, the action for all machines at $t = k\Delta t$ depends on the state S^k, so it is denoted as $A^k = \pi(S^k)$, where $\pi(\cdot)$ represents the policy function.

Let $V(S^k)$ denote the maximum expected long-term reward during one given epoch [$k\Delta t$, $k'\Delta t$) when only considering states S^k, denoted as $V(S^k) = E(G_k \mid S^k)$, which is the value function of the MDP. Let $Q(S^k, A^k)$ denote the maximum expected long-term reward during one given epoch [$k\Delta t$, $k'\Delta t$) when considering both states S^k and actions A^k, denoted as $Q(S^k, A^k) = E(G_k \mid S^k, A^k)$, which is the Q-value function of the MDP. The optimal policy of PM and quality inspection, i.e., the optimal policy function, can be denoted as

$$\pi^*\left(S^k\right) = \underset{A^k}{argmax} Q^*\left(S^k, A^k\right). \tag{23}$$

Also, under the optimal policy, the value function and the Q-function satisfy that,

$$V\left(S^k\right) = \underset{A^k}{max} Q^*\left(S^k, A^k\right). \tag{24}$$

5. The deep deterministic policy gradient algorithm

5.1. Algorithm framework

Traditional dynamic programming or heuristic algorithms can finish the optimization task in the finite-horizon MDP with enumerable state and action spaces. However, in this study, defective proportion Q^k and degradation status D^k are continuous state spaces, and quality inspection has a continuous action space whose sampling fraction can be any value located in [0, 1]. Therefore, the MDP of manufacturing networks has uncountable state and action spaces. Furthermore, the state and action spaces of the constructed MDP will also increase exponentially with machine numbers, leading to the curse of dimensionality [23]. Therefore, traditional dynamic programming cannot solve this infinite-horizon sequential decision problem. Although heuristic algorithms can provide an optimal solution through their strong search capability, they cannot continually guarantee optimal MS performance due to their low capacity for transfer learning when the manufacturing scenario varies.

In current research on the algorithm, the DRL algorithm's learning capacity has been proven effective in dealing with infinite-horizon MDP. In particular, the DQN algorithm [36], the actor-critic algorithm [45], and the DDPG algorithm [23] have been proven to be effective in solving different MDPs about maintenance. All these three algorithms are suitable for MDP with continuous or discrete state spaces, but each has its

own applicative type of action space [46]. However, the DDPG, drawing advantages of the neural network-based Q function and the actor-critic framework, shows more excellent stability than the other two algorithms [47]. Therefore, the DDPG algorithm suitable for continuous action space is utilized in this study.

The DDPG algorithm utilizes two neural networks with parameters θ_μ and θ_Q to approximate policy and value functions. Here, the policy and value functions are defined as the actor $\mu(S)$ and critic $Q(S, A)$ in the actor-critic framework, as shown in Fig. 4. Besides, the activation functions are Relu function, and hidden layers are fully connected layers. In the actor network, the state matrix S^k is the input, and the corresponding action A^k is the output to maximize long-term reward. Also, due to owning the same normalized neuron in the final output layer, quality inspection actions S_a^k and PM actions P_m^k have continuous output values in [0, 1]. To deal with the inconsistency between continuous quality inspection actions and discrete PM actions, we discretize PM actions by a given PM criterion $c_d \in [0, 1]$. In detail, no PM is carried out when $p_{mi}^k \le c_d$; otherwise, a PM will be carried out when $p_{mi}^k > c_d$. Afterward, both state matrix S^k and action vector A^k are used as inputs of the critic network, and the corresponding expectation of the long-term reward, Q value $Q(S^k, A^k)$, is the output.

5.2. Training algorithm of the agent

Based on constructed neural networks, the agent iteratively interacts with the manufacturing network environment. During the interaction, the process in the interval $[t, t+\Delta t)$ (where $t = k\Delta t$) is defined as a step of the DRL algorithm, during which a state transition from S^k to S^{k+1} of a manufacturing network occurs. An epoch refers to the evaluation period of long-term rewards during $[k\Delta t, k'\Delta t)$, composed of multiple steps. An episode refers to the process of an agent executing a mission, which is composed of multiple epochs. In the training process, the agent will iteratively simulate the DDPG algorithm until the maximum number of steps. At time t=0 (k=0), assume no quality inspection and PM is implemented in the initial action A^0. Also, assume there is no defective, machine degradation, breakdown machine, and idle machine in the initial state S^0. The detailed training process is depicted as follows and shown in Fig. 5.

Procedure 1. Executing actions and simulating the manufacturing network operation during the epoch $[k\Delta t, k'\Delta t)$ where $t = k\Delta t$. Denote $i = k$.

Subprocedure 1–1. State evaluation at time $t = k\Delta t$. Based on the dynamic reliability and quality model proposed in Section 3, evaluate machine states at time $t = k\Delta t$. Then, based on the method in Section 4.1, evaluate the state S^k of the manufacturing network at time $t = k\Delta t$.

Subprocedure 1–2. Generate action based on current policy $\pi(S)$. By inputting the observed state S^k to the actor network $\mu(S)$, get the action $A^k = [S_a^k, P_m^k]$. Here, a stochastic noise N_r obeying normal distribution is added to help the DDPG algorithm try admissible actions and explore a better strategy, namely $A^k = \pi(S^k) + N_r$. Then, according to the PM criterion c_d, the PM actions P_m^k are converted to discrete executable actions {0, 1}, where 0 represents that no PM is implemented, and 1 represents that PM is implemented. To avoid preference caused by criterion c_d, we define c_d =0.5, representing the actor network's median of the output range [0,1].

Subprocedure 1–3. Execute actions and simulate the manufacturing network operation during the next step $[k\Delta t, k\Delta t+\Delta t)$. After obtaining the action A^k at time $t = k\Delta t$, a new sampling fraction S_a^k will be executed immediately in the quality inspection for the corresponding machines. Simultaneously, PM actions will be executed. Also, if machine breakdown occurs during $[k\Delta t, k\Delta t+\Delta t)$, a CM will be executed as soon as it happens. The PM time t_{pm} and CM time t_{cm} are determined by the normal distribution $N(\mu_{pm}, \sigma_{pm}^2)$ and $N(\mu_{cm}, \sigma_{cm}^2)$ defined in Section 3.2.

Subprocedure 1–4. Evaluating step reward r_k and judging the end condition of the subprocedure. According to Eq. (21), calculate the step reward r_i. Then, $i = i + 1$. If $i < k'$, back to Subprocedure 1–1; else ($i = k'$), stop the subprocedure, and start Procedure 2.

Procedure 2. Obtaining transition record. After the operation during an epoch $[k\Delta t, k'\Delta t)$, calculate the long-term reward G_k by the Eq. (22). Based on the method in Subprocedure 1–1, obtain the current state $S^{k'}$ at $t = k'\Delta t$. Then, the transition record $\{S^k, A^k, G_k, S^{k'}\}$ is stored in the experience buffer. When the number of transition records reaches the maximum storage L, the earliest record will be discarded when a new transition record needs to be stored.

Procedure 3. Updating the actor and critic. Sample a random mini-batch M of transition records from the experience buffer and update the actor $\mu(S)$ and critic $Q(S, A)$.

Procedure 4. Judging the end condition of the procedure. If the episode reaches the predefined maximum training steps or a stable long-term reward G_k is obtained, stop training; else, updating the simulating epoch: $[k\Delta t, k'\Delta t) \leftarrow [k'\Delta t, 2k'\Delta t\text{-}k\Delta t)$, and back to procedure 1.

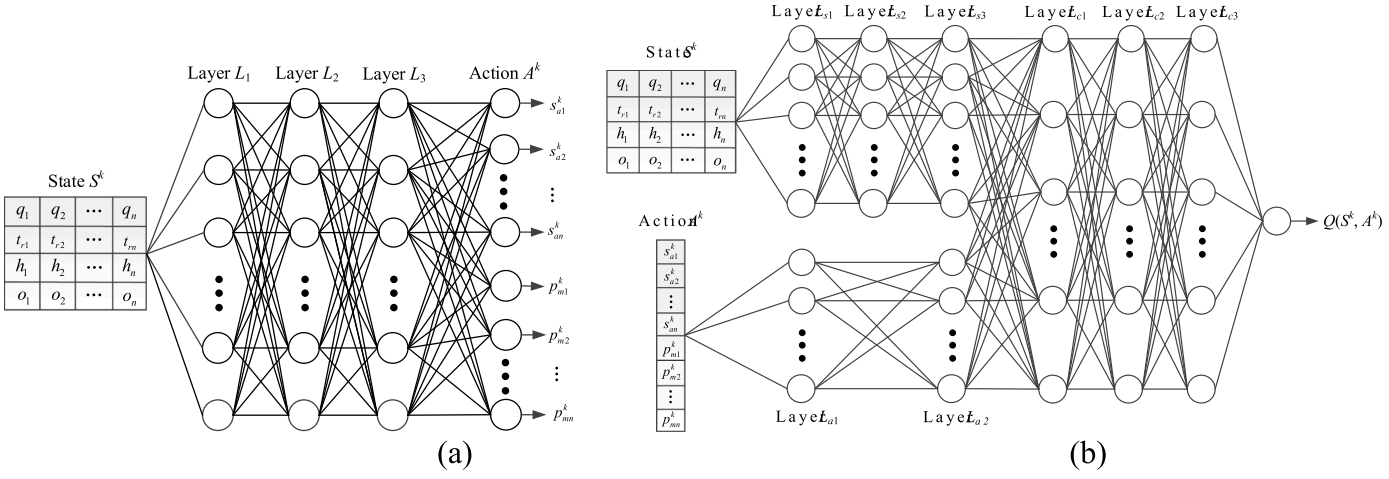


Fig. 4. Structures of neural networks in the DDPG algorithm (a) The actor network (b) The critic network.

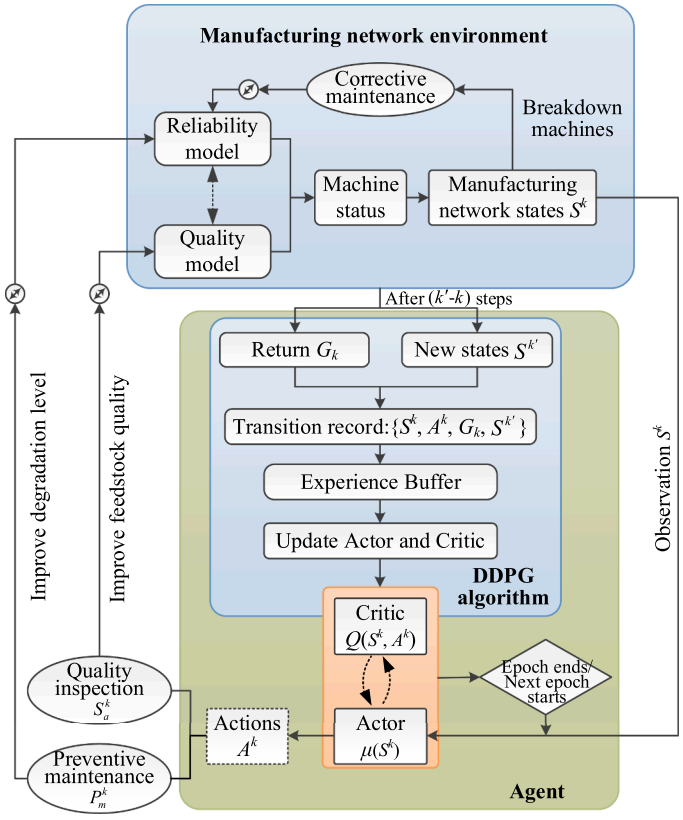


Fig. 5. The training process of DDPG-based DRL model.

5.3. Updating the algorithm of actor and critic

Before training, two neural networks owning the same structures with actor and critic are utilized to construct target actor $\mu'(S)$ and target critic $Q'(S, A)$, respectively. Also, the actor $\mu(S)$ and critic $Q(S, A)$ are initialized with random parameters θ_μ and θ_Q, and the target actor $\mu'(S)$ and target critic $Q'(S, A)$ are initialized with $\theta_{\mu'} = \theta_\mu$ and $\theta_Q = \theta_Q$. The target actor and target critic are periodically updated based on the latest actor and critic parameters to improve optimization stability [46]. Based on the transition record in the experience buffer, the neural networks will be updated in each training epoch according to the flowchart shown in Fig. 6. The updating algorithm is illustrated as follows.

Procedure 1. Sample a random mini-batch M of transition records from experience buffer: $\{S^k_{(i)}, A^k_{(i)}, G^k_{(i)}, S^{k'}_{(i)}\}$ $i = 1,2,\ldots, M$.

Procedure 2. For the transition records $i = 1,2,\ldots, M$, calculate the future target action $\mu'(S^{k'}_{(i)})$ and target future long-term reward $Q'(S^{k'}_{(i)}, \mu'(S^{k'}_{(i)}))$. And set the value function target $y_i = G^k_{(i)} + Q'(S^{k'}_{(i)}, \mu'(S^{k'}_{(i)}))$.

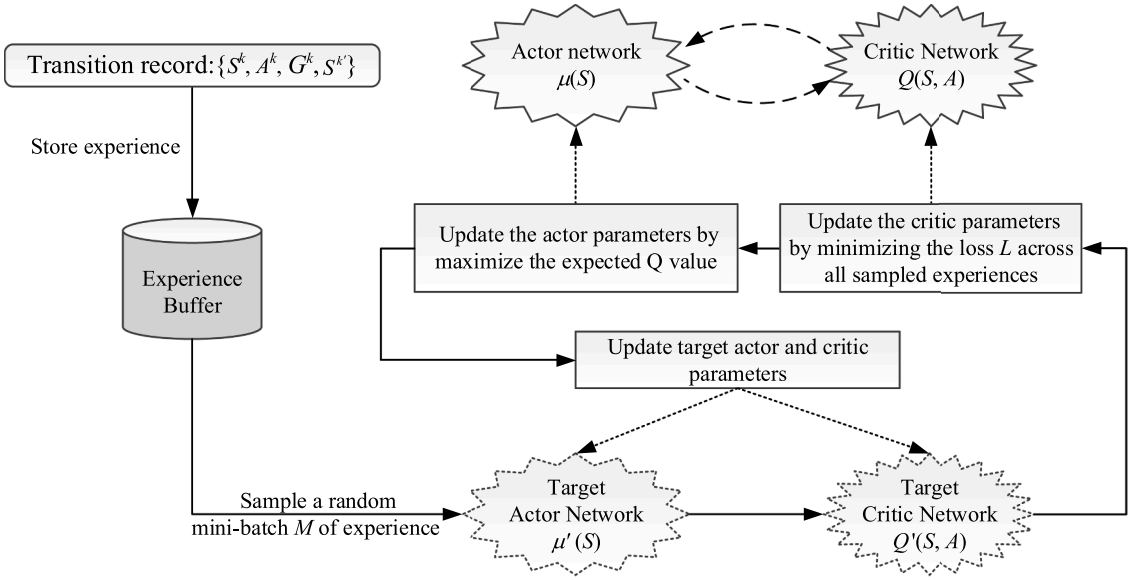


Fig. 6. The updating process of actor and critic in the DDPG algorithm.

Procedure 3. Update the parameters θ_Q of the critic network $Q(S, A)$ by minimizing the following Loss function:

$$Loss = \frac{1}{M}\sum_{i=1}^{M}\left(y_i - Q\left(S_{(i)}^k, A_{(i)}^k\right)\right)^2. \tag{25}$$

Procedure 4. Update the parameters θ_μ of the actor network $\mu(S)$ by maximizing the expected cumulative long-term rewards:

$$\theta_\mu = \underset{\theta_\mu}{argmax}\sum_{i=1}^{M} Q\left(S_{(i)}^k, \mu\left(S_{(i)}^k\right)\right) \tag{26}$$

Procedure 5. Update the parameters of the target actor and critic networks by the following E.q.:

$$\theta_{\mu'} = \tau\theta_\mu + (1-\tau)\theta_{\mu'}, \tag{27}$$

$$\theta_{Q'} = \tau\theta_Q + (1-\tau)\theta_{Q'}, \tag{28}$$

where τ is the smoothing factor.

6. Case study

6.1. Manufacturing network and agent parameters

In this case, a directed acyclic network with 30 nodes is used to model a manufacturing network with multiple process routes [44], as shown in Fig. 7. Here, the machine is depicted as nodes that have different reliability parameters, and WIP flows between machines is represented by edges. The WIP flows along directed edges randomly, where the quantity is restricted by machine capacity. This manufacturing network has 4 source nodes, 4 sink nodes, and 51 directed edges, formulating 724 different process routes.

The agent is trained by MATLAB R2021a. Based on the learning rate used in related research [48], the learning rates of the critic and actor networks in this training are set as 2×10^{-3} and 1×10^{-3}. Because the number of neurons in hidden layers dramatically depends on the dimension of the concerned problems, considering similar research [29], the neuron number of hidden layers are respectively set as L_{s1}=128, L_{s2}=256, L_{s3}=128, L_{a1}=64, L_{a2}=128, L_{c1}=256, L_{c2}=128, L_{c3}=64, L_1=128, L_2=256, L_3=128. Also, based on existing studies, other parameters are: the experience buffer length $L = 1 \times 10^6$, the mini-batch size of sampling M = 1280, the smoothing factor τ=1×10^{-3}, and the decision cost c_D=100.

6.2. DRL agent training

In this section, we train the DRL agent to obtain the maximum return during an epoch of 5000-unit time, namely $(k'\text{-}k)\times\Delta t = 5000$. Considering the possible sensitivity of agent performance to step size [49], different step sizes Δt and decision frequencies k'-k are used to construct three different manufacturing scenarios: {Δt = 50, k'-k = 100}, {Δt = 100, k'-k = 50)}, and {Δt = 500, k'-k = 10}, meaning the agent will generate actions A^k and interact with the manufacturing network 100, 50, and 10 times during each epoch. The training stops when the step number reaches a pre-specified value.

Besides, the genetic algorithm (GA) is used as the baseline of the proposed method. For the encoding, a 1 × 60 matrix represents the chromosome divided into two portions, i.e., PM actions and quality inspection actions of the 30 nodes. For the PM action, genes are numbers 1 and 0, where 1 denotes the 'PM' action and 0 represents the 'skip maintenance' action [11]. For quality inspection actions, genes are random decimals in the range [0,1] representing sampling fractions. Also, an initial population with 70 individuals is generated randomly, and stochastic mutation and one-point crossover are used in our GA algorithm [4,5]. Then, an elitist strategy is adopted, where the return G_k during 5000-unit time is used as the fitness function to select the best individual. For constructed three manufacturing scenarios, the maximum evolutionary generation (*MEG*) is set as the time that the manufacturing network operates 7×10^5 steps, which equals the training-step number under the DRL algorithm, namely {Δt = 50, *MEG* =100}, {Δt = 100, *MEG* =200)}, and {Δt = 500, *MEG* =1000}.

Based on the DRL and GA algorithms, three training episodes are implemented for each manufacturing scenario. The DRL algorithm's mean, standard deviations (*SD*), and Coefficient of Variation (*CV*) are calculated based on returns of the last 100 epochs when the return becomes stable. For the GA algorithm, they are calculated based on returns of the last 50 generations when the return becomes stable, as shown in Table 3. The *CV* shown in Eq. (29) represents the relative dispersions of returns.

$$CV = SD/mean. \tag{29}$$

For the DRL algorithm, the highest returns are 4.83×10^4, 3.63×10^4, and 6.47×10^4 when Δt = 50, 100, and 500, respectively; and for GA, the highest returns are 2.63×10^4, 4.25×10^4, and 6.39×10^4. Only when Δt = 100 could GA help the manufacturing network get a better average return than the DRL algorithm. Therefore, the proposed DRL algorithm has better adaptability to various manufacturing scenarios than GA. Also, in most training, the *CV*s under GA are more significant than the ones under the DRL algorithm, which implies that obtained returns by GA have lower stability than the DRL algorithm.

Further, training trajectories of the highest return under different manufacturing scenarios by both algorithms are illustrated in Fig. 8. They show that our constructed DRL agent can improve the manufacturing network's long-term rewards, proving our model's effectiveness. When Δt = 50, the trajectory implies that higher frequent interaction between manufacturing environment and DRL agent will result in slow convergence of the learning process and lead to a significant loss, as the negative return shown in Fig. 8. When Δt = 100 and Δt = 500, the trajectories show that a fast convergence can be realized when lower frequent interactions are implemented, but the obtained return is non-optimal compared to the one under GA when Δt = 100. As the green line shown in Fig. 8, the maximum return (4.22×10^4) under the DRL algorithm verges on the best mean under GA when Δt = 100, but its return trajectory shows a clear downtrend subsequently. Compared to GA, therefore, the DRL algorithm also has an equivalent capability in exploring the best solution when Δt = 100. However, its

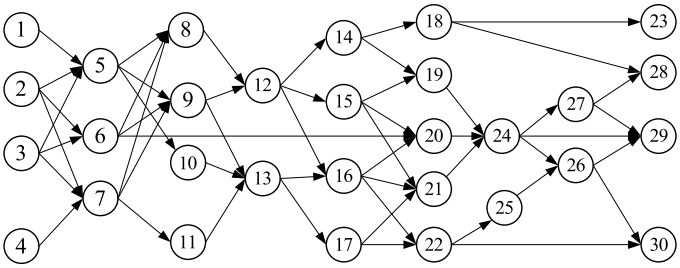


Fig. 7. A directed acyclic manufacturing network.

Table 3
Training returns by the DRL algorithm and GA.

Δt	Training No.	*DRL* Mean ($\times 10^4$)	*SD* ($\times 10^2$)	*CV* ($\times 10^{-2}$)	*GA* Mean ($\times 10^4$)	*SD* ($\times 10^2$)	*CV* ($\times 10^{-2}$)
50	1	4.83	1.01	2.08	2.53	5.44	2.15
	2	3.45	7.53	2.18	2.55	5.53	2.17
	3	3.16	6.26	1.98	2.63	4.90	1.86
100	1	3.63	2.71	0.75	4.21	4.21	1.00
	2	3.61	3.45	0.96	4.25	4.61	1.09
	3	3.61	2.80	0.78	4.23	3.84	0.91
500	1	6.47	3.46	0.53	6.33	6.90	1.09
	2	5.97	3.40	0.57	6.39	7.90	1.24
	3	5.97	2.86	0.48	6.38	6.66	1.04

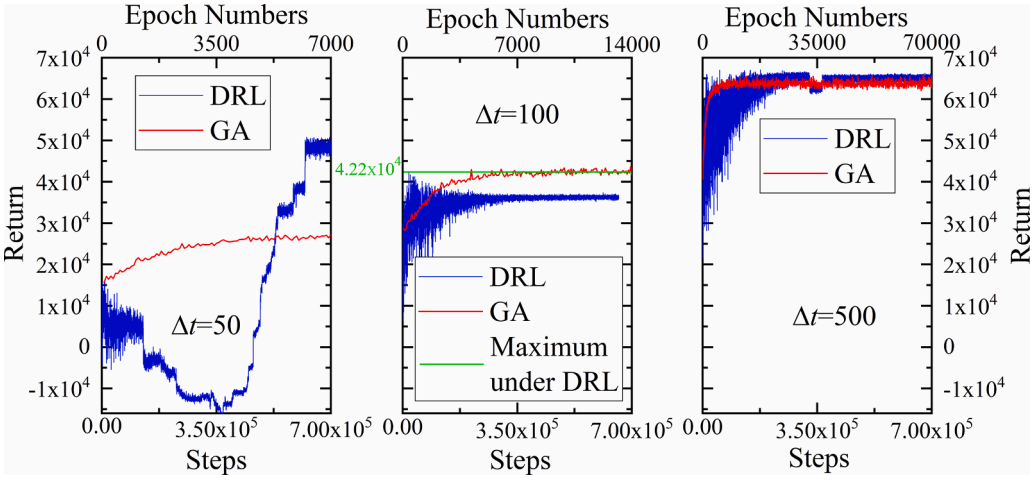


Fig. 8. Trajectories of highest returns under different manufacturing scenarios.

ability to maintain the optimal solution is inferior to GA. In summary, because of the high nonlinearities, high dimensions, and dynamics of manufacturing networks, the learning performance of DRL is sensitive to the manufacturing scenario with different step sizes.

6.3. Experiments based on trained agents

Through the optimal DRL agents trained by the above three manufacturing scenarios, the control of maintenance and quality inspection for the given manufacturing network during 50,000-unit time is implemented. Accumulative returns and *CV* of step rewards are shown in Fig. 9. By DRL agent, the manufacturing network can obtain

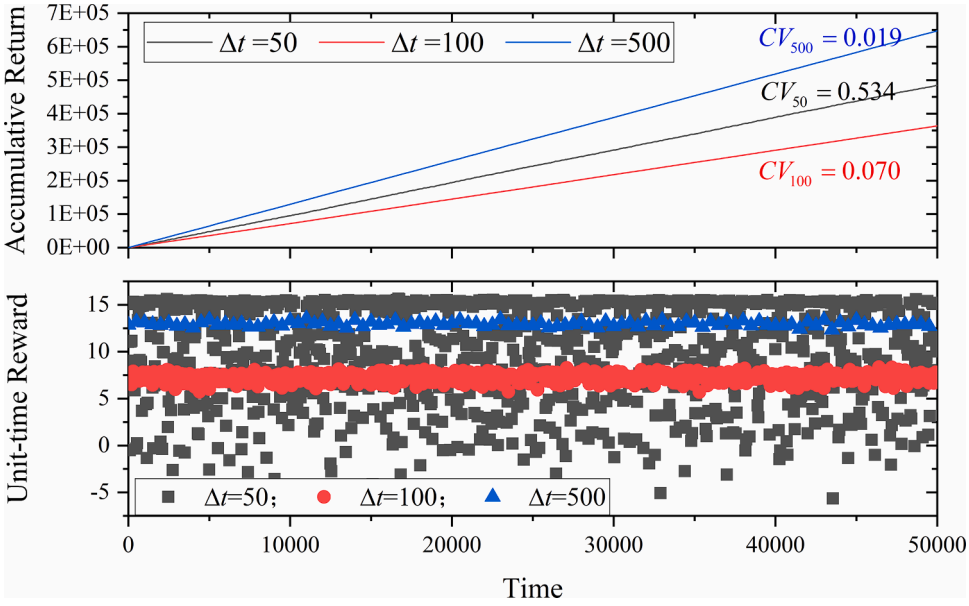


Fig. 9. Reward analysis.

continuously increasing accumulative returns under all manufacturing scenarios. Again, when interacting step size Δt equals 500, the manufacturing network can achieve the highest accumulative return. Simultaneously, the CV represents that the dispersions of step rewards under different step sizes have the relations: $CV_{500}<CV_{100}<CV_{50}$. Therefore, the performance agents keep consistent with their training results, representing that the experimental process is steady and effective.

Also, this study calculates the unit-time reward in each step based on the experimental results. For a step k, its unit-time reward $r_{u(k)}$ can be obtained by the Eq.,

$$r_{u(k)} = r_k/\Delta t. \tag{30}$$

Then, a scatter diagram of unit-time rewards is given in Fig. 9. In the scatter diagram, the unit-time rewards are stable and centralized when step size Δt equals 100 or 500, and the unit-time rewards when $\Delta t = 500$ are greater than those when $\Delta t = 100$. When $\Delta t = 50$, however, the unit-time rewards become dispersive and volatile. This shows that, because of the high nonlinearity, high dimension, and dynamics of manufacturing networks, a smaller step size made the agent difficult to give optimal decisions continuously.

To analyze the trends of step rewards, we implement the low pass filter on the time-series rewards by the Fast Fourier Transform (FFT), as shown in Fig. 10. The red curves of filtered step rewards show that cycle C_w of waves satisfies the properties of $C_{w500}>C_{w100}>C_{w50}$, representing that the DRL agent has greater volatility and poor stability when $\Delta t = 50$.

At last, this study analyzes the connectivity of the manufacturing network during the 50,000-unit time under the intervention of three trained DRL agents. Here, connectivity refers to the probability that the manufacturing network at least has one process route connected from source nodes to sink nodes. A detailed calculation can be found in Reference [44]. The curves of connectivity are illustrated in Fig. 11. When step size $\Delta t = 500$, the manufacturing network has the worst connectivity and the highest variance, representing that the connectivity is not only the worst but also the most unstable. When step size $\Delta t = 100$, however, it has the best connectivity and lowest variance, verifying the philosophy that high return comes with high risk. In detail, when the manufacturing network keeps a higher long-term reward ($\Delta t = 500$), it will have a high risk of operational interruptions (the worst connectivity).

7. Conclusion

This work investigated the DRL-based joint optimization of maintenance and quality inspection in manufacturing networks in the presence of interactions between machine reliability and WIP quality. First, mathematical models are proposed to construct manufacturing networks' nonlinear, high-dimensional, and dynamic environments. The proposed models provide an adequate state transition model for manufacturing networks' control. Second, an effective DRL model suitable for the reliability-quality joint control of manufacturing networks is constructed, where mixed discrete-continuous states and actions can be realized simultaneously. Furthermore, contrast training with GA provides validation for the effect of the proposed DRL model. The proposed DRL algorithm is more adaptable to dynamic and diverse manufacturing scenarios than GA. Also, experiments represent that our proposed models can help to balance the contradiction between economic profit and operational risk of the manufacturing network.

This article presents a state transition model to depict the operation of a networked MS, providing a feasible reliability-quality-based digitization scheme of MS. In the future, a more detailed digital-twin model can be developed based on the proposed MS model, which could provide a realistic training environment for AI-based research on maintenance and quality inspection. Besides, as mentioned in Section 1, joint optimization of MS includes many other aspects, such as production scheduling, human reliabilities, or the reliability of soft systems. In future expansion, a more comprehensive joint optimization should be considered, where the operator skill training and soft system maintenance should be integrated, to reduce the MS failure caused by human error or soft bugs.

CRediT authorship contribution statement

Zhenggeng Ye: Conceptualization, Methodology, Software, Writing – original draft, Project administration. **Zhiqiang Cai:** Validation, Supervision, Writing – review & editing, Project administration. **Hui Yang:** Methodology, Writing – review & editing. **Shubin Si:** Supervision, Writing – review & editing. **Fuli Zhou:** Investigation, Writing – review & editing.

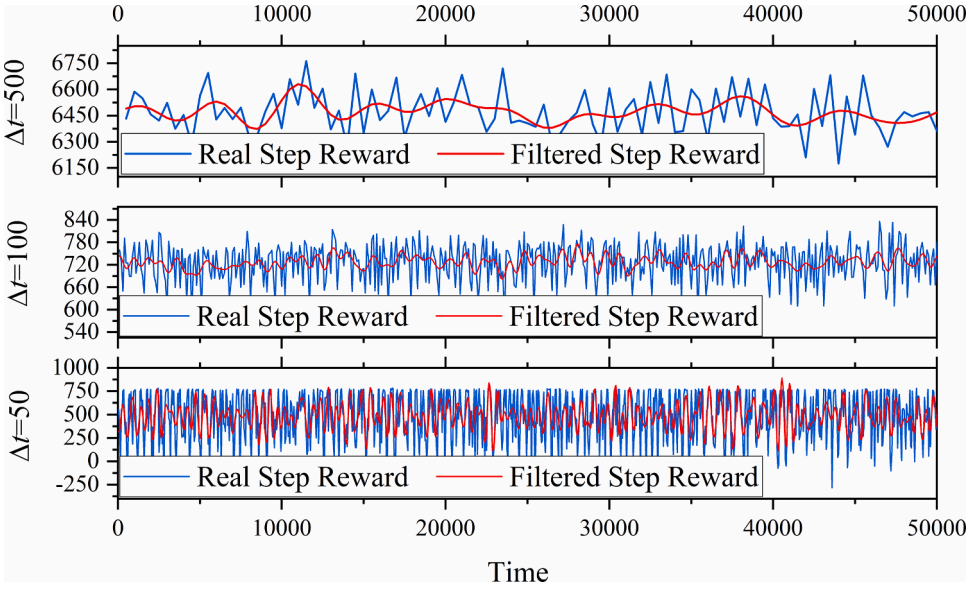


Fig. 10. Trend analysis of step rewards.

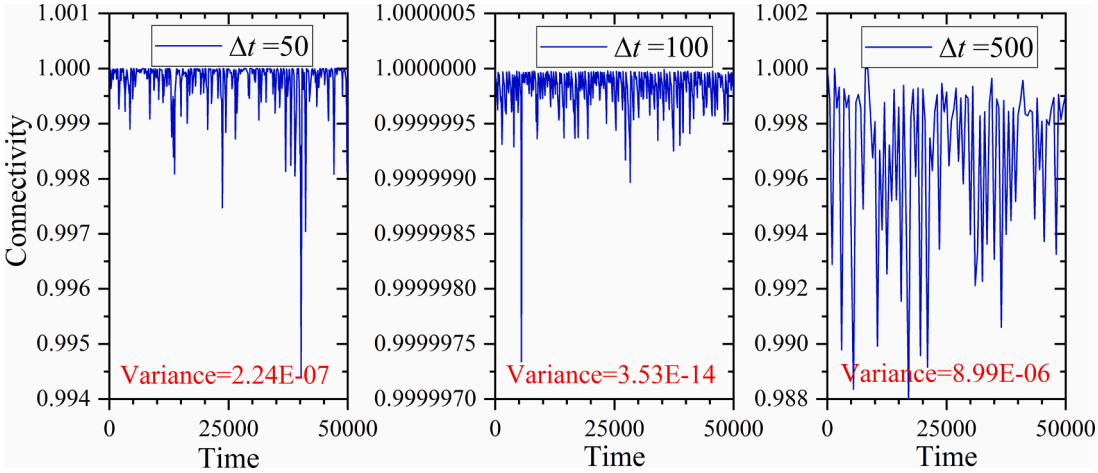


Fig. 11. Connectivity of manufacturing networks under the control of different trained DRL agents.

Declaration of Competing Interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Data availability

Data will be made available on request.

Fundings

This work was supported by the National Natural Science Foundation of China [Grant Nos. 72201250, 72271200], the Key Technologies R&D Programme of Henan Province from Henan Science & Technology Department [Grant No. 222102210005], the Humanities & Social Sciences Project from the Ministry of Education in PRC [Grant No. 22YJC630220], the Natural Science Basic Research Program of Shaanxi [Grant No. 2020JM-150].

References

[1] Yang H, Kumara S, Bukkapatnam STS, Tsung F. The internet of things for smart manufacturing: a review. IISE Trans 2019;51:1190–216.
[2] Han X, Wang Z, Xie M, He Y, Li Y, Wang W. Remaining useful life prediction and predictive maintenance strategies for multi-state manufacturing systems considering functional dependence. Reliab Eng Syst Saf 2021;210:107560.
[3] Si S, Zhao J, Cai Z, Dui H. Recent advances in system reliability optimization driven by importance measures. Front Eng Manag 2020;7:335–58.
[4] Xiao L, Song S, Chen X, Coit DW. Joint optimization of production scheduling and machine group preventive maintenance. Reliab Eng Syst Saf 2016;146:68–78.
[5] Liu Y, Zhang Q, Ouyang Z, Huang H-Z. Integrated production planning and preventive maintenance scheduling for synchronized parallel machines. Reliab Eng Syst Saf 2021;215:107869.
[6] Lu B, Zhou X, Li Y. Joint modeling of preventive maintenance and quality improvement for deteriorating single-machine manufacturing systems. Comput Ind Eng 2016;91:188–96.
[7] Kolus A, El-Khalifa A, Al-Turki UM, Duffuaa SO. An integrated mathematical model for production scheduling and preventive maintenance planning. Int J Qual Reliab Manag 2020;37:925–37.
[8] Ait El Cadi A, Gharbi A, Dhouib K, Artiba A. Joint production and preventive maintenance controls for unreliable and imperfect manufacturing systems. J Manuf Syst 2021;58:263–79.
[9] Cheng G, Zhou B, Li L. Integrated production, quality control and condition-based maintenance for imperfect production systems. Reliab Eng Syst Saf 2018;175:251–64.
[10] Ait-El-Cadi A, Gharbi A, Dhouib K, Artiba A. Integrated production, maintenance and quality control policy for unreliable manufacturing systems under dynamic inspection. Int J Prod Econ 2021;236:108140.
[11] Tambe PP, Kulkarni MS. A reliability based integrated model of maintenance planning with quality control and production decision for improving operational performance. Reliab Eng Syst Saf 2022;226:22.
[12] He Y, Gu C, Chen Z, Han X. Integrated predictive maintenance strategy for manufacturing systems by combining quality control and mission reliability analysis. Int J Prod Res 2017;55:5841–62.
[13] Zhou X, Lu B. Preventive maintenance scheduling for serial multi-station manufacturing systems with interaction between station reliability and product quality. Comput Ind Eng 2018;122:283–91.
[14] He YH, Liu FD, Cui JM, Han X, Zhao YX, Chen ZX, et al. Reliability-oriented design of integrated model of preventive maintenance and quality control policy with time-between-events control chart. Comput Ind Eng 2019;129:228–38.
[15] Bouslah B, Gharbi A, Pellerin R. Joint production, quality and maintenance control of a two-machine line subject to operation-dependent and quality-dependent failures. Int J Prod Econ 2018;195:210–26.
[16] Alimian M, Ghezavati V, Tavakkoli-Moghaddam R. New integration of preventive maintenance and production planning with cell formation and group scheduling for dynamic cellular manufacturing systems. J Manuf Syst 2020;56:341–58.
[17] Ye Z, Cai Z, Si S, Zhang S, Yang H. Competing failure modeling for performance analysis of automated manufacturing systems with serial structures and imperfect quality inspection. IEEE Trans Ind Inf 2020;16:6476–86.
[18] Li YF, Jia C. An overview of the reliability metrics for power grids and telecommunication networks. Front Eng Manag 2021;8:531–44.
[19] Wang L, Bai Y, Huang N, Wang Q. Fractal-based reliability measure for heterogeneous manufacturing networks. IEEE Trans Ind Inf 2019;15:6407–14.
[20] Meng X, Cai Z, Si S, Duan D. Analysis of epidemic vaccination strategies on heterogeneous networks: based on SEIRV model and evolutionary game. Appl Math Comput 2021;403:126172.
[21] Zuo M. System reliability and system resilience. Front Eng Manag 2021;8:615–9.
[22] Cai B, Sheng C, Gao C, Liu Y, Shi M, Liu Z, et al. Artificial intelligence enhanced reliability assessment methodology with small samples. IEEE Trans Neural Netw Learn Syst 2021;online:1–13. https://doi.org/10.1109/TNNLS.2021.3128514.
[23] Chen Y, Liu Y, Xiahou T. A deep reinforcement learning approach to dynamic loading strategy of repairable multistate systems. IEEE Trans Reliab 2022;71:484–99.
[24] Zhang H, Di X, Zhang Y. Real-time CU-net-based welding quality inspection algorithm in battery production. IEEE Trans Ind Electron 2020;67:10942–50.
[25] Zhou Z, Oguz OS, Leibold M, Buss M. Learning a low-dimensional representation of a safe region for safe reinforcement learning on dynamical systems. IEEE Trans Neural Netw Learn Syst 2021;online:1–15. https://doi.org/10.1109/TNNLS.2021.3106818.
[26] Paraschos PD, Koulinas GK, Koulouriotis DE. Reinforcement learning for combined production-maintenance and quality control of a manufacturing system with deterioration failures. J Manuf Syst 2020;56:470–83.
[27] Xanthopoulos AS, Kiatipis A, Koulouriotis DE, Stieger S. Reinforcement learning-based and parametric production-maintenance control policies for a deteriorating manufacturing system. IEEE Access 2018;6:576–88.
[28] Wang X, Wang H, Qi C. Multi-agent reinforcement learning based maintenance policy for a resource constrained flow line system. J Intell Manuf 2016;27:325–33.
[29] Huang J, Chang Q, Arinez J. Deep reinforcement learning based preventive maintenance policy for serial production lines. Expert Syst Appl 2020;160:113701.
[30] Epureanu BI, Li X, Nassehi A, Koren Y. Self-repair of smart manufacturing systems by deep reinforcement learning. CIRP Ann 2020;69:421–4.
[31] Skordilis E, Moghaddass R. A deep reinforcement learning approach for real-time sensor-driven decision making and predictive analytics. Comput Ind Eng 2020;147:106600.
[32] Zhao YF, Smidts C. Reinforcement learning for adaptive maintenance policy optimization under imperfect knowledge of the system degradation model and partial observability of system states. Reliab Eng Syst Saf 2022;224:108541.
[33] Yousefi N, Tsianikas S, Coit DW. Reinforcement learning for dynamic condition-based maintenance of a system with individually repairable components. Qual Eng 2020;32:388–408.

[34] Hu Y, Miao X, Zhang J, Liu J, Pan E. Reinforcement learning-driven maintenance strategy: a novel solution for long-term aircraft maintenance decision optimization. Comput Ind Eng 2021;153:107056.
[35] Liu Y, Chen Y, Jiang T. Dynamic selective maintenance optimization for multi-state systems over a finite horizon: a deep reinforcement learning approach. Eur J Oper Res 2020;283:166–81.
[36] Zhang N, Si W. Deep reinforcement learning for condition-based maintenance planning of multi-component systems under dependent competing risks. Reliab Eng Syst Saf 2020;203:107094.
[37] Andriotis CP, Papakonstantinou KG. Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints. Reliab Eng Syst Saf 2021;212:107551.
[38] Nguyen V, Do P, Vosin A, Iung B. Artificial-intelligence-based maintenance decision-making and optimization for multi-state component systems. Reliab Eng Syst Saf 2022;228:108757.
[39] Taghipour S, Banjevic D. Optimum inspection interval for a system under periodic and opportunistic inspections. IIE Trans 2012;44:932–48.
[40] Che H, Zeng S, Li K, Guo J. Reliability analysis of load-sharing man-machine systems subject to machine degradation, human errors, and random shocks. Reliab Eng Syst Saf 2022;226:108679.
[41] Gao H, Cui L, Qiu Q. Reliability modeling for degradation-shock dependence systems with multiple species of shocks. Reliab Eng Syst Saf 2019;185:133–43.
[42] Wei S, Nourelfath M, Nahas N. Analysis of a production line subject to degradation and preventive maintenance. Reliab Eng Syst Saf 2023;230:108906.
[43] Li Y, Xia T, Chen Z, Pan E. Multiple degradation-driven preventive maintenance policy for serial-parallel multi-station manufacturing systems. Reliab Eng Syst Saf 2023;230:108905.
[44] Ye Z, Si S, Yang H, Cai Z, Zhou F. Machine and Feedstock Interdependence Modeling for Manufacturing Networks Performance Analysis. IEEE Trans Ind Inf 2022;18:5067–76.
[45] Lee J, Mitici M. Deep reinforcement learning for predictive aircraft maintenance using probabilistic remaining-useful-life prognostics. Reliab Eng Syst Saf 2023;230:108908.
[46] Reinforcement learning toolbox™ user's guide. Natick: Natick: The MathWorks, Inc; 2020.
[47] Zhao YP, Wang XQ, Wang RY, Yang YP, Lv F. Ieee. Path Planning for Mobile Robots Based on TPR-DDPG. In: Proceedings of the international joint conference on neural networks (IJCNN). electr network; 2021. Ieee.
[48] Qiu C, Hu Y, Chen Y, Zeng B. Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications. IEEE Internet Things J 2019;6:8577–88.
[49] Asi H, Duchi JC. The importance of better models in stochastic optimization. Proc Natl Acad Sci 2019;116:22924–30.