Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 2942940, 10 pages
https://doi.org/10.1155/2020/2942940

Research Article

Preventive Maintenance Interval Optimization for Continuous Multistate Systems

Ruiying Li [1,2] and Xufeng Zhang [1]

[1] *School of Reliability and Systems Engineering, Beihang University, Beijing 100191, China*
[2] *Science and Technology on Reliability and Environmental Engineering Laboratory, Beijing 100191, China*

Correspondence should be addressed to Ruiying Li; liruiying@buaa.edu.cn

Received 24 September 2019; Revised 6 December 2019; Accepted 21 December 2019; Published 20 January 2020

Academic Editor: Zhiwei Gao

Preventive maintenance (PM), which is performed periodically on the system to lessen its failing probability, can effectively decrease the loss caused by the system breakdown or the performance degradation. The optimal PM interval has been well studied for both binary-state systems (BSSs) and discrete multistate systems (MSSs). However, in reality, the performance of many systems can change continuously, ranging from complete failure to perfect functioning. Considering such characteristics of systems, two types of performance-based measures, performance availability and probabilistic resilience, are addressed to quantify the system's behaviour for continuous MSS. A Monte Carlo-based method is given to analyse the performance change process of the system, and an optimization framework is proposed to find the optimal PM interval with the considerations of per-unit-time cost, system breakdown rate, performance availability, and probabilistic resilience. A computer cluster is used as an example to illustrate the effectiveness of our proposed method.

1. Introduction

In industry, preventive maintenance (PM), a planned or scheduled maintenance activity, is an essential maintenance policy [1]. PM was proposed in 1940s, requiring engineers to take maintenance actions when the system is about to fail. In this way, the probability of system breakdown can be reduced or even avoided. Hence, how to determine the proper PM interval to avoid excessive or insufficient maintenance is a critical issue. With the practice of PM, maintenance activities changed from reactive to proactive. Later, condition-based maintenance (CBM) was proposed. Using such strategy, the optimal maintenance timing can be determined according to system's current condition. However, some system's performance cannot be easily monitored without interrupting the system's operation, so PM is still widely applied in industry.

The topic of how to determine an optimal PM interval for systems attracts a lot of researchers. The excessive maintenance may affect the normal operation of the system and result in high maintenance costs. On the contrary, the insufficient maintenance may cause frequent system breakdown, affecting the user's experience. The optimal PM planning problem was firstly studied for binary-state systems (BSSs), which simplifies the system states into normal and failure [2, 3]. Group/block replacement models [4, 5] and opportunistic maintenance models [6] were proposed successively to find the optimal PM planning. However, if a system with multiple performance states is assumed as a BSS, its reliability is computed as the probability that their performance stayed above the threshold, and its performance changing process is then neglected. Considering that the system may have multiple operable states or operate at multiple performance levels, reliability theory then started to focus on discrete multistate systems (MSSs) [7], which has more than two but finite performance states. Levitin and Lisnianski [8] firstly studied the maintenance decision problem of discrete MSS and proposed the imperfect PM optimization framework in [9]. Then, many researchers proposed or applied several methods, such as the Markov process [10] and the universal generating function (UGF) [8, 9, 11], to find the optimal PM interval for discrete MSS. For example, Liu et al. [12] proposed a value-centric dynamic PM policy, and they determined which component to maintain and the maintenance level by maximizing the maintenance net value. However, in reality, the performance of many systems can change continuously, ranging from complete failure to perfect functioning. These systems are named as continuous MSS, or continuous-state systems briefly [13]. Obviously, binary-state-based or discrete multistate-based PM interval optimization frameworks are insufficient to describe the characteristics of the continuous MSS. For example, the Markov method is based on the models of system states and the state transitions, and UGF method is based on the possible system states and its corresponding probability. The infinite number of performance states makes the Markov and UGF methods difficult to apply for continuous MSS.

The continuous changing performance of such systems requires advanced measures. Performance-based measures, e.g., performance availability and resilience, provide a new perspective to analyse such systems. Performance availability, an extension of traditional availability in the performance dimension, qualifies the average performance of the continuous MSS over its lifetime. Probabilistic resilience, a new measure to assess the system's ability to withstand disturbances and return to a normal state quickly [14], focuses on the system performance after the disturbance occurs. Using such performance-based measures, Zhang and Li [15] compared different PM intervals of a discrete MSS. However, how to find the optimal PM interval and how to apply such method to continuous MSS was not addressed.

To solve such problems, we propose a PM interval optimization method for continuous MSS. The contributions of this paper include the following:

(1) Two performance-based measures, performance availability and probabilistic resilience, are proposed to analyse and optimize the PM interval of continuous MSS with the consideration of its infinite number of performance states. Using such measures, the continuous MSS can be analysed more accurate, as its continuous changing performance is depicted.

(2) A general PM interval optimization framework is provided for continuous MSS. In the optimization model, we consider four types of decision elements, including the performance availability, the probabilistic resilience, the system breakdown rate, and the per-unit-time cost. Compared with the PM interval optimization methods for BSS or discrete MSS, the system's continuous performance changing process is analysed by a Monte Carlo-based method in our framework.

The remainder of the paper is organized as follows: Section 2 describes our PM decision problem in detail. In Section 3 four key parameters, performance availability, probabilistic resilience, system breakdown rate, and per-unit-time cost, are proposed and applied as the decision elements in the PM interval optimization models for the continuous MSS. Monte Carlo method is used to compute the performance degradation and recovery process of the system and its components. In Section 4, the optimal PM interval is calculated for a computer cluster, and the optimal result is compared with the one obtained by enumeration to verify the effectiveness of our method. Finally, concluding remarks are provided in Section 5.

2. Problem Description

In this paper, we consider a continuous MSS which is composed of continuous multistate components. When a disturbance occurs on a component, its performance changes as Figure 1 shows. There are two phases:

(1) Degradation phase: after the disturbance occurs on the component, its performance begins to decline. The component continuously operates in a degraded state until it breaks down or the PM activity starts.

(2) Recovery phase: the performance of the component begins to restore back after the corrective maintenance (CM) or preventive maintenance (PM) activity is taken.

By performance monitoring and data fitting, functions of both the degradation and recovery processes of the components can be obtained. Cimellaro et al. [16] proposed three typical functions to describe the recovery behaviour of the MSS, including linear function, exponential function, and trigonometric function. These functions can also be used to describe the performance degradation process, see the study of Crk [17]. So equations (1)–(3) are applied in our paper to describe both the performance degradation and recovery processes:

$$f(t) = a\left(\frac{t - t_s}{T_p}\right) + b, \tag{1}$$

$$f(t) = a \times \exp\left[\frac{-b \times (t - t_s)}{T_p}\right], \tag{2}$$

$$f(t) = \frac{a}{2}\left\{1 + \cos\left[\frac{\pi b (t - t_s)}{T_p}\right]\right\}, \tag{3}$$

where a and b are the constant values that can be calculated by curve fitting, t_s is the start time of the corresponding process (i.e., the time that the component's performance begins to decline or restore in the degradation and recovery process, respectively), and T_p is the time duration of the degradation or recovery process.

As is well known, the performance of the system depends on that of its components. When the performance of components starts to degrade, that of the system may also decline. Then, the system will operate in a degradation state. When the system's performance drops below the threshold Q_L, CM activities will be taken on the components, and the system will restore back along with the components. Note

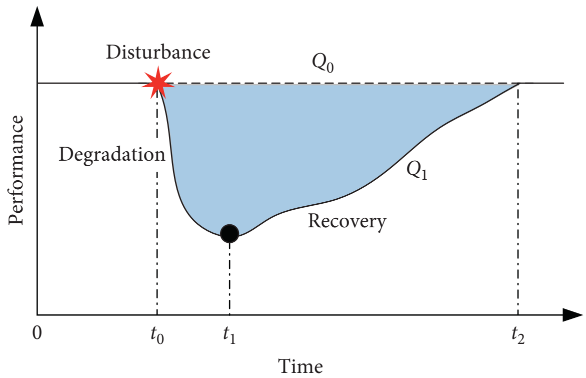

FIGURE 1: The resilience process of a continuous component/system.

that, if the system's PM interval arrives before the system breaks down, PM activities are taken instead.

In this paper, we aim to find an optimal PM interval for the continuous MMS composed of continuous multistate components. Our assumptions are as follows:

(1) The system after PM or CM is "as good as new" (i.e., perfect maintenance)
(2) There is only one maintenance channel, and only one component can be repaired each time
(3) The maintenance activity is performed immediately without administrative delay
(4) The disturbance occurrence time of components are independently and identically distributed
(5) Both the time duration of the performance degradation and recovery processes are also independently and identically distributed

3. Methods

In this section, we propose a PM interval optimization framework for continuous MSS based on the Monte Carlo method.

3.1. Decision Elements. Generally, the optimal PM interval is the time when the system is about to fail. As the performance changing is very essential for continuous MSS, we propose two types of performance-based measures, performance availability and probabilistic resilience, and apply them together with traditional PM decision elements, system breakdown rate, and per-unit-time cost, to find the optimal PM interval.

3.1.1. Performance Availability. Availability is the proportion of time that a system is in a functioning condition for binary-state systems, and it measures the degree to which the system can be used. For MSS, the performance behaviour of the system also affects its availability. In Figure 1, the solid line and the dotted line describe the real performance and the ideal performance of the system, respectively, and the performance availability can then be defined.

Definition 1. Performance availability is the ratio of the area under the system's real performance curve to that under the ideal performance curve over time 0 to T.

It can be computed as

$$A_P = \frac{\int_0^T Q_1(t)\mathrm{d}t}{\int_0^T Q_0(t)\mathrm{d}t}, \tag{4}$$

where Q_0 and Q_1 are the ideal and real performance functions of the system, respectively, and T is the operation cycle (e.g., 1 year). The performance availability is the average proportion of performance that a system behaves in [0, T]. One can see that the performance availability is an extension of the traditional availability in the performance dimension. For BSS, we have $Q_0 = 1$, $Q_1 = 1$ (if the system is operational or up), and $Q_1 = 0$ (if the system fails or down). So our performance availability can be calculated as follows:

$$A_P = \frac{\text{uptime}}{\text{operation cycle}}, \tag{5}$$

for BSS.

3.1.2. Probabilistic Resilience. Resilience is also a performance-based measure, and it reflects the ability of the system to withstand disturbance and return to a normal state quickly. Under a given disturbance, a deterministic resilience measure can be defined.

Definition 2. Deterministic resilience is the ratio of the area beneath the performance curve within its maximum allowable recovery time after a given disturbance to the ideal one in the normal state.

This deterministic measure can be calculated as

$$\mathbb{R}_D = \frac{\int_{t_0}^{t_0+T_a} Q_1(t)\mathrm{d}t}{\int_{t_0}^{t_0+T_a} Q_0(t)\mathrm{d}t}, \tag{6}$$

where t_0 is the disturbance occurrence time and T_a is system's maximum allowable recovery time. $\mathbb{R}_D$ presents the average performance of the system in $[t_0, t_0 + T_a]$ after the disturbance. Such measure is with clear physical meaning which reflects the "bounce back" ability of the system and can be used to compare the resilience of various systems on the same relative scale. Obviously, the system's resilience is a random variable, and we define probabilistic resilience under a random disturbance.

Definition 3. Probabilistic resilience is the probability that the system's resilience can satisfy users' requirements under random disturbances.

The probabilistic resilience can be computed as

$$\mathbb{R}_P = \Pr\{\mathbb{R}_D \geq \mathbb{R}_D^*\}, \tag{7}$$

where $\mathbb{R}_D^*$ is the resilience threshold defined by users. Such measure describes how well the system can meet its resilience requirements.

3.1.3. System Breakdown Rate. System breakdown rate implies the probability that the system breaks down before the PM interval arrives. It can be defined as

$$\eta = \Pr\{\text{System breaks down before PM interval arrives}\}. \tag{8}$$

3.1.4. Per-Unit-Time Cost. Cost is usually used in PM interval optimization to analyse the expense of the maintenance strategy. Considering the operation and maintenance process of the system, the cost includes six aspects: (1) operation cost; (2) maintenance labor cost; (3) maintenance material cost; (4) on-site maintenance cost; (5) system breakdown loss; and (6) system performance loss. Here, the operation cost reflects the resource consumption during the system operation; maintenance labor and material cost includes the labor and spare consumption and maintenance tool, equipment, and facility depreciation during the maintenance process; on-site maintenance cost refers to the cost of the preparation activity of each maintenance; system breakdown loss contains both the direct and indirect losses caused by the system breakdown; and system performance loss reflects the loss caused by the system performance degradation. Thus, we define the per-unit-time cost function as

$$C_{\mathrm{PUT}} = \frac{T\sum_{i}\int_{\mathrm{all}\, s_i} C_{O,i}(s_i) f_i(s_i)\mathrm{d}s_i + C_{\mathrm{MMHC}} \times (t_{\mathrm{PM}} + t_{\mathrm{CM}}) + \sum_{i} C_{\mathrm{MMC},i} N_i + C_{\mathrm{BM}} N_M + C_D \times t_D + C_P \int_0^T (Q_0(t) - Q_1(t))\mathrm{d}t}{T}, \tag{9}$$

where T is the operation cycle (e.g., 1 year); $C_{O,i}(s_i)$ is the operation cost of component i under state s_i per unit time, $f_i(s_i)$ is the probability density function of state s_i; C_{MMHC} is the maintenance labor cost per unit time, t_{PM} and t_{CM} are the total time of PM and CM over time $[0, T]$, respectively; $C_{\mathrm{MMC},i}$ is the material cost for component i, and N_i is the number of times that the performance of component i degrades and needs to be repaired over time $[0, T]$; C_{BM} is the on-site cost for one maintenance, and N_M is the number of on-site maintenance over time $[0, T]$; C_D is the system breakdown loss per unit time, and t_D is the length of the system downtime over time $[0, T]$; and C_P is the system performance loss per unit area in Figure 1.

3.2. Monte Carlo Simulation. For BSS and discrete MSS, Universal Generation Function and Markov process are often used in reliability analysis. However, for continuous MSS, it is impossible to model all the states and solve the corresponding differential equations. So the Monte Carlo method is used here to analyse the performance changing process of the continuous MSS under given PM intervals. As the system after PM or CM is supposed "as good as new," we regard one simulation cycle as the time period from the time that the system is put into use to the time that its performance fully restores back after the maintenance activity (i.e, PM or CM) is taken. In Figure 1, this time period is denoted as time $[0, t_2]$. Multiple simulation cycles together form the operation cycle T. The randomness of both disturbances that the system suffered and the system's response are then considered in each simulation cycle. In each simulation cycle, the simulation process is as follows:

(1) Sample and obtain the disturbance occurrence time and time duration of the performance degradation for each component according to their corresponding distributions.

(2) Sort the disturbance occurrence time in the ascending order, and denote them as t_1, t_2, t_3....

(3) Let $\alpha = 0$ and $\beta = 0$, where α is the number of the disturbances that have occurred and β is the number of disturbances in the current continuous disturbance behaviour. A continuous disturbance behaviour describes the situation that a new disturbance occurs on the system before the performance degradation of the previous one ends, and then these disturbances affect the system together. These continuous disturbances are denoted as a disturbance group.

(4) Let $\alpha = \alpha + 1$ and $\beta = \beta + 1$. Calculate the performance degradation end time of the current disturbance group t_{de} by comparing it with the occurrence time of the $(\alpha + 1)^{\mathrm{th}}$ disturbance.

   (a) If $t_{\mathrm{de}} > t_{\alpha+1}$, continue Step (4)
   (b) If $t_{\mathrm{de}} > t_{\alpha+1}$, the α^{th} disturbance is the last one in the current disturbance group, go to Step (5)

(5) Calculate the performance degradation process of the current disturbance group, and determine whether the performance degradation in the current disturbance group will cause the system breakdown (i.e., the system performance drops below the threshold Q_L). If so, record the time that the system breaks down as t_{Q}.

(6) Compare t_{de} and t_{Q} with the PM interval T_{PM}:

   (a) If $\min\{t_{\mathrm{de}}, t_{\mathrm{Q}}, T_{\mathrm{PM}}\} = t_{\mathrm{Q}}$, it means that the system breaks down in the current continuous disturbance process before the PM interval arrives. Go to Step (7) and start CM.

(b) If $\min\{t_{\mathrm{de}}, t_{\mathrm{Q}}, T_{\mathrm{PM}}\} = \mathrm{T}_{\mathrm{PM}}$, it means that the PM interval arrives in this continuous disturbance process before the system breaks downs. Go to Step (7) and start PM.
(c) If $\min\{t_{\mathrm{de}}, t_{\mathrm{Q}}, T_{\mathrm{PM}}\} = t_{\mathrm{de}}$, it means that neither system breaks down nor PM interval arrives in this continuous disturbance process. Let $\beta = 0$, and go to Step (4).

(7) Sample and obtain the performance recovery time for each degraded component according to their distributions, and compute the recovery process of the system.

From the Monte Carlo simulation, the system's performance can be recorded every Δt time, and the decision elements can be estimated. For example, the performance availability can be estimated as

$$\widehat{A}_P = \frac{\sum_{j=1}^{N}\sum_{k=0}^{\left[t_{e,j}/\Delta t\right]-1}\left(Q_{1,j,k} + Q_{1,j,k+1}\right)}{\sum_{j=1}^{N}\sum_{k=0}^{\left[t_{e,j}/\Delta t\right]-1}\left(Q_{0,j,k} + Q_{0,j,k+1}\right)}, \quad (10)$$

where N is the number of iterations in the simulation, $t_{e,j}$ is the system recovery end time in the j^{th} simulation iteration, and $Q_{0,j,k}$ and $Q_{1,j,k}$ are the ideal and real system performance at time $k\cdot\Delta t$ in the j^{th} simulation iteration, respectively. The performance-based probabilistic resilience can be estimated as

$$\widehat{\mathbb{R}}_P = \frac{N_{\mathbb{R}}}{N}, \quad (11)$$

where N is the number of simulation iterations, $N_{\mathbb{R}}$ is the number of iterations whose deterministic resilience $\widehat{\mathbb{R}}_D$ satisfies the resilience threshold $\mathbb{R}_D^*$ defined by users, and

$$\widehat{\mathbb{R}}_{D,j} = \frac{\sum_{k=\left[t_{s,j}/\Delta t\right]}^{\left[t_{e,j}/\Delta t\right]-1}\left(Q_{1,j,k} + Q_{1,j,k+1}\right)}{\sum_{k=\left[t_{s,j}/\Delta t\right]}^{\left[t_{e,j}/\Delta t\right]-1}\left(Q_{0,j,k} + Q_{0,j,k+1}\right)}, \quad (12)$$

where $t_{s,j}$ is the occurrence time of the first disturbance in the j^{th} simulation iteration. The system breakdown rate can be estimated as

$$\widehat{\eta} = \frac{N_d}{N}, \quad (13)$$

where N_d is the number of the system breaks down (i.e., the number of CM) in the simulation. The per-unit-time cost can be estimated as

$$\widehat{C}_{\mathrm{PUT}} = \frac{\sum_{i=1}^{n}C_{O,i}(s_i)\Delta t_i(s_i) + C_{\mathrm{MMHC}}\times\sum_{j=1}^{N}\left(t_{\mathrm{PM},j} + t_{\mathrm{CM},j}\right) + \sum_{i=1}^{n}C_{\mathrm{MMC},i}N_i + C_{\mathrm{BM}}\left(N_{\mathrm{PM}} + N_{\mathrm{CM}}\right) + C_D\times\sum_{j=1}^{N}t_{D,j} + C_P\sum_{j=1}^{N}\sum_{k=\left[t_{s,j}/\Delta t\right]}^{\left[t_{e,j}/\Delta t\right]-1}\left(Q_{0,j,k} - Q_{1,j,k}\right)\Delta t}{\sum_{j=1}^{N}t_{e,j}}, \quad (14)$$

where $\Delta t_i(s_i)$ is the time length of component i under state s_i in the whole simulation; $t_{\mathrm{PM},j}$ and $t_{\mathrm{CM},j}$ are the maintenance time of PM and CM in the j^{th} iteration, respectively; n is the number of components in the system; N_{PM} and N_{CM} are the number of PM or CM in the whole simulation; and $t_{D,j}$ is the system downtime in the j^{th} iteration.

After N rounds of iterations, we can obtain N performance change processes of the system, and then the statistical values of the four decision elements under the given PM interval can be further calculated. The value of N depends on the requirements of simulation accuracy ε. According to the central limit theorem, the number of iterations can be computed as

$$N \geq \max_x\left[\left(\frac{Z_{\alpha/2}\sigma_x}{\varepsilon_x}\right)^2\right], \quad (15)$$

where $Z_{\alpha/2}$ is the $100(1-(\alpha/2))^{\mathrm{th}}$ percentile of the standard normal distribution, $1-\alpha$ is the confidence level (e.g., $1-\alpha = 95\%$), and ε_x and σ_x are the accuracy requirement and mean square error for the parameter x (i.e., the four decision elements).

3.3. Optimization Model and Algorithm. Using the four decision elements mentioned above, we can build a PM interval optimization model for the continuous MSS. Depending on the problem, the previous PM interval optimization models can be divided into integer linear programming model [18], dynamic programming model [19], robust optimization model [20], and so on. For the continuous MSS, goal(s) and constraints in the optimization model can be determined according to the real problem. Hence, we provide two types of optimization models as examples. The first optimization model is a single-objective problem that minimizing the per-unit-time cost under the other three constraints as follows:

$$\begin{aligned} \min \quad & C_{\mathrm{PUT}} \\ \text{s.t.} \quad & \eta \leq \eta^* \\ & A_P \geq A_P^* \\ & \mathbb{R}_P \geq \mathbb{R}_P^*, \end{aligned} \quad (16)$$

where A_P^*, $\mathbb{R}_P^*$, and η^* are the threshold of the performance availability, probabilistic resilience, and system breakdown rate, respectively. The second optimization model is a multiobjective optimization problem which minimizes the per-unit-time cost and the system breakdown rate while maximizes the performance availability and probabilistic resilience:

$$\begin{aligned} \min \quad & [C_{\text{PUT}}, \eta, -A_P, -\mathbb{R}_P]^T \\ \text{s.t.} \quad & C_{\text{PUT}} \le C^*_{\text{PUT}} \\ & \eta \le \eta^* \\ & A_P \ge A^*_P \\ & \mathbb{R}_P \ge \mathbb{R}^*_P, \end{aligned} \tag{17}$$

where C^*_{PUT} is the threshold of the per-unit-time cost. Using the direct weighting method, this multiobjective optimization problem in equation (17) can be converted into a single one as

$$\begin{aligned} \min \quad & f = k_1 C_{\text{PUT}} + k_2 \eta - k_3 A_P - k_4 \mathbb{R}_P \\ \text{s.t.} \quad & C_{\text{PUT}} \le C^*_{\text{PUT}} \\ & \eta \le \eta^* \\ & A_P \ge A^*_P \\ & \mathbb{R}_P \ge \mathbb{R}^*_P, \end{aligned} \tag{18}$$

where f is the joint objective function and k_i is the weighting factor for each objective. k_i can be computed as $k_i = k_i^1 \times k_i^2$, where k_i^1 reflects the importance of each objective, $\sum k_i^1 = 1$, and k_i^2 is used to adjust the effect of the difference in magnitude among objectives (e.g., $k_i^2 = 1/\|\nabla f_i\|^2$).

To solve the PM interval optimization problem, three algorithms were usually applied, including genetic algorithms (GAs) [8, 9, 21], particle swarm optimization (PSO) [22], and simulated annealing (SA) [23]. Generally, the appropriate optimization algorithm is selected according to the type of the optimization model. For example, the simplex method can be used to solve linear optimization problems, golden section method and binary search method can be applied for nonlinear optimization problems with unimodal objective functions, and genetic algorithms, particle swarm optimization, and simulated annealing (SA) are applicable for nonlinear optimization problems with multimodal objective functions. In practice, the optimization algorithm should be determined according to the specific problem.

4. A Numerical Example

A computer cluster contains 10 connected computers that work together, as shown in Figure 2. Both the system and its components are with continuous multistates, and the load capacity of system (Q) is the sum of that of computers (q_i). Under the normal state, the load capacity of the system and components are both normalized as 100%. The components may suffer various disturbances, and the occurrence probability of these disturbances follows the exponential distribution with parameter λ. Once a disturbance occurs on a computer, its load capacity degrades gradually following the exponential function in equation (2), and the time duration of the performance degradation follows exponential distribution with parameter τ. When the computer's performance degrades below the threshold ($q_L = 50\%$), it stops working and waits for maintenance. The system's load capacity degrades along with that of the

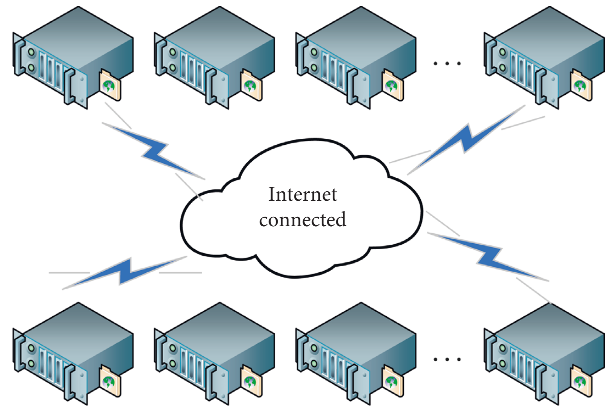

Figure 2: A computer cluster.

computers, and it can work under degradation state until the system breaks down (i.e., the load capacity of the system drops below the system's threshold ($Q_L = 60\%$)) or the PM interval (T) arrivals. If the system breaks down before the PM interval arrives, CM starts. During CM, computers that suffered disturbance are repaired offline one after another. When the number of perfect computers reaches the threshold that can provide enough performance, the system restarts working. Then, the CM activity continues until all computers are repaired and the whole system returns back to normal. If the PM interval arrives before the system breaks down, PM starts and the degraded computers are repaired online one by one. The recovery process of computers also follows the exponential function in equation (2), and the maintenance time for PM and CM follows normal distributions with parameters (μ_{PM}, ${\sigma_{\text{PM}}}^2$) and (μ_{CM}, ${\sigma_{\text{CM}}}^2$), respectively. It is assumed that the maintenance strategy is first-in-first-out (i.e., first-degrade-first-repair), and there is only one maintenance channel. Our aim is to find the optimal PM interval for the continuous multistate computer cluster. The resilience-related and cost-related parameters are shown in Tables 1 and 2.

4.1. Resilience Behaviour Analysis. We run the Monte Carlo simulation and obtained the resilience process of the computer cluster. During the operation, system may suffer both single and continuous disturbances. The difference is whether the performance degradation of the system under the disturbance ends before a new one occurs. In Figure 3, the first disturbance is a single one, and the other three are continuous ones which forms a disturbance group. In this figure, the four red stars show the time that the disturbance occurred. One can see that the performance degradation caused by the last three disturbances is superimposed on each other.

Figure 4 shows two types of typical resilience behaviours for the computer cluster, and the PM interval was set as 900 hours and 2000 hours, respectively. In Figure 4(a), four disturbances occurred in sequence, resulting in the gradual load capacity degradation of four computers in the system. The performance of the system degraded along with that of

Table 1: Resilience-related parameters.

Parameter	Distribution	Parameter 1	Value	Parameter 2	Value
Disturbance/failure occurrence time	Exponential	λ	1/8760 h^{-1}	—	—
Degradation time	Exponential	τ	1/100 h^{-1}	—	—
PM time	Normal	μ_{PM}	24	σ_{PM}	2
CM time	Normal	μ_{CM}	24	σ_{CM}	2

Table 2: Cost-related parameters.

Parameter	Value
Operation cost per unit time of a single computer (C_O)	5 yuan/hour
Maintenance labor cost per unit time (C_{MMHC})	100 yuan/hour
Maintenance material cost of a single computer (C_{MMC})	3000 yuan
On-site cost per maintenance (C_{BM})	1000 yuan
System breakdown loss per unit time (C_D)	10000 yuan/hour
System performance loss per unit area (C_P)	100 yuan/area

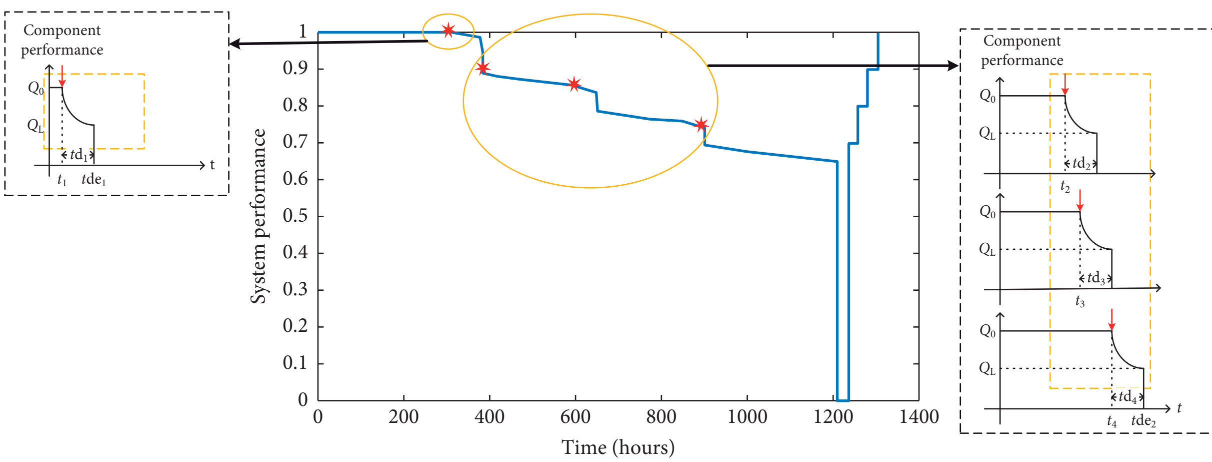

Figure 3: Single and continuous disturbance effect.

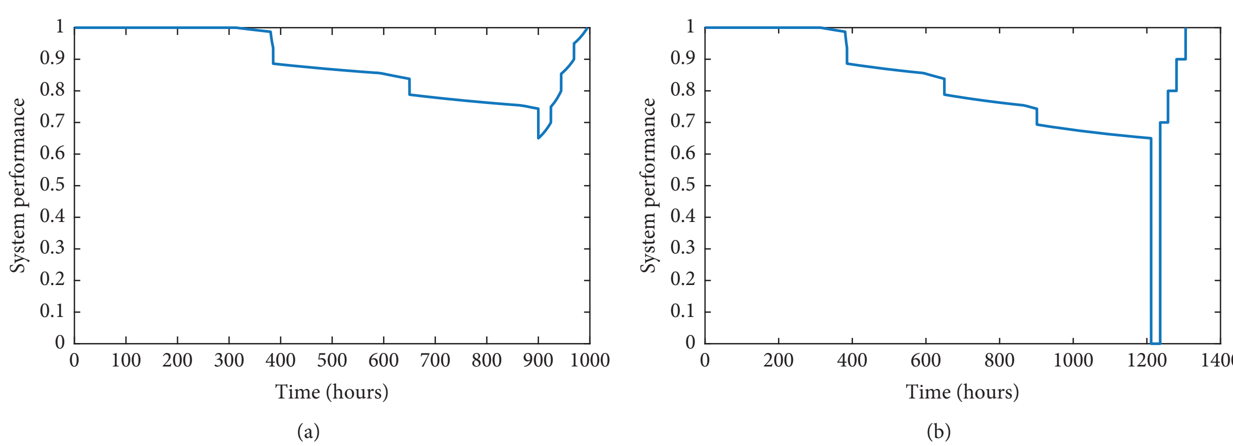

Figure 4: Typical resilience processes of the cluster system. (a) Recover by PM (PM interval = 900 hours). (b) Recover by CM (PM interval = 2000 hours).

the computers, as shown in this figure. At time $t = 900$ hours, the PM interval arrived while the system was still working, and PM was then taken to restore the system. The four degraded computers were repaired one by one according to the disturbance occurrence sequence. As the PM activity was online, the system's performance restored back gradually in this figure. In Figure 4(b), the PM interval was set as 2000 hours, and the load capacity of the system

Table 3: Optimal results.

Method	Genetic algorithms	Golden section method	Enumeration method
Computation time (s)	332.09	49.08	1830.01
Joint objective function	−4.4749	−4.4723	−4.4735
Performance availability	0.9835	0.9980	0.9985
Probabilistic resilience	0.998	1.00	1
System breakdown rate	0.00	0.00	0.00
Per-unit-time cost (yuan/hour)	￥58.92	￥64.46	￥64.96
Optimal PM interval (h)	108.8549	98.8276	92

dropped below its threshold Q_L at time $t = 1212$ hours. Then, the offline CM was taken, and the four computers were completely repaired before they restarted working. So one can see that the system performance recovery process in Figure 4(b) presents a step-like performance recovery process.

4.2. PM Interval Optimization. Firstly, a multiobjective optimization model was built to find the optimal PM interval as follows:

$$\begin{aligned} \min \quad & f = k_1 C_{PUT} + k_2 \eta - k_3 A_P - k_4 \mathbb{R}_P \\ \text{s.t.} \quad & A_P \geq 0.9 \\ & \mathbb{R}_P(\mathbb{R}_D \geq 0.8) \geq 0.9 \\ & \eta \leq 0.1, \end{aligned} \tag{19}$$

where k_i is the weighting factor for each decision element. According to the direct weighing method, $k_i = k_i^1 \times k_i^2$. Here, $K^1 = [0.6, 0.2, 0.1, 0.1]^{\mathrm{T}}$ described the importance of the four elements, and K^2 was determined according to the magnitude difference among these four elements as $K^2 = [0.0018, 66.4896, 35.28, 10.2043]^{\mathrm{T}}$. Compared with the optimization model in equation (17), the constraint of the per-unit-time cost is omitted, as there is no budget limit in this problem.

Secondly, we used the Monte Carlo method to analyse the behaviour of the cluster system under different PM intervals. We run the simulation for 1000 times at the PM interval 1000 hours and obtained the corresponding sample standard deviations for the four decision elements. Let $\varepsilon_{\widehat{C}_{\mathrm{PUT}}} = 5$, $\varepsilon_{\widehat{\eta}} = 0.002$, $\varepsilon_{\widehat{A}_P} = 0.002$, $\varepsilon_{\widehat{\mathbb{R}}_P} = 0.002$, and $1 - \alpha = 95\%$, we finally determined the number of iterations as $N = 2700$ according to equation (15). Under such 2700 simulation iterations, the error can be computed as $\varepsilon'_{\mathrm{CPUT}} < 4.3627$, $\varepsilon'_{A_P} < 0.0019$, $\varepsilon'_{\eta} < 0.0018$, and $\varepsilon'_{\widehat{\mathbb{R}}_P} < 0.0017$.

Finally, optimization algorithms, the golden section method and genetic algorithm, were used to solve the optimization problem in equation (19). The specific algorithms are as follows:

(1) Golden-section search is a technique for finding an extremum of a unimodal function inside a specified interval $[a, b]$. The functional values of $f(x)$ at two golden section points, $c = a + 0.382(b - a)$ and $d = a + 0.618(b - a)$, are calculated first. By comparing the functional values at a, c, d, and b, we can determine the extremum either in $[a, d]$ or $[c, b]$ using the characteristic of the unimodal function. In either case, a new narrower search interval can be obtained. Successively narrowing the range of values, the extremum value will finally be obtained within its accuracy requirement.

(2) Genetic algorithm is a metaheuristic inspired by the process of natural selection. The initial population of the first generation is generated randomly, and then bio-inspired operators such as mutation, crossover, and selection are used to search the extremum. In each generation, individual solutions with higher values of fitness function are more likely to be inherited by the next generation. Repeat the process for several generations, the genetic algorithm then has a tendency to converge towards the global optima of the problem. See Ref. [24] for details.

Table 3 compares the optimal results with the enumeration method. One can see that the golden section method is the most efficiency and accurate one.

4.3. Discussion. Using the enumeration method, we can obtain the simulation results at different PM intervals (Figure 5). One can see that both the performance availability and probabilistic resilience decrease along with the increase of the PM interval in Figures 5(a) and 5(b). As the PM is an online maintenance activity, the more frequent inspections and maintenances help reduce the system's performance degradation, resulting in better performance availability and probabilistic resilience. Figure 5(c) shows an obvious result, i.e., the system breakdown rate increases along with the value of the PM interval. This is because it is more convenient to discover the abnormal behaviour of the system before the system breaks down with a smaller PM interval. In Figure 5(d), the per-unit-time cost drastically decreases at the very beginning and then increases gradually along with the increase of the PM interval. If the PM interval is too small, excessive PM activities are triggered, leading to large onsite cost. If the PM interval is too large, PM is insufficient, resulting in more system breakdown and larger performance loss. This also illustrates the importance of choosing a suitable PM interval. The result is also consistent with that of the related studies in binary-state systems and discrete MSS.

Figure 6 shows the result of the joint objective function in equation (19). It illustrates more clearly the necessity of determining the optimal PM interval.

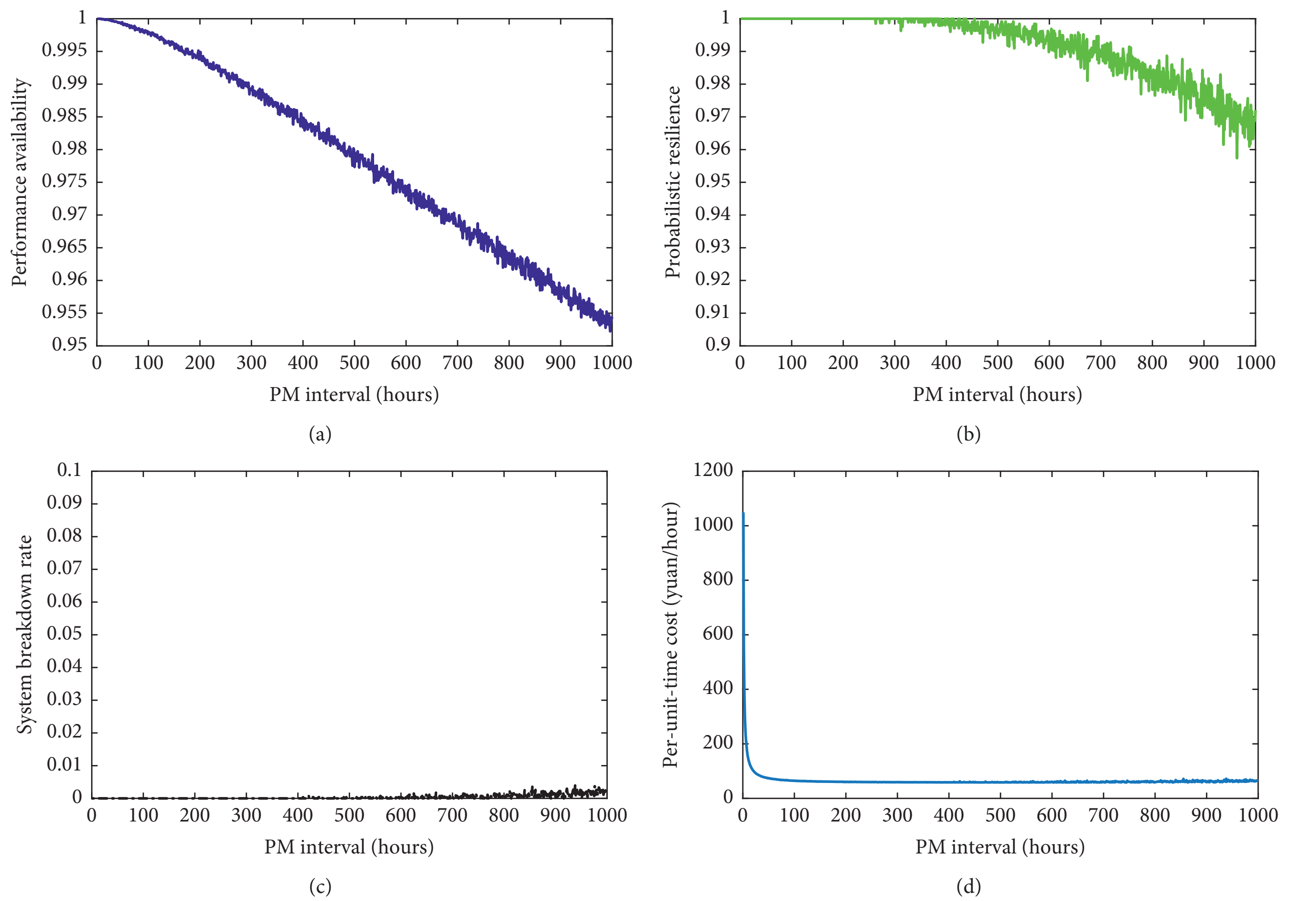

Figure 5: The four decision elements of the system under different PM intervals. (a) Performance availability. (b) Probabilistic resilience. (c) System breakdown rate. (d) Per-unit-time cost (yuan/hour).

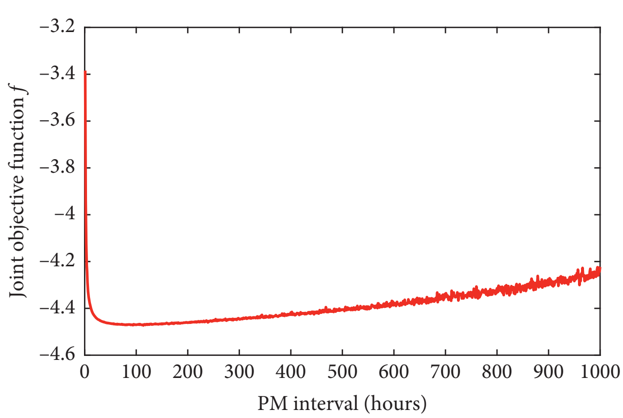

Figure 6: The joint objective function under different PM intervals.

5. Conclusions

In this paper, we provide a general PM interval optimization framework for continuous MSS. In order to analyse the continuous change behaviour of the system's performance, two types of performance-based measures, performance availability and probabilistic resilience, are proposed. Monte Carlo-based simulation is used to obtain the performance data of the system. The optimization model and algorithm proposed in this paper are flexible and can be adjusted according to users' requirements. A case of computer cluster is applied to demonstrate the use of our method, and a multiobjective optimization model is built and solved by both golden section method and genetic algorithm. Comparing with the enumeration results, the effectiveness of our optimization method is verified.

In practice, before the system is put into use, predicted data (including the probability of disturbance occurrence, performance degradation function, recovery function, etc.) can be used to determine the optimal PM interval. During operation, operational data of the system can be combined to find more appropriate PM interval. It should be noted that the decision elements in the PM interval optimization model and the model itself are different for BSS, discrete MSS, and continuous MSS. The decision elements, model, and algorithm in the PM interval optimization problem should be determined according to the system characteristics. If inappropriate ones are applied, the result obtained is likely to deviate from the optimal value.

In the future, we will further study the optimal PM interval for heterogeneous system with continuous multi-state components.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (61773044), National key Laboratory of Science and Technology on Reliability and Environmental Engineering (WDZC2019601A301), and National Natural Science Foundation of China (61872018).

References

[1] M. W. Wakjira and A. P. Ajit, "Total productive maintenance: a case study in manufacturing industry," *Global Journal of Research in Engineering*, vol. 12, no. 1, pp. 25–32, 2012.

[2] D. G. Nguyen and D. N. P. Murthy, "Optimal preventive maintenance policies for repairable systems," *Operations Research*, vol. 29, no. 6, pp. 1181–1194, 1981.

[3] B. L. Kralj and R. Petrović, "Optimal preventive maintenance scheduling of thermal generating units in power systems—a survey of problem formulations and solution methods," *European Journal of Operational Research*, vol. 35, no. 1, pp. 1–15, 1988.

[4] D. Sherwin, "A note on block and bad-as-old renewal of components with limited system time horizon," *Journal of Quality in Maintenance Engineering*, vol. 3, no. 2, pp. 143–147, 1997.

[5] R. Dekker, R. E. Wildeman, and F. A. van der Duyn Schouten, "A review of multi-component maintenance models with economic dependence," *Mathematical Methods of Operations Research*, vol. 45, no. 3, pp. 411–435, 1997.

[6] X. Zheng and N. Fard, "A maintenance policy for repairable systems based on opportunistic failure-rate tolerance," *IEEE Transactions on Reliability*, vol. 40, no. 2, pp. 237–244, 1991.

[7] Y. Gu and L. Jing, "Multi-state system reliability: a new and systematic review," *Procedia Engineering*, vol. 29, no. 4, pp. 531–536, 2012.

[8] G. Levitin and A. Lisnianski, "Joint redundancy and maintenance optimization for multistate series-parallel systems," *Reliability Engineering and System Safety*, vol. 64, no. 1, pp. 33–42, 1999.

[9] G. Levitin and A. Lisnianski, "Optimization of imperfect preventive maintenance for multi-state systems," *Reliability Engineering & System Safety*, vol. 67, no. 2, pp. 193–203, 1999.

[10] I. W. Soro, M. Nourelfath, and D. Aït-Kadi, "Performance evaluation of multi-state degraded systems with minimal repairs and imperfect preventive maintenance," *Reliability Engineering & System Safety*, vol. 95, no. 2, pp. 65–69, 2010.

[11] Y. Liu and H. Z. Huang, "Optimal selective maintenance strategy for multi-state systems under imperfect maintenance," *IEEE Transactions on Reliability*, vol. 59, no. 2, pp. 356–367, 2010.

[12] B. Liu, Z. Xu, M. Xie, and W. Kuo, "A value-based preventive maintenance policy for multi-component system with continuously degrading components," *Reliability Engineering & System Safety*, vol. 132, pp. 83–89, 2014.

[13] R. D. Brunelle and K. C. Kapur, "Continuous-state system-reliability: an interpolation approach," *IEEE Transactions on Reliability*, vol. 47, no. 2, pp. 181–187, 1998.

[14] R. Li, Q. Dong, C. Jin, and R. Kang, "A new resilience measure for supply chain networks," *Sustainability*, vol. 9, no. 1, p. 144, 2017.

[15] X. Zhang and R. Li, "Preventive maintenance interval analysis for multi-state systems with consideration of system resilience," in *Proceedings of the 48th International Conference on Computers and Industrial Engineering*, Auckland, New Zealand, December 2018.

[16] G. P. Cimellaro, A. M. Reinhorn, and M. Bruneau, "Framework for analytical quantification of disaster resilience," *Engineering Structures*, vol. 32, no. 11, pp. 3639–3649, 2010.

[17] V. Crk, "Reliability assessment from degradation data," in *Proceedings of the Annual Reliability and Maintainability Symposium*, pp. 1–7, Los Angeles, CA, USA, January 2000.

[18] M. Rezaei-Malek, R. Tavakkoli-Moghaddam, A. Siadat, and J.-Y. Dantan, "A novel model for the integrated planning of part quality inspection and preventive maintenance in a linear-deteriorating serial multi-stage manufacturing system," *The International Journal of Advanced Manufacturing Technology*, vol. 96, no. 9–12, pp. 3633–3650, 2018.

[19] R. K. Reddy Guduru, S. H. Shaik, S. Yaramala, N. Prakash, and A. Domeika, "A dynamic optimization model for multi objective maintenance of sewing machine," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 20, pp. 33–43, 2018.

[20] M. Alimian, M. Saidi-Mehrabad, and A. Jabbarzadeh, "A robust integrated production and preventive maintenance planning model for multi-state systems with uncertain demand and common cause failures," *Journal of Manufacturing Systems*, vol. 50, pp. 263–277, 2019.

[21] I. Maatouk, I. Jarkass, E. Châtelet, and N. Chebbo, "Preventive maintenance optimization and comparison of genetic algorithm models in a series-parallel multi-state system," *Journal of Intelligent Systems*, vol. 28, no. 2, pp. 219–230, 2019.

[22] C.-H. Wang and T.-W. Lin, "Improved particle swarm optimization to minimize periodic preventive maintenance cost for series-parallel systems," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8963–8969, 2011.

[23] M.-C. Fitouhi and M. Nourelfath, "Integrating noncyclical preventive maintenance scheduling and production planning for multi-state systems," *Reliability Engineering & System Safety*, vol. 121, pp. 175–186, 2014.

[24] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.