ENGINEERING Management
https://doi.org/10.1007/s42524-026-5185-4

RESEARCH ARTICLE

Yian WEI, Sangqi ZHAO, Yao CHENG

# An optimal bi-level inspection and maintenance policy for a multi-component system: An enhanced successively approximated point-based value-iteration algorithm



**Abstract** Modern engineered systems are composed of multiple components. These components deteriorate over time, resulting in a decrease in the system's overall performance. Inspecting the health states of all components provides comprehensive information for making optimal maintenance decisions, which, however, may incur high costs and result in prolonged system downtime. This necessitates combining system-level and component-level inspections into an optimal inspection and maintenance (IM) policy design to maximize overall profit. To date, this topic remains challenging and underexplored, which is investigated in this paper. First, we propose a bi-level IM policy where, at each decision epoch, the operator sequentially determines (i) whether to conduct a component-level inspection and (ii) which components to maintain, based on the system-level performance. Next, we consider that components have different deterioration processes and are subject to non-negligible IM duration, which renders the problem both partially observable and non-equidistant in decision timing. We adopt a Partially Observable Semi-Markov Decision Process (POSMDP) to model the decision-making process and compute the POSMDP quantities. To address the curse of dimensionality posed by the exponential growth of the belief space with respect to the number of components and their discrete states, we develop an enhanced Successively Approximated Point-Based Value-Iteration Algorithm (SARSOP). Two improvements make the algorithm scalable and efficient. First, we introduce macro-actions that enable the operator to defer intervention until system-level performance falls below a predetermined threshold, thereby reducing the depth of the SARSOP search tree by aggregating multiple primitive actions in the original POSMDP. Second, we propose a reward-shaping scheme that encourages SARSOP to prioritize exploration of these macro-actions. A comprehensive case study of photovoltaic (PV) panels demonstrates both the superiorities of the proposed bi-level IM policy and the developed solution algorithm.

**Keywords** bi-level IM policy, partially observable semi-Markov decision process, macro-actions, enhanced SARSOP algorithm

## 1 Introduction

Modern engineered systems are essential for the normal operation of society. These systems typically consist of multiple components whose states jointly determine overall system performance. As components deteriorate, the system's performance deteriorates accordingly. Therefore, timely and effective inspection and maintenance (IM) actions for these components are crucial for guaranteeing the system's performance. In practice, measuring the system's output performance (defined as "system-level inspection") is more technically and economically practical than obtaining detailed information on individual components' states (defined as "component-level inspection"), as the latter usually incurs additional time and costs. However, a system-level inspection fails to provide comprehensive information for making subsequent maintenance-related decisions, especially when these decisions could be made for individual components separately to achieve low maintenance-associated cost. For instance, the power-generating capability of a string of photovoltaic (PV) panels can be directly monitored by an inverter-inte-

Received Jul. 9, 2025; revised Dec. 9, 2025; accepted Dec. 25, 2025

Yian WEI
City University of Hong Kong (Dongguan), Dongguan 523808, China; City University of Hong Kong, Tat Chee Avenue, Hong Kong SAR 999077, China

Sangqi ZHAO, Yao CHENG (✉)
Department of Data and Systems Engineering, The University of Hong Kong, Hong Kong SAR 999077, China
E-mail: yaocheng@hku.hk

This work was supported by the Research Grant Council, Hong Kong SAR, China (No. 17200324).

grated string power meter, which provides the aggregate output power at negligible cost. In contrast, assessing the power-generating capability of each individual panel typically requires temporarily isolating the module and performing an I-V curve test with portable measuring equipment, which incurs additional labor costs and non-negligible inspection durations. This consideration motivates the operator to include both "system-level inspection " and "component-level inspection" to design an optimal bi-level IM policy that decides the optimal timing for these two types of inspections, and thereby, the maintenance decisions. This requires us to resolve the following challenges.

The first challenge lies in making inferences about component states based on observed system performance due to the components' heterogeneous state deterioration processes and their complex types of dependencies. This naturally gives rise to the challenge in deciding component-level inspection decisions, which is a quite implicit decision problem. Second, economic dependency is commonly observed among components, making it challenging to make joint component-level maintenance decisions post-inspection for all components, especially when the components have different maintenance durations and costs. Third, the number of components in the system is typically large, and each component usually has a large number of health states. Under these circumstances, the size of the belief space expands exponentially, rendering the derivation of an optimal bi-level inspection and maintenance policy computationally challenging.

While existing research has devoted substantial effort to designing optimal IM policies for multi-component systems, some studies assume that component-level inspections are conducted at every decision epoch to reveal the states of all components. This practice may be unnecessary (e.g., when the system's overall performance is satisfactory, such that the operators believe the components are in good health conditions) and incur high costs. Therefore, a gap remains in developing a bi-level inspection policy that allows an operator to skip component-level inspections at certain decision epochs to enhance overall system profitability. Moreover, once component-level inspection is not a by-default action, inspection decisions must be made under partially observable information, rendering the widely adopted Markov Decision Process (MDP) framework methodologically inadequate. Considering the non-negligible durations of inspection and maintenance activities, a Partially Observable Semi-Markov Decision Process (POSMDP) framework is necessary to characterize this decision-making problem accurately. To the best of our knowledge, this remains an underexplored area. Moreover, in the existing MDP-based research on maintenance policy optimization, most approaches rely on monolithic batch backup algorithms (e.g., value iteration) to obtain optimal solutions to the policy. Such methods often suffer severely from the curse of dimensionality when scaled to multi-component systems with large state spaces, intricate dependencies, and partial observations. This calls for more efficient solution techniques and algorithmic enhancements.

In this study, we address these gaps by proposing and solving a novel bi-level IM policy for a multi-component system when component-level inspections are inconvenient or uneconomical. Our contributions can be summarized as follows:

• We propose a bi-level IM policy where, at each decision epoch, the system-level performance can be observed, and the operator must sequentially determine (i) whether or not to conduct component-level inspections and (ii) which components to maintain. The proposed policy outperforms existing benchmarks.

• We develop a POSMDP to characterize the non-equidistant decision-making process. Next, we derive key POSMDP quantities, including the expected duration between decision epochs, state transition probabilities, and expected rewards for state-action pairs. We also formulate a belief update procedure based on both system-level and component-level inspection results. The proposed POSMDP can be extended to cases where the system has a general coherent structure, and the inspection and the maintenance are imperfect.

• We propose an enhanced Successively Approximated Point-Based Value-Iteration (SARSOP) algorithm that efficiently solves the resulting high-dimensional POSMDP. Specifically, we introduce macro-actions that allow the operator to take no action until the system performance deteriorates to a predetermined threshold. These macro-actions significantly reduce the depth of the decision tree in SARSOP, thereby accelerating the convergence. Furthermore, we employ a reward-shaping scheme to incentivize the exploration of these macro-actions. By a detailed case study of PV panels, we demonstrate the superiority of the proposed IM policy and the enhanced SARSOP algorithm over existing alternatives.

The remainder of this study is organized as follows. In Section 2, we review existing research on optimal IM policy design for multi-component systems. Section 3 presents the problem statement. In Section 4, we formulate the POSMDP, derive its key quantities and the belief-updating process. Section 5 develops an enhanced SARSOP algorithm. Section 6 demonstrates the superiority of the proposed bi-level policy and the enhanced SARSOP algorithm through a case study of photovoltaic panels. Section 7 presents our conclusions and outlines directions for future research.

## 2 Literature review

In many existing studies on multi-component system IM policy optimization, it is by-default assumed that at each

decision epoch, a component-level inspection is performed to reveal all components' states. Subsequently, based on the inspection results, maintenance decisions are made for components with stochastic, economic, or structural dependencies (Dui et al., 2024, Xiao et al., 2020, Zheng and Zhou, 2022, Sun et al., 2020, Hu et al., 2021). Among them, Zheng and Zhou (2022) optimized a dynamic inspection and replacement policy for a production system composed of two nonidentical units with dependent deterioration processes and replacement durations. Wei et al. (Wei and Cheng, 2025, Wei et al., 2025a, Wei et al., 2025b) optimized a multi-dimensional maintenance policy for a fleet of self-service systems under imperfect monitoring or maintenance actions. Vu et al. (2018) optimized a proactive group-maintenance policy for a multi-component coherent system whose maintenance duration is non-negligible. Zheng et al. (2023) proposed a bi-level preventive maintenance policy based on a criticality analysis of the components. At the component level, each component is replaced after a fixed number of preventive maintenance actions, and these actions are then grouped to form a system-level group maintenance policy. Other similar studies include the optimal maintenance policy design for system with two balanced components (Zhao and Wang, 2022, Fang et al., 2025, Wang et al., 2020a), mission-critical systems (Wang et al., 2021, Qiu et al., 2025, Qiu et al., 2022, Zheng et al., 2026), *k*-out-of-*n* systems (Song and Liang, 2025, Wang et al., 2024, Wang et al., 2022a, Xiahou et al., 2023).

Several studies used sampling to assess these components or systems' states for making maintenance-related decisions, where the operator first infers the system state from the sampling results and determines maintenance schedules. For example, Chen et al. (2025) proposed a deep-learning-enhanced active sampling method that trains an neural network surrogate on adaptively selected evidential samples to approximate system responses and extremum calculations, thereby enabling accurate and efficient propagation of evidential uncertainty with far fewer expensive model evaluations. Cheng et al. (2022) investigated a sampling-based sequential IM policy in which the components for sampling follow different lifetime distributions and the sampling times are non-equidistant. Lv et al. (2024) designed the maintenance policy for a production system whose products are periodically sampled and monitored; a preventative maintenance action is triggered if no alarm is made in *k* consecutive inspections and otherwise, a corrective maintenance action is undertaken. A Bayesian control policy is proposed in (Naderkhani and Makis, 2015), where the system state can be sampled at two distinct intervals. In (Yang et al., 2024a, Tan et al., 2025), the component's health state is predicted (or prognosed) by results of different types of inspections and the maintenance-related decision is optimized based on these results. More similar studies can be found in (Yang et al., 2024b, Bouslah et al., 2016, Wang et al., 2023, Zhao et al., 2025b, Qiu et al., 2021).

For a multi-component system, performing inspection on individual components can be both time-consuming and expensive when compared to only inspecting the system's performance due to the complex disassembly and reassembly procedures (Dinh et al., 2022, Zhou et al., 2015). Therefore, the widely adopted strategy in the above studies that always perform a component-level inspection (or sampling) at every decision epoch, may be suboptimal. A better alternative is to perform component-level inspection when the system's overall performance (e.g., functional status for binary systems, production rate for manufacturing systems, transmission capacity for transport systems, etc.) is undesirable (Hu and Sun, 2026), since the system overall performance can typically be observed at a much lower cost than revealing the states of all components. Therefore, a more general IM policy that enables the operator to make bi-level inspection decisions, e.g., to decide whether to perform a component-level inspection based on the observed system output performance, must be superior.

However, to date, only a limited number of studies have incorporated different inspection levels into IM-policy optimization, and most of them focus on single-component systems. For instance, Liu et al. (2021) analyzed a four-state system and a bi-level inspection policy in which a major inspection can reveal all defective states of the system, whereas a minor inspection can expose only a subset of them. Wang (2000) developed a model of multiple nested inspections performed at different intervals for a production plant and determined the optimal inspection intervals by means of a branch-and-bound algorithm. Mahmoudi et al. (2017) adopted delay-time models to deal with multiple inspection methods and multiple defect types for a generic system. Xiao et al. (Xiao et al., 2023, Kou et al., 2023) leveraged the production system's idle waiting time and, optimized a non-periodic inspection-maintenance strategy that minimizes long-run maintenance costs considering multiple failure modes. Other related studies that consider such multi-type inspections with varying costs and detection capabilities include (Nguyen et al., 2019, Papakonstantinou et al., 2018, Hao et al., 2020, Yang et al., 2023).

When several inspection levels are mixed for a multi-component system and component-level inspections are skipped in some decision epochs, the states of all components cannot always be perfectly observed. Few studies have addressed bi-level inspection in the context of multi-component systems. Liu et al. (2020) allowed, at each decision epoch, maintenance to be executed on any subset of all components and proposed a metric to quantify the effectiveness of a given bi-level inspection strategy; a tailored ant colony optimization algorithm is adopted to derive the optimal multi-level IM policy. This work is

subsequently generalized to (Zhang et al., 2024a). Wei et al. (2025c) examined a bi-level IM policy for a system containing a protection component, where inspections can be performed either on the entire system or only on the protection component. However, we note that in practical engineering, due to the non-negligible IM duration, the adoption of POSMDP rather than POMDP becomes necessary (Tang et al., 2024), whose state space grows exponentially with the number of components and their possible states. Furthermore, monolithic batch backup algorithms, such as value iteration or point-based value iteration (Qiu et al., 2020, Sun et al., 2025, Khaleghei and Kim, 2021), are often employed. For example, Sun et al. (2025) formulated the mission abort decision for a mission-critical system under imperfect condition monitoring as a POMDP, and used a modified point-based value iteration algorithm to efficiently compute near-optimal abort policies. Some studies employ deep reinforcement learning to cope with such a large state space (Zhang et al., 2024b, Chen et al., 2022). Among them, Zhang et al. (2021) develop a Q-learning reinforcement learning approach for maintenance optimization. Zhao et al. (Zhao et al., 2025a, Zhao et al., 2026) use a D3QN network to optimize the maintenance policy of a k-out-of-n load-sharing system, and then generalize it to a series *k*-out-of-*n* load-sharing system by adopting a multi-agent RL network to address the resulting huge state space. Chen et al. (2024) propose a dynamic inspection and maintenance policy for multi-state systems under time-varying demands and use Proximal Policy Optimization to solve it. Existing DRL-based maintenance studies largely demonstrate that deep policies can handle larger state-action spaces than traditional methods; however, their black-box structures still result in a lack of interpretability of the obtained solutions and potential convergence issues, thereby leaving a gap for a state-space-friendly, interpretable, model-driven planning method.

## 3 Problem statement

We consider a multi-component system that continuously generates revenue during its operation, and the revenue is determined by its performance level. The system consists of $n$ multi-state components. Throughout the system's operation, the components are subject to deterioration and inspection and maintenance are performed. The assumptions associated with the states of the components, along with the inspection and maintenance, are characterized below.

1) The state of component $i$ is denoted as $s_i$, which is an integer ranging from 0 to $S_i$ with 0 representing the worst state and $S_i$ representing the best state. The uncontrolled state transition rate matrix of component $i$ is defined as $\boldsymbol{Q}_i$.

2) The performance of a component is defined as its state.

3) The performance of the system is determined by the minimum performance of all components. When the system performance is $z$, it generates a revenue of $v_z$ per time unit.

4) The deterioration processes and sudden failures of the components are characterized by their state transition probability matrices.

5) Failures of the components are non-self-announcing.

6) Two types of inspections can be implemented: *system-level inspections* that reveal the system performance, and *component-level inspections* that reveal the states of system. Decision epochs are defined as the instants immediately after the completion of system-level inspections or component-level inspections.

7) After a system-level inspection, the operator must choose one of the following actions:

a) Do nothing and wait for the next system-level inspection after $\tau$ time units.

b) Perform a component-level inspection.

8) After a component-level inspection, the operator must choose one of the following actions:

a) Do nothing and wait for the next system-level inspection after $\tau$ time units.

b) Maintain some components and wait for the next system-level inspection after $\tau$ time units since the completion of maintenance action.

9) Components are restored to an as-good-as-new condition after maintenance.

10) The duration of a system-level inspection is negligible. The duration of a component-level inspection is $\tau_o$ and the duration of a maintenance action for component $i$ is $\tau_i^m$. During a component-level inspection or maintenance, the system does not generate revenue.

11) The cost of conducting a component-level inspection for the entire system is $c_o$ and the maintenance cost for component $i$ is $c_i$.

Note that in this paper, for clarity of exposition we model the system performance as the minimum of the component performance levels (assumption 1). This setting is representative of series-connected PV strings and bottleneck-type production systems. Nevertheless, as discussed in Remarks 2, the framework extends directly to general coherent systems by replacing the "minimum" aggregation with the corresponding coherent structure function.

A schematic representation of a possible sequence of operator decisions and the corresponding transitions in component states is illustrated in Fig. 1, where DN, CI, M1 and M2 respectively represents "doing nothing," "component-level inspection," "maintaining component 1" and "maintaining component 2."

The operator's objective is to determine the optimal IM policy to maximize the system's discounted long-run profit. Considering that after a system-level inspection, only system performance is observed, whereas the exact

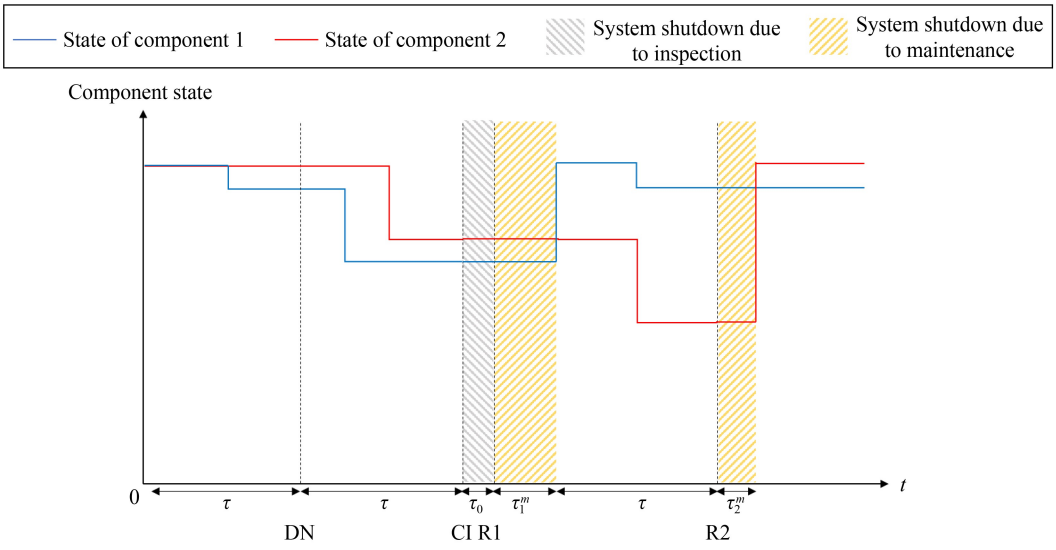


**Fig. 1** A schematic of operator's IM decisions and the components' state transition processes.

component state vector remains hidden. The operator must infer the component states from the observed system output and historical information. This partial observability motivates the use of belief states and motivates us to formulate the decision problem as a POSMDP. In what follows, we model the POSMDP and customize an enhanced SARSOP algorithm to solve it.

## 4 POSMDP modeling

POSMDPs offer a structured framework for modeling non-equidistant decision-making problems whose uncertainties arise due to partially observable system state. In such models, the decision-makers update and refine their beliefs about the system state by including new information observed at each decision epoch. The belief about system states observed at decision epochs forms a semi-Markov chain, where the sojourn time between two successive decision epochs depends on the chosen action.

A POSMDP model can be defined by a seven-tuple consisting of $\mathcal{S}$, $\mathcal{A}$, $\mathcal{U}$, $\mathcal{Z}$, $\mathcal{T}$, $O$, $\mathcal{R}$, where these seven elements respectively represent the state space, action space, interval between two decision-making epochs, set of observations, state transition probabilities, observation probabilities, and reward function. In what follows, we model the POSMDP based on this seven-tuple and investigate the belief updating process.

### 4.1 Characterization of the POSMDP and its quantities

State space $\mathcal{S}$

The system state is defined as the collection of all components' states. Therefore, the dimension of the state space $\mathcal{S}$ is $\prod_{i=1}^{n}(S_i+1)$ and is presented as follows:

$$\mathcal{S}=\prod_{i=1}^{n}\times \boldsymbol{S}_i,\ \mathbf{S}_i=\{0,1,\ldots,S_i\}, \tag{1}$$

where $\times$ denotes the Cartesian product operator. We define a belief $\boldsymbol{b}$ as a probability distribution on the state space $\mathcal{S}$, that is:

$$\boldsymbol{b}\colon \mathcal{S}\to[0,1],\ \text{with}\ \textstyle\sum_{s\in\mathcal{S}}\boldsymbol{b}(s)=1. \tag{2}$$

All possible beliefs $\boldsymbol{b}$ form the belief space $\boldsymbol{B}$. Considering that the operators know the system state after a component-level inspection, which is reflected in their belief that there exists a system state $s\in\mathcal{S}$ such that $\boldsymbol{b}(s)=1$ and $\boldsymbol{b}(s')=0$ for all $s'\neq s$. We define set $\boldsymbol{B}_p$ as a subset of $\boldsymbol{B}$ such that:

$$\boldsymbol{B}_p=\{\boldsymbol{b}\in\boldsymbol{B}|\exists s\in\mathcal{S},\ \boldsymbol{b}(s)=1\ \text{and}\ \mathbf{b}(s')=0,\ \forall s'\neq s\}. \tag{3}$$

where the set $\boldsymbol{B}_p^c$ is defined as the complement set of $\boldsymbol{B}_p$ to $\boldsymbol{B}$.

Action space $\mathcal{A}$

The action space varies with the system state. Specifically, at each decision epoch, the operator can choose to do nothing (DN), perform a component-level inspection (CI), or maintain some components. But after each component-level inspection, the operator can only choose between DN or maintaining some components.

Note that after a system-level inspection, although the performance of the system is revealed, the system state (i.e., all components' states) is not fully determined, i.e., $\boldsymbol{b}\in\boldsymbol{B}_p^c$; whereas after a component-level inspection, the system state is determined (i.e., $\boldsymbol{b}\in\boldsymbol{B}_p$). Therefore, we determine whether the current decision epoch is after a system-level inspection or after a component-level inspection based on whether $\boldsymbol{b}\in\boldsymbol{B}_p$, and we write $\mathcal{A}$ as a function of $\boldsymbol{b}$:

$$\mathcal{A}(b)=\begin{cases}\{\mathrm{DN},\mathrm{CI}\}, & \text{if } \boldsymbol{b}\in\boldsymbol{B}_p^c\\ \{\mathrm{DN},\mathbf{M}\}, & \text{if } \boldsymbol{b}\in\boldsymbol{B}_p\end{cases}, \tag{4}$$

where $\boldsymbol{M}$ is the action set of maintaining some components and is expressed as:

$$\boldsymbol{M}=\left\{m=(m_1,m_2,\ldots,m_n)\in\{0,1\}^n\,\middle|\,\sum_{i=1}^n m_i\geqslant 1\right\}. \tag{5}$$

In this definition, $m_i=1$ indicates that component $i$ is to be maintained, and $m_i=0$ indicates that component $i$ is not maintained. The condition $\sum_{i=1}^n m_i\geqslant 1$ ensures that at least one component is being maintained, excluding the action where no components are maintained (which corresponds to the DN action).

**Remark 1.** In the present model, a component-level inspection is assumed to reveal the states of all components simultaneously, which is appropriate when inspections are scheduled jointly and the marginal cost of inspecting additional components during a visit is relatively small. Allowing CI to be selectively applied to only a subset of components at each decision epoch would introduce an additional combinatorial layer in the action space and is therefore left as an interesting extension for future research.

Interval between two decision epochs $\mathcal{U}$

The time until the next decision epoch $\mathcal{U}$ depends on the chosen action. Specifically, when the operator chooses "DN," the next decision epoch will occur after $\tau$ time units. If the operator chooses "CI," then the next decision epoch will occur at the end of the component-level inspection. If the operator decides to maintain some components (i.e., chooses an action $m\in\boldsymbol{M}$), the next decision epoch will be delayed by the duration of the inspection or by the maximum maintenance duration of all the components that are maintained. Therefore, we write $\mathcal{U}$ as a function of the action $a\in\mathcal{A}$:

$$\mathcal{U}(a)=\begin{cases}\tau, & \text{if } a=\mathrm{DN}\\ \tau_{\mathrm{o}}, & \text{if } a=\mathrm{CI}\\ \tau+\max_{i,m_i=1}\tau_i^m, & \text{if } a=\boldsymbol{m}\in\boldsymbol{M}\end{cases}. \tag{6}$$

Set of observations $\mathcal{Z}$

It is straightforward that the observation is either regarding the system's performance (after a system-level inspection) or regarding all components' states (after a component-level inspection). Therefore, the set of observations can be written as the union of all possible system performance levels and all possible combinations of component states, as follows:

$$\mathcal{Z}=\boldsymbol{S}^y\cup\mathcal{S},\ \text{with}\ \mathbf{S}^y=\{0,1,\ldots,\min_{i\in\{1,\ldots,n\}}S_i\}, \tag{7}$$

where $\min_{i\in\{1,\ldots,n\}}S_i$ represents the maximum system performance level.

State transition probabilities $\mathcal{T}$

We define $\mathcal{T}$ in the form of $\mathcal{T}(s,a,s'),s,s'\in\mathcal{S}$, which denotes the probability that if action $a$ is adopted when the system state is $s$, then at the next decision epoch, the system state is $s'$. For notational simplicity, in this section, the state of component $i$ in system state $s$ is denoted as $s_i$ and that in system state $s'$ is denoted as $s_i'$. We discuss the state transition behaviors when different actions are chosen as follows.

(i) When action CI is chosen:

During the period until the next decision epoch (i.e., until the component-level inspection is completed), the component states remain unchanged. Therefore, we write $\mathcal{T}(s,\mathrm{CI},s')$ as:

$$\mathcal{T}(s,\mathrm{CI},s')=\begin{cases}1, & \text{if } s'=s\\ 0, & \text{if } s'\neq s\end{cases}. \tag{8}$$

(ii) When action DN is chosen:

All components experience deterioration or sudden failure until the next system-level inspection that is performed after $\tau$ time units. Therefore, the state transition probability $\mathcal{T}(s,\mathrm{DN},s')$ can be obtained as:

$$\mathcal{T}(s,\mathrm{DN},s')=\prod_{i=1}^n\left(\alpha_{s_i}\mathrm{e}^{\boldsymbol{Q}_i\tau}\alpha_{s_i'}^T\right), \tag{9}$$

where $\alpha_{s_i}$ is an $(S_i+1)\times 1$ initial state distribution vector of component $i$; it has its $(S_i+1)^{\text{th}}$ element being 1 and other elements being 0.

(iii) When action $m\in\boldsymbol{M}$ is chosen:

Components $i$ with $m_i=1$ are maintained, after that, all components experience deterioration or sudden failure over a period of time $\tau$ until the next system-level inspection. The state transition probability $\mathcal{T}(s,m,s')$ can be obtained as:

$$\mathcal{T}(s,m,s')=\prod_{i=1}^n\left(\left[\alpha_{s_i}(1-m_i)+\alpha_{S_i}m_i\right]\mathrm{e}^{\boldsymbol{Q}_i\tau}\alpha_{s_i'}^T\right). \tag{10}$$

Observation probabilities $O$

We define $O$ in the form of $O(s',a,z)$, $s'\in\mathcal{S}$, $z\in\mathcal{Z}$, which represents the probability of observing outcome $z$ given that action. $a$. has been performed before the system is in state $s'$. When $a=\mathrm{CI}$, the observation $z$ must be in the set $\mathcal{S}$, where all components' states are revealed; otherwise, the observation $z$ is in the set $\mathcal{S}^y$, where only the system performance is revealed. Based on Eqs. (9) and (10), we have:

$$O(s',\mathrm{CI},z)=\begin{cases}1, & \text{if } z=s'\\ 0, & \text{otherwise}\end{cases}, \tag{11}$$

$$O(s',a,z|a\in\boldsymbol{M}\cup\{\mathrm{DN}\})=\begin{cases}1, & \text{if } z=\min_{i=1}^n s_i'\\ 0, & \text{otherwise}\end{cases}. \tag{12}$$

**Remark 2.** In Section 3 we assume that the system's performance is equal to the minimum of the component states. In fact, to extend the model developed in this

paper to a generic coherent multi-component system (Wei et al., 2025d, Hashemi and Asadi, 2021), one need only replace the term $\min_{i=1}^{n} s_i'$ in Eq. (12) with the specific functional relationship between the system's output performance and the component states.

Reward function $\mathcal{R}$

The reward function $\mathcal{R}$ can be written in the form of $\mathcal{R}(s,a)$, where $s \in \mathcal{S}$ and $a \in \mathcal{A}$, which indicates the expected reward incurred when action $a$ is chosen when the system state is $s$, until the next decision epoch. It includes two parts: the inspection and maintenance-induced cost (if CI or $m \in \boldsymbol{M}$ is chosen) and the revenue yielded by the system (if DN or $m \in \boldsymbol{M}$ is chosen). Therefore, we write it as:

$$\mathcal{R}(s,\mathrm{CI}) = -c_o, \tag{13}$$

$$\mathcal{R}(s,\mathrm{DN}) = \int_0^{\tau} \left( \sum_{s' \in \mathcal{S}:\, \min_{i=1}^{n} s_i' = z} v_z \underbrace{\prod_{i=1}^{n} \alpha_{s_i} \mathrm{e}^{\boldsymbol{Q}_i t} \alpha_{s_i'}^{T}}_{A} \right) \mathrm{d}t, \tag{14}$$

$$\mathcal{R}(s,m) = -\sum_{i=1}^{n} r_i c_i + \int_0^{\tau} \left( \sum_{s' \in \mathcal{S}:\, \min_{i=1}^{n} s_i' = z} v_z \underbrace{\prod_{i=1}^{n} \left[ \alpha_{s_i}(1-m_i) + \alpha_{S_i} m_i \right] \mathrm{e}^{\boldsymbol{Q}_i t} \alpha_{s_i'}^{T}}_{B} \right) \mathrm{d}t, \tag{15}$$

where the term A in Eq. (14) and term B in Eq. (15) denote the probabilities that the system is in state $s'$ at time $t$ when DN and $m \in \boldsymbol{M}$ is chosen, respectively. Note that if a setup cost $c_s$ (Sun et al., 2018) for maintenance actions needs to be considered, it can be incorporated by adding a fixed cost term $c_s$ to the reward function $\mathcal{R}(s,m)$.

## 4.2 Belief updating process

According to the observations made before each decision epoch (either after a system-level inspection or a component-level inspection), which include either the system output performance or the states of all components, the decision-makers update their beliefs about the states of the components. Let $\boldsymbol{b}$ denote the belief about the components' states at a given decision epoch. If an observation $z$ is obtained after action $a$ is chosen, then the updated belief $\boldsymbol{b}'$ at the next decision epoch can be derived as:

$$\boldsymbol{b}'(s'|\boldsymbol{b},a,z) = \frac{O(s',a,z)\sum_{s \in \mathcal{S}} \boldsymbol{b}(s) \mathcal{T}(s,a,s')}{\sum_{s' \in \mathcal{S}} O(s',a,z) \sum_{s \in \mathcal{S}} \boldsymbol{b}(s) \mathcal{T}(s,a,s')}. \tag{16}$$

The explicit forms of terms in Eq. (16) are given in Eqs. (6)–(15).

## 4.3 Bellman equations

The value function of state $\boldsymbol{b}$ is defined as $V(\boldsymbol{b})$. Considering the non-negligible durations of inspection and maintenance actions, the Bellman equations (Bellman, 1966) of the proposed POSMDP can be written as follows:

$$V^*(\boldsymbol{b}) = \max_{a \in \mathcal{A}} \left[ \mathcal{R}(\boldsymbol{b},a) + \sum_{z \in \mathcal{Z}} O(z|\boldsymbol{b},a) \gamma^{\mathcal{U}(a)} V^*(\boldsymbol{b}'|\boldsymbol{b},a,z) \right], \tag{17}$$

where $\gamma$ is the discount factor, and in Eq. (17) we have:

$$\mathcal{R}(\boldsymbol{b},a) = \sum_{s \in \mathcal{S}} \boldsymbol{b}_s \mathcal{R}(s,a), \tag{18}$$

and

$$O(z|\boldsymbol{b},a) = \sum_{s \in \mathcal{S}} O(s,a,z) \sum_{s' \in \mathcal{S}} \boldsymbol{b}_s \mathcal{T}(s,a,s'). \tag{19}$$

## 4.4 Model extensions

The POSMDP model developed in Sections 4.1–4.3 relies on two simplifying assumptions: (i) component-level inspections are perfect, i.e., a component-level inspection reveals the exact state of every component, and (ii) maintenance actions restore components to an as-good-as-new condition. These assumptions are convenient for exposition and model derivation, but may limit the practical applicability of the model. In fact, the model can be generalized without changing the overall POSMDP structure. Below we provide two extensions regarding how to incorporate imperfect component-level inspections and imperfect maintenance.

### 4.4.1 Imperfect component-level inspections

Component-level inspections may in practice be imperfect, so that the observed condition of a component can deviate from its true state. To capture this, for each component $i$ we introduce an inspection error function $Q_i(\tilde{x}_i|x_i)$ that characterizes the discrepancy between the reported and true states of that component, i.e.,

$$Q_i(\tilde{x}_i|x_i) = P\{\text{reported state of component } i \text{ is } \tilde{x}_i | \text{ true state of component } i \text{ is } x_i\} \tag{20}$$

Then, with imperfect inspection, the observation probability term $O(s',\mathrm{CI},z)$ in Eq. (11) should be revisited as:

$$O(s',\mathrm{CI},z) = \prod_{i=1}^{n} Q(z_i|s_i'). \tag{21}$$

The other parts of the POSMDP formulation remain intact.

### 4.4.2 Imperfect maintenance actions

In practice, maintenance may also be imperfect, so that the post-maintenance state of a component can deviate from the ideal as-good-as-new condition. To capture this effect, for each component $i$ we introduce a maintenance

function $M_i(y_i|x_i)$ that characterizes the discrepancy between the state immediately after maintenance and the pre-maintenance state, i.e.,

$$M_i(y_i|x_i) = P\{\text{state of component } i \text{ is } \tilde{x}_i \text{ after maintenance}|\text{pre-maintenance state is } x_i\} \tag{22}$$

Note that under perfect maintenance, we have $M_i(S_i|x_i) = 1$ and $M_i(y_i|x_i) = 0$ for all $y_i \neq S_i$.

With imperfect maintenance, the state-transition term $\mathcal{T}(s,m,s')$ in Eq. (10) should be revisited as:

$$\mathcal{T}(s,m,s') = \prod_{i=1}^{n} \tilde{P}_i(s_i'|s_i,m_i), \tag{23}$$

where for each component $i$,

$$\tilde{P}_i(s_i'|s_i,m_i) = \begin{cases} \sum_{y_i=0}^{S_i} M_i(y_i|s_i)\alpha_{y_i} e^{\boldsymbol{Q}_i\tau}\alpha_{s_i'}^T, & \text{if } m_i = 1, \\ \alpha_{s_i} e^{\boldsymbol{Q}_i\tau}\alpha_{s_i'}^T, & \text{if } m_i = 0. \end{cases} \tag{24}$$

Similarly, the reward term $\mathcal{R}(s,m)$ in Eq. (15) should be revisited as:

$$\tilde{\mathcal{R}}(s,m) = -\sum_{i=1}^{n} r_i c_i + \int_0^{\tau} \sum_{s' \in \mathcal{S}:\ \min_{i=1}^{n} s_i' = z} v_z \tilde{P}_i(s_i'|s_i,m_i)\,\mathrm{d}t. \tag{25}$$

The other parts of the POSMDP formulation remain intact.

## 5 Solution technique: An enhanced SARSOP algorithm

It should be noted that the entire belief space grows exponentially with the number of components, i.e., when a new component $i$ with $S_i$ states is added, its belief state expands by a factor of $(S_i + 1)$ to incorporate the belief over component $i$'s states. This imposes significant computational difficulty on both the belief-updating and solution processes. In fact, not all beliefs in the belief space $\boldsymbol{B}$ are reachable. For example, for any component $i$, it is unlikely that the belief over the component's state assigns probability 0.5 to its best state $S_i$, 0.5 to its worst state 0, and 0 to other intermediate states. Therefore, excluding such beliefs from the belief space would significantly help reduce the computational complexity. In what follows, we customize the SARSOP algorithm (Kurniawati et al., 2008) to solve the problem by leveraging a comparatively small set of representative belief points.

Due to the inherent complexity of the proposed POSMDP, computing the value function $V(\boldsymbol{b})$ for every belief $\boldsymbol{b}$ is computationally intractable. The SARSOP algorithm addresses this by maintaining an upper bound denoted as $V^{up}(\boldsymbol{b})$ and a lower bound denoted as $V^{lb}(\boldsymbol{b})$, on the true value $V(\boldsymbol{b})$. It then builds and refines a search tree over the belief space iteratively, which contains only a carefully selected subset of beliefs, by point-based backups and pruning until the bounds converge.

In this study, we obtain $V^{up}(\boldsymbol{b})$ and $V^{lb}(\boldsymbol{b})$ from the solutions to two different POSMDPs. In particular, for $V^{up}(\boldsymbol{b})$, we employ the Q-value MDP (QMDP) approximation, which is based on the fully observable SMDP underlying the original POSMDP. For $V^{lb}(\boldsymbol{b})$, we solve a more tractable POSMDP that is clearly inferior to the original model, in which all state components are fully observed at each decision epoch.

### 5.1 Initial upper bound $V^{up}(\boldsymbol{b})$: From a QMDP

It is clear that if all components' states in the system are always observable, the decision-maker always makes at least as good, and generally strictly better, decisions than in the partially observable case. Therefore, we assume that for any belief $\boldsymbol{b}$, the resulting state becomes fully known immediately after action $a$ is chosen. Based on this assumption, we define the QMDP approximation as:

$$V_{\text{QMDP}}(\boldsymbol{b}) = \max_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}} b(s)\, V_{\text{MDP}}^*(s,a), \tag{26}$$

where $V_{\text{MDP}}^*(s,a)$ denotes the optimal action-value function of the underlying fully observable MDP when taking action $a$ in state $s$. Clearly, solving $V_{\text{MDP}}^*(s,a)$ involves a much smaller state space than the original POSMDP and is thus tractable. Moreover, $V_{\text{QMDP}}(\boldsymbol{b})$ provides a valid and tight upper bound on the value function at belief $\boldsymbol{b}$ for the proposed POSMDP.

Specifically, to compute $V_{\text{MDP}}^*(s,a)$, we only need to treat the belief space $\boldsymbol{B}$ as identical to the state space $\mathcal{S}$, and set the observation probability $O(s',a,z)$ in Eqs. (11) and (12) to:

$$O(s',a,z) = \begin{cases} 1, & \text{if } z = s' \\ 0, & \text{otherwise} \end{cases}, \forall a \in \mathcal{A}. \tag{27}$$

### 5.2 Initial lower bound $V^{lb}(\boldsymbol{b})$: From an inferior POSMDP

We obtain a family of inferior POSMDPs by reducing the action space in the proposed POSMDP. It is straightforward that these restricted POSMDPs yield more conservative value functions at every belief $\boldsymbol{b} \in \boldsymbol{B}$. Motivated by this, we construct a POSMDP that employs a threshold-based maintenance-all (TBMA) policy, where the decision-makers never perform component-level inspections and carry out maintenance on all components once the system performance reaches a threshold $d$. This policy is a special case of the proposed POSMDP and produces a more conservative value function at all beliefs. Hence, we may use the value function under this policy at any belief $\boldsymbol{b}$, denoted by $V_{\text{TBMA}}(\boldsymbol{b})$, as a lower bound for $V(\boldsymbol{b})$, i.e., $V^{lb}(\boldsymbol{b}) = V_{\text{TBMA}}(\boldsymbol{b})$. Moreover, this policy drastically reduces the action space, thereby significantly simplifying

the solution process. Specifically, the belief space $\boldsymbol{B}$ should be partitioned into two subsets:

$$\boldsymbol{B}_d = \{\boldsymbol{b} \in \boldsymbol{B}\colon \exists s \in \mathcal{S}, \min_i s_i \leqslant d \text{ and } \mathbf{b}(s) > 0\}, \quad (28)$$

and

$$\boldsymbol{B}_d^c = \{\boldsymbol{b} \in \boldsymbol{B}\colon \boldsymbol{b} \notin \boldsymbol{B}_d\}. \quad (29)$$

In Eq. (28), any belief $\boldsymbol{b} \in \boldsymbol{B}_d$ must assign a positive probability to at least one state $s$ whose system performance is below $d$. This implies that once the system performance reaches $d$, the belief over component states enters $\boldsymbol{B}_d$. Accordingly, to reflect the TBMA policy, we modify the state-specific action sets in Eq. (4) as follows:

$$\mathcal{A}(\boldsymbol{b}) = \begin{cases} \text{DN}, & \text{if } \boldsymbol{b} \in \boldsymbol{B}_d^c, \\ \left(\underbrace{1,1,\ldots,1}_{n}\right), & \text{if } \boldsymbol{b} \in \boldsymbol{B}_d. \end{cases} \quad (30)$$

Then, based on these two modifications, one can obtain $V_{\text{TBMA}}(\boldsymbol{b})$ by solving this inferior POSMDP.

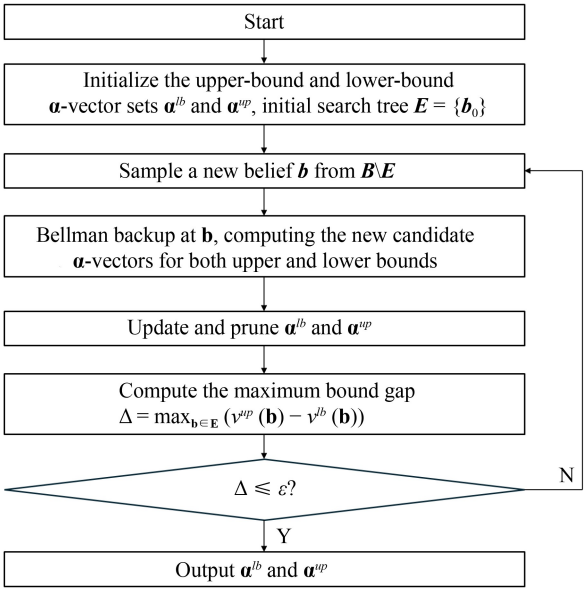


**Fig. 2** Flowchart of the SARSOP algorithm to solve the proposed POSMDP.

## 5.3 The enhanced SARSOP algorithm

In this section, we first elaborate on the application of the standard SARSOP algorithm to solve the POSMDP. Subsequently, leveraging the features of this POSMDP, we introduce macro-actions and reward shaping based on the system performance to accelerate the convergence of the SARSOP algorithm.

### 5.3.1 The standard SARSOP procedure for the proposed POSMDP

The value function of the POSMDP is piecewise-linear and convex, which allows us to describe it using a set of $\boldsymbol{\alpha}$-vectors. For example, the value function $V(\boldsymbol{b})$ at belief $\boldsymbol{b}$ can be written as:

$$V(\boldsymbol{b}) = \max_i \sum_{s\in\mathcal{S}} \boldsymbol{b}(s)\boldsymbol{\alpha}_i(s), \quad (31)$$

where $\boldsymbol{\alpha}_i(s)$ is the component at state $s$ of the $i$-th $\boldsymbol{\alpha}$-vector $\boldsymbol{\alpha}_i$.

Based on these results, we can define the upper bound $V^{up}(\boldsymbol{b})$ and the lower bound $V^{lb}(\boldsymbol{b})$ for any arbitrary belief $\boldsymbol{b} \in \boldsymbol{B}$ via two sets of $\boldsymbol{\alpha}$ -vectors, denoted $\boldsymbol{\alpha}^{up}$ and $\boldsymbol{\alpha}^{lb}$. Through sampling, point-based backups, and pruning, the SARSOP algorithm enables exploration of those beliefs most likely to be encountered in normal operation, and ensures that the gap $V^{up}(\boldsymbol{b}) - V^{lb}(\boldsymbol{b})$ converges. A flowchart is given in Fig. 2.

Next, we detail the sampling, point-based backup, and pruning operations used in this study. First, during sampling, the entire search tree is denoted as $\boldsymbol{E}$ and grows with each new sample. Initially, the tree contains only the root node $\boldsymbol{b}_0$ which is the state that all components are "as-good-as-new," i.e., $\boldsymbol{b}_0(s') = 1$ with $s_i' = S_i$ for all $i$. This root node represents the belief over component states when the system is brand new. Then, from the root belief $\boldsymbol{b}_0$, the forward-exploration phase first selects the next action using an upper-bound-greedy rule:

$$a^* = \text{argmax}_a \left(\mathcal{R}(\boldsymbol{b}, a) + E_{\boldsymbol{b}'}[V^{up}(\boldsymbol{b}'|\boldsymbol{b}, a)]\right). \quad (32)$$

Once $a^*$ has been determined, SARSOP chooses an observation $z^*$ that maximizes the product of its occurrence probability and the value gap of the successor belief:

$$z^* = \text{argmax}_z O(z|\boldsymbol{b}, a)\left(V^{up}(\boldsymbol{b}'|\boldsymbol{b}, a, z) - V^{lb}(\boldsymbol{b}'|\boldsymbol{b}, a, z)\right), \quad (33)$$

where all quantities appearing in Eqs. (32) and (33) can be obtained from the model in Eqs. (11)–(19). With the selected action $a^*$ and observation $z^*$, the next belief node is computed by Eq. (16) as $\boldsymbol{b}'(s'|\boldsymbol{b}_0, a, z)$. Then, we apply the Bellman backup (Eq. (17)) to update its bounds, $V^{up}(\boldsymbol{b}')$ and $V^{lb}(\boldsymbol{b}')$. These updated bounds are used to update and prune the $\boldsymbol{\alpha}$-vector sets $\boldsymbol{\alpha}^{lb}$. Then, this belief exploration process and bellman backup process continues for $\boldsymbol{b}'$ and so forth. When the difference between $V^{up}(\boldsymbol{b})$ and $V^{lb}(\boldsymbol{b})$ converges below a predefined threshold $\varepsilon$ for all $\boldsymbol{b} \in \boldsymbol{E}$, the algorithm terminates.

Note that SARSOP only samples those belief points that are reachable from the initial belief, thereby eliminating redundant backups. Furthermore, by incrementally expanding nodes in the belief-tree at each step, SARSOP achieves a more focused and computationally efficient update mechanism than the monolithic batch backups of other POMDP solvers such as point-based value iteration (PBVI). A comprehensive algorithmic analysis is presented in Section 6.

### 5.3.2 Macro-actions and reward shaping based on system performance

In this section, we introduce the customized macro-actions and reward shaping to improve the SARSOP and facilitate the solution of the proposed POSMDP.

**Macro-actions.**

Intuitively, component-level inspections are not recommended when system performance is high. Therefore, we may reasonably approximate that, when the system's performance falls below a certain threshold, the operator should conduct component-level inspections and maintain the deteriorated components as needed. However, in the standard SARSOP procedure, if the belief vector $\boldsymbol{b}$ is optimistic about component states, the sampling point is chosen among the next-beliefs reachable from $\boldsymbol{b}$ with the highest value. This action may be repeated many times along the DN path, until the system performance encoded in the belief falls below a low threshold. As a result, the search tree becomes exceedingly deep, and the algorithm's convergence slows down as the exploration near the root is highly inefficient. Motivated by the fact that operators may avoid unnecessary inspections when the system's performance remains high, we introduce a family of macro-actions $\text{Mac}_d$ for $d = 0,1,\ldots$, where macro-action $\text{Mac}_d$ directs the policy to DN action repeatedly until the performance level falls to $d$. Integrating these macro-actions into SARSOP effectively collapses sequences of primitive actions into single decision nodes, which reduces the depth of the policy tree and accelerates the convergence.

Each macro-action $\text{Mac}_d$ is defined by the tuple $< \boldsymbol{I}_d, \pi_d, \boldsymbol{\beta}_d >$, where $\boldsymbol{I}_d$ is the set of beliefs for which $\text{Mac}_d$ can be performed, $\pi_d$ is the local policy applied during its execution, and $\boldsymbol{\beta}_d$ is the set of beliefs that terminate $\text{Mac}_d$. According to Eqs. (28) and (29), we have:

$$\boldsymbol{I}_d = \boldsymbol{B}_d,\ \pi_d(\text{DN}|\mathbf{b}) = 1 \forall \boldsymbol{b} \in \boldsymbol{B}_d,\ \boldsymbol{\beta}_d = \boldsymbol{B}_d^c. \tag{34}$$

A schematic illustrating how the macro-action accelerates exploration of the search-tree root node and reduces the overall search-tree depth is shown in Fig. 3.

Note that in the standard forward-exploration phase of the sampling procedure in SARSOP, once the starting belief point $\boldsymbol{b}$ is determined, the next belief $\boldsymbol{b}'$ is determined in a two-step process. Specifically, an action $a$ that leads to the reachable belief with the highest upper bound of the value function is chosen. Then, the observation $z$ with the largest weighted gap between the upper bound and the lower bound of the value function is identified. That is,

$$a = \text{argmax}_a \left( \mathcal{R}(\boldsymbol{b},a) + \textstyle\sum_{z\in\mathcal{Z}} O(\boldsymbol{b},a,z)\gamma^{\mathcal{U}(a)} V^{up}(\boldsymbol{b}'|\boldsymbol{b},a,z) \right), \tag{35}$$

and

$$z = \text{argmax}_z \left( O(\boldsymbol{b},a,z) \left( V^{up}(\boldsymbol{b}'|\boldsymbol{b},a,z) - V^{lb}(\boldsymbol{b}'|\boldsymbol{b},a,z) \right) \right). \tag{36}$$

Based on these, the belief $\boldsymbol{b}'$ is determined by Eq. (16).

**Remark 3.** When the proposed method is extended to any coherent system according to Remark 2, an improvement in a component state cannot deteriorate the overall system performance. Therefore, when the system performance is above a specified threshold, the macro-actions that repeatedly choose DN until the system performance falls below a threshold remain reasonable.

**Reward shaping.**

Clearly, when the belief $\boldsymbol{b}$ is optimistic about each component state, choosing the default action DN yields a higher upper bound than choosing a macro-action $\text{Mac}_d$. Therefore, to encourage SARSOP to select macro-actions more frequently during the exploration, we introduce a potential-based reward-shaping mechanism (Hossain et al., 2024) that accelerates convergence without altering the overall optimal policy or the true value function. Specifically, denote the shaped reward by $\tilde{\mathcal{R}}(s,a)$:

$$\tilde{\mathcal{R}}(s,a) = \textstyle\sum_{s'} \mathcal{T}(s,a,s') \left( \mathcal{R}(s,a,s') + \gamma^{\mathcal{U}(a)} \varphi(s') - \varphi(s) \right). \tag{37}$$

Here, $\varphi(s)$ is defined as a monotonically decreasing function of system performance under state $s$, e.g.,

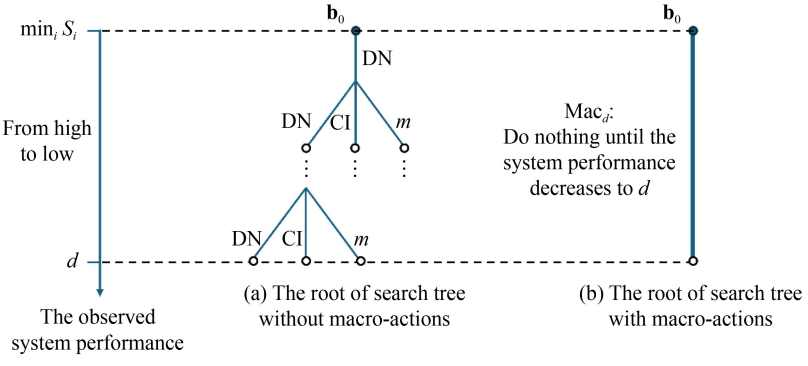


**Fig. 3** Examples of search trees with and without macro-actions.

$\varphi(s) = -\kappa \min_i s_i, \kappa > 0$. Then, substituting it into Eq. (37) shows that when the system state decreases after an action, an additional reward proportional to the magnitude of the decrease is obtained.

**Remark 4.** Moreover, the use of macro-actions makes the computational burden of the proposed two-level IM policy more robust to the choice of the decision interval $\tau$. Specifically, when $\tau$ decreases, the macro-action $\text{Mac}_d$ will be executed once the system performance drops below $d$, so the previously consecutive DN actions can be aggregated into this macro-action, thereby reducing the computational burden. Further, under this reward-shaping scheme, performing macro-action when system performance is high yields a larger reward than DN, because the macro-action incurs a greater performance loss. Referring to Eq. (32), the such reward shaping mechanism encourages the belief nodes search by selecting macro-actions during the algorithm's searching process. This encourages the adoption of macro actions when the system performance is high and is expected to accelerate the solution process. A detailed analysis of how these improvements contribute to SARSOP's performance is presented in Subsection 6.3.

## 6 Case study

In this section, we illustrate the proposed enhanced SARSOP algorithm for solving the optimal bi-level IM policy. We consider the case of three PV panels connected in series investigated in (Huang et al., 2025). During operation, PV panels are subject to dirt (including dust and debris) accumulation that degrades their power-generating capabilities. We categorize their states into five levels: perfectly functioning (state 4), slightly deteriorated (state 3), moderately deteriorated (state 2), heavily deteriorated (state 1), and failure (state 0). Following (Huang et al., 2025), we employ the models in (Wang et al., 2022b, Wang et al., 2020b) to characterize the dirt-accumulation process. By simulating the accumulation process and computing the resulting power-generating capability over time, we approximate the state transition rate matrix $Q$ of each PV panel. The fitted $Q$ and the relationship between the system's revenue $v_z$ and performance $z$ (i.e., the minimum of the component states) are summarized in Tables 1 and 2. The time unit is a day.

Because all components in a series of PV panels carry identical current, the string's maximum-power-point current is limited by the smallest short-circuit current among the PV panels (i.e., the power that the panel would deliver when operated individually). A system-level inspection corresponds to remotely reading the string power output, whereas a component-level inspection corresponds to dispatching a crew to perform on-site I–V curve testing on individual panels and, if necessary, cleaning or replacing them. The interval of system-level inspection $\tau$, the durations and costs associated with inspections $\tau_o, c_o$ and maintenance actions $\tau_i^m, c_i, i = 1,2,3$ are given in Table 2. The discount factor is set as $\gamma = 0.99$.

**Table 1** The state transition rate matrix of each PV panel

| State | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| 4 | −0.0258 | 0.0258 | 0 | 0 | 0 |
| 3 | 0 | −0.0314 | 0.0301 | 0.0013 | 0 |
| 2 | 0 | 0 | −0.0325 | 0.0305 | 0.0020 |
| 1 | 0 | 0 | 0 | −0.0347 | 0.0347 |
| 0 | 0 | 0 | 0 | 0 | 0 |

**Table 2** Other related parameters

| $v_z$ | $\tau_o$ | $\tau_i^m$ | $\tau$ | $c_o$ | $c_i, i = 1,2,3$ | $\gamma$ |
|---|---|---|---|---|---|---|
| $160z$ | 0.5 | 0.5 | 7 | 2000 | 2000 | 0.99 |

In what follows, we introduce the implementation setting of the enhanced SARSOP algorithm, present the optimal bi-level IM policy, and then perform comparative analyses both from the algorithmic perspective and at the maintenance policy level against the existing benchmarks.

### 6.1 Implementation setting

We set the precision threshold for the algorithm's convergence to $\varepsilon = 0.01$. For the macro-actions, we add three macro-actions $\text{Mac}_d$ that chooses "Do Nothing" until the inspected system output falls below state $d$, where $d$ ranges from 3 to 0. The state-transition probabilities after taking $\text{Mac}_d$ from an arbitrary system state are obtained by simulations. For reward shaping, we set $\kappa = 5000$ in $\varphi(s) = -\kappa \min_i s_i, \kappa > 0$. We exemplify the macro-actions $\text{Mac}_d$ and reward shaping by comparing the POSMDP quantities when starting from state (4,4,4) under actions DN and $\text{Mac}_d$, such as the reward incurred until the next decision epoch and the time elapsed until the next decision epoch, as presented in Table 3.

It can be observed that these macro-actions can be regarded as a sequence of consecutive primitive action DN instances, and they are expected to replace several successive DN actions during execution when system

**Table 3** POSMDP quantities when different actions are chosen

| Action | Expected time until the next decision epoch | Expected reward until the next decision epoch | |
|---|---|---|---|
| | | with reshaped reward | without reshaped reward |
| DN | 6.9928 | 6641.1 | 4089.5 |
| $\text{Mac}_3$ | 16.3851 | 14740.7 | 8820.3 |
| $\text{Mac}_2$ | 37.8374 | 29122.6 | 17931.5 |
| $\text{Mac}_1$ | 58.9001 | 40005.1 | 23678.0 |
| $\text{Mac}_0$ | 80.1482 | 47582.8 | 26640.0 |

performance is high, thereby reducing the depth of the belief tree in SARSOP. In Table 3, the expected reward increases when the index $d$ in $\text{Mac}_d$ increases, indicating that reward shaping incentivizes the agent under high-level system performance to choose actions that produce a larger performance decrease in the subsequent decision epoch, which is expected to promote the selection of $\text{Mac}_d$. All codes were run on a machine equipped with an Intel Core i7-13700K processor and 32 GB of RAM.

## 6.2 The optimal bi-level IM policy

The system output must be 0 when the belief that any panel is in state 0 is nonzero, upon which the optimal bi-level IM policy indicates that component-level inspection followed by maintenance must be carried out. Therefore, for illustrative purposes, we present the optimal bi-level IM policy under the following cases in Table 4 as the belief of the state of PV panel 1 varies:

We adopt the ternary plot from (Guo and Liang, 2022) to depict the state distribution of PV panel 1. In this diagram, the three vertices correspond to the three possible states; the probability of the panel being in each state is given by the perpendicular distance from the plotted point to the side opposite that vertex.

**Table 4** Different cases of beliefs about the states of PV panels 2 and 3

| Case | Belief about state of PV Panel 2 | Belief about state of PV Panel 3 |
|---|---|---|
| 1 | (0, 0.5, 0.5, 0, 0) | (0, 0.5, 0.5, 0, 0) |
| 2 | (1/3, 1/3, 1/3, 0, 0) | (1/3, 1/3, 1/3, 0, 0) |
| 3 | (0, 1/3, 1/3, 1/3, 0) | (0, 1/3, 1/3, 1/3, 0) |
| 4 | (0, 0, 0.5, 0.5, 0) | (0, 0, 0.5, 0.5, 0) |

Figure 4 shows that increasingly pessimistic beliefs about a PV panel generally raise the need for component-level inspection. Specifically, as the belief states for PV panels 2 and 3 deteriorate from Case 1 to Case 4, the operator, while keeping the belief for PV panel 1 fixed, has an increasingly stronger inclination to perform component-level inspection (i.e., the shaded region in the ternary plot becomes larger). Moreover, in any given case, when the belief regarding the state of PV panel 1 becomes pessimistic (that is, the case that probability of PV panel 1 is in state 4 is 0 in contrast to the scenario in which the probability that panel 1 is in state 1 is 0), the operator should perform component-level inspections more frequently. With the beliefs about the states of PV panels 2 and 3 held constant, the decision to conduct a component-level inspection, rather than a system-level inspection, is dominantly determined by the probability that PV panel 1 is in its worst state; this is manifested by the nearly horizontal demarcation line between the two inspection regimes in all cases. This phenomenon is particularly obvious when the beliefs regarding the states of PV panels deteriorate, which is evidenced by the increasingly horizontal demarcation line from Case 1 to Case 4 and from Cases (a) in which the probability of PV panel 1 being in state 1 equals zero to Cases (b) that the probability of it being in state 4 equals zero. The underlying rationale is that, under these circumstances, overall system performance has already reached a low level, and therefore, the operator should inspect the system at this point to restore system performance. Additionally, we

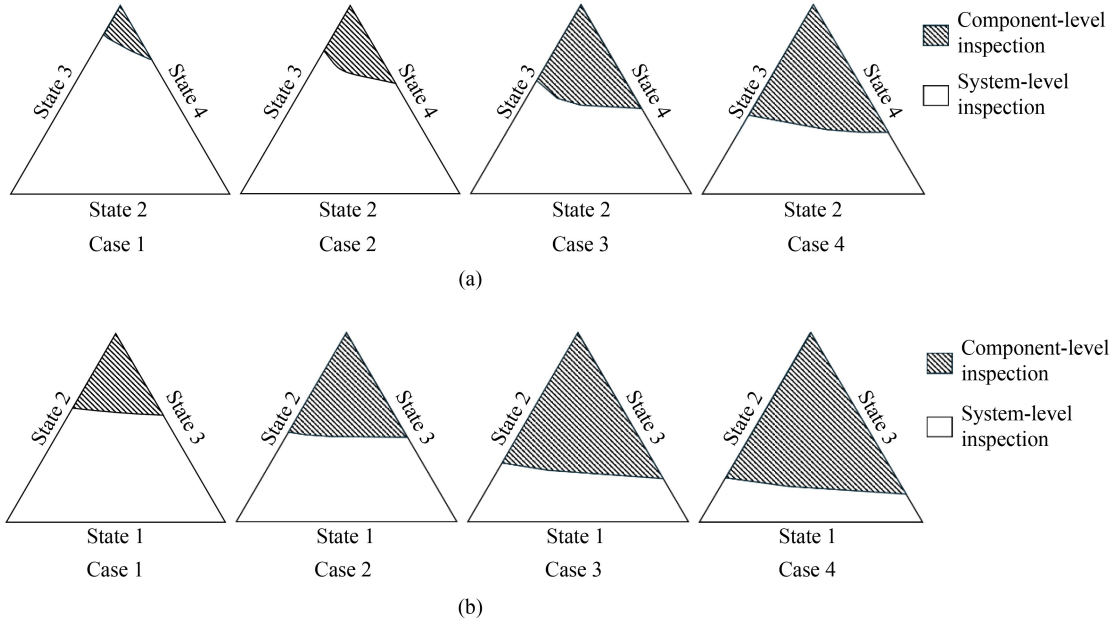


**Fig. 4** Optimal bi-level inspection policies under different cases for (a) when the belief about state of PV panel 1 being 1 is 0, and (b) when the belief about state of PV panel 1 being 4 is 0.

notice that no obvious pattern exists for the system performance threshold that triggers component-level inspection, which distinguishes our proposed bi-level IM policy from the existing threshold-based inspection policy. More details are elaborated in Subsection 6.4.

Next, we give the optimal maintenance policies under all possible component-level inspection outcomes in Fig. 5, where the number following "M" specifies the components on which maintenance is carried out; for example, M12 represents maintenance for PV Panels 1 and 2.

Generally, the optimal maintenance policy for each PV panel is a control-limit policy, where the control limit increases as the states of other panels deteriorate. The interpretation is that other panels are more likely to be maintained when showing worse states during inspection, it is economical to perform opportunistic maintenance on the investigated panel. Moreover, in the optimal bi-level IM policy, the maintenance-related decision-making process is more specific and complex than the inspection-related part. This is because the action set for maintenance is larger than that for inspection-level actions (i.e., DN and CI), which results in more $\boldsymbol{\alpha}$-vectors being associated with their corresponding Bellman backups.

## 6.3 Comparison with other solution algorithms

We compare the proposed enhanced SARSOP algorithm with (i) the standard SARSOP algorithm (i.e., without macro-actions and reward shaping) and (ii) the standard PBVI algorithm (Pineau et al., 2006), which updates the value function at every belief point in the sampled belief pool during each value-iteration step. The 95% confidence intervals of the optimal discounted profit, average time until convergence, and number of $\boldsymbol{\alpha}$ -vectors $|\Gamma|$ at convergence are recorded across 50 independent runs. In the PBVI algorithm, the convergence threshold of the maximum Bellman residual of the value function is set to $\varepsilon = 0.01$.

Table 5 shows that incorporating macro-actions and reward shaping into SARSOP maintains the level of discounted profit obtained by the standard solver while substantially lowering both runtime and policy size. The enhanced SARSOP converges more rapidly than the unmodified SARSOP implementation and much faster than the PBVI, and requires markedly fewer $\boldsymbol{\alpha}$ -vectors, amounting to a reduction of roughly one order of magnitude relative to PBVI. These outcomes indicate that macro-actions compress sequences of primitive decisions and that reward shaping directs the search toward high-value regions of the belief space, enabling SARSOP to converge with fewer backups and more aggressive pruning, thereby producing a more compact policy without sacrificing the solution quality. Additionally, compared with the standard SARSOP, the enhanced SARSOP exhibits a lower variance in discounted profit, a reduction attributable to macro-actions that regularize decision sequences and decreases stochastic fluctuations at the primitive action level.

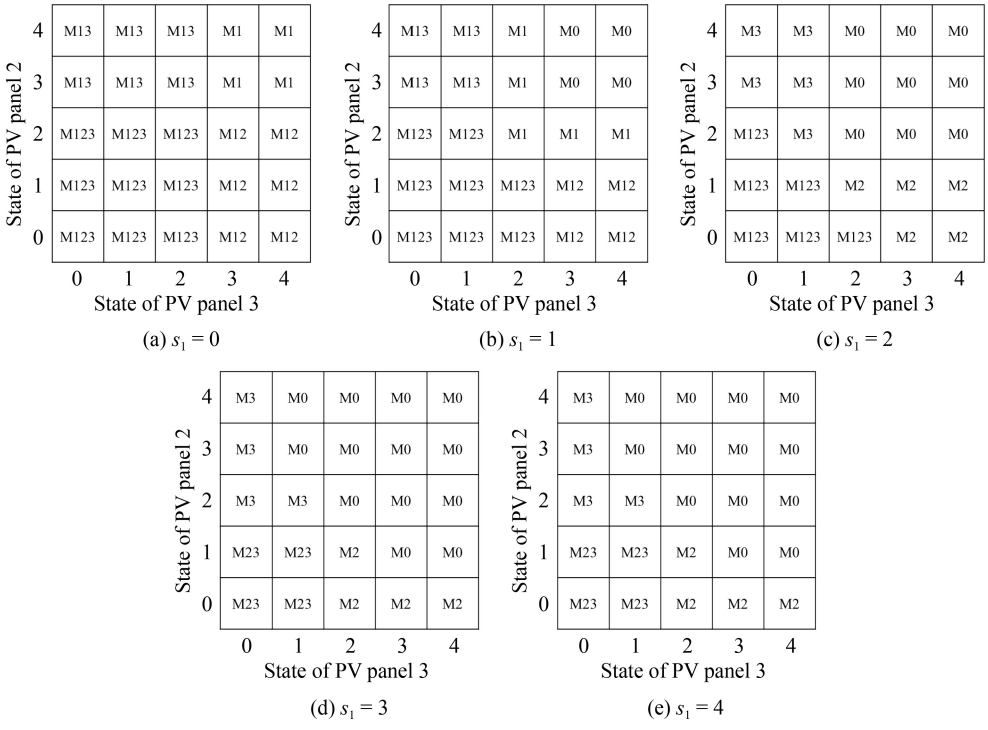


**Fig. 5** Optimal maintenance policy under different component-level inspection outcomes.

**Table 5** Algorithmic performance comparison ($|S| = 125$, $|A| = 9$, $|O| = 5$)

| | Optimal discounted profit | Running time (s) | Number of $\alpha$ - vectors $\|\Gamma\|$ |
|---|---|---|---|
| The enhanced SARSOP | 35225.0 ± 36.5 | 19 | 108 |
| The standard SARSOP | 35220.9 ± 52.9 | 28 | 187 |
| The standard PBVI | 34953.1 ± 25.8 | 101 | 1132 |

## 6.4 Comparison with other IM policies

We compare the proposed bi-level IM policy (abbreviated as **BLIM** in this section) with several benchmarks:

- **Threshold-based inspection and condition-based maintenancepPolicy (TICBM):** When the system performance decreases to a predefined threshold, a component-level inspection is undertaken and maintenance decisions are made based on the inspection results.
- **Periodic inspection and condition-based maintenance policy (PICBM):** A component-level inspection is conducted at every decision epoch; based on the inspection results, the operator makes condition-based maintenance decisions.
- **Periodic inspection and threshold-based maintenance policy (PITBM):** A component-level inspection is performed at every decision epoch; when a component's state reaches a predefined threshold, a maintenance action is performed for the component.
- **Threshold-based maintenance-all (TBMA) policy:** Maintenance actions are performed for all components when the system performance decreases to a predefined threshold.

Table 6 summarizes the optimal discounted profits yielded by these policies, starting from a brand-new system state. For a more in-depth comparison, Fig. 6 presents the discounted profit and its components obtained under different IM policies, whereas Figs. 7–9 illustrates the distribution of an arbitrary component's state (say, PV panel 1) upon system-level inspections, component-level inspections, and maintenance actions for that component across different IM policies.

Table 6 and Fig. 6 show that the proposed BLIM achieves the optimal profit. Furthermore, compared with other policies that conduct component-level inspection, BLIM yields the lowest inspection cost, which reflects both the necessity of incorporating a bi-level inspection scheme and the effectiveness of the proposed solution algorithm. In addition, compared with TICBM, the proposed BLIM generates lower revenue yet attains a higher overall profit by effectively containing IM-related costs. Compared with the remaining alternatives, the proposed BLIM achieves higher revenue and lower IM cost, indicating that it offers superior timing for component-level inspection and a more effective condition-based maintenance policy.

**Table 6** A Comparison of IM Policies' Performance

| | BLIM | TICBM | PICBM | PITBM | TBMA |
|---|---|---|---|---|---|
| The optimal discounted profit | 30225.0 | 29189.9 | 27422.6 | 26952.0 | 25534.0 |

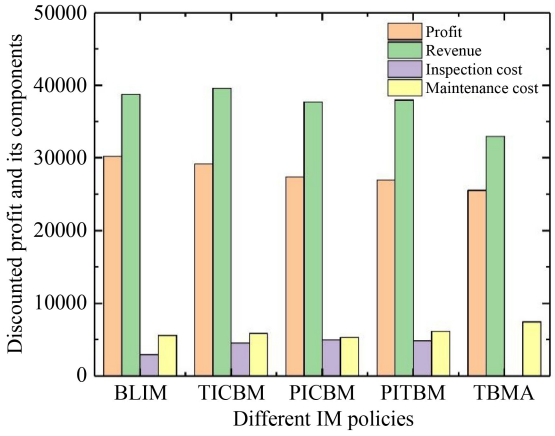


**Fig. 6** The discounted profit and its components under different IM policies.

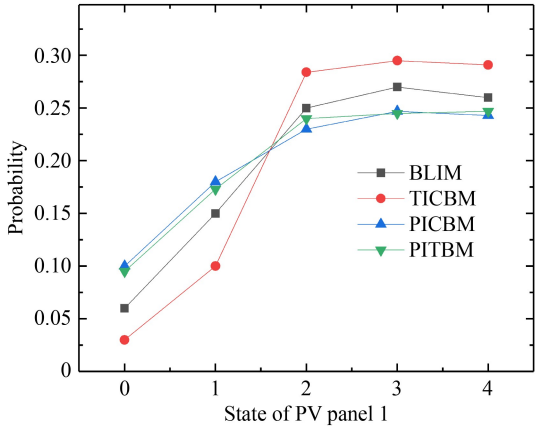


**Fig. 7** Distribution of states of PV Panel 1 when a system-level inspection is performed.

Note that TBMA is not included in Figs. 7 and 8 because it does not involve component-level inspection. Figure 7 shows that, when system-level inspection is performed, the component's state is mostly concentrated in states 2–4. This is because, when the component is in a deteriorated state (i.e., 0 or 1), a component-level inspection and subsequent maintenance actions are carried out, which is also reflected in Fig. 8. Furthermore, although Fig. 7 shows that TICBM keeps the component state above 1 more efficiently than the proposed BLIM, this advantage comes at the expense of more frequent component-level inspection and maintenance, which is reflected in terms of the higher IM cost of TICBM in Fig. 6. In addition, Fig. 8 demonstrates that, during component-level inspection, the component sojourns in state 1 for most of the time under the proposed BLIM, rather than

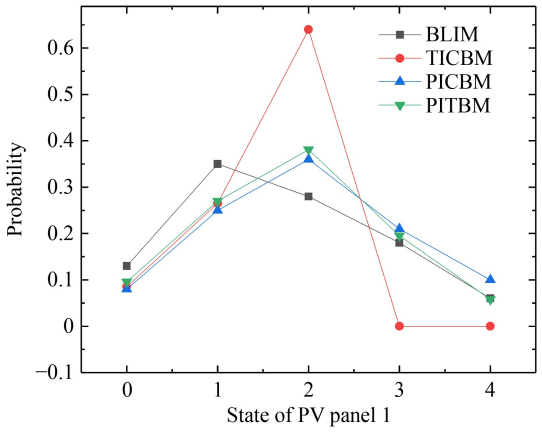


**Fig. 8** The distribution of states of PV Panel 1 when a component-level inspection is performed.

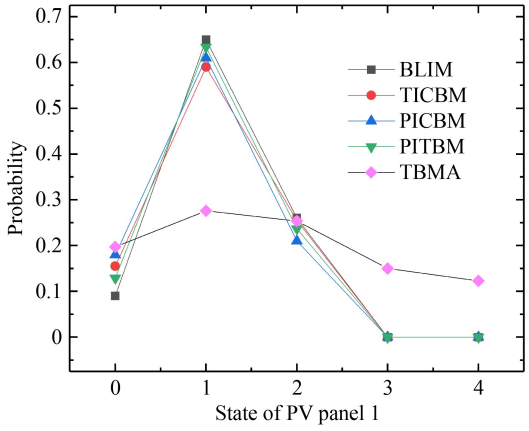


**Fig. 9** The distribution of states of PV Panel 1 when being maintained.

staying in state 2 under other policies. This implies that, for these alternatives, system performance may have to deteriorate to state 2, followed by multiple consecutive component-level inspections before a maintenance decision is made, leading to higher inspection costs. Overall, these comparisons collectively explain the superiority of the proposed BLIM in trading off the frequency of IM actions (i.e., IM cost) against the component state (i.e., revenue), thereby achieving higher profit compared with the alternatives.

## 7 Conclusions and future research

This study proposes and solves an optimal bi-level inspection and maintenance (IM) policy for a coherent multi-component system operating under partial observability and non-negligible action durations. By embedding the sequential IM decisions into a partially observable semi-Markov decision process (POSMDP) and devising an enhanced SARSOP algorithm with macro-actions and potential-based reward shaping, we provide a tractable way to optimize inspection and maintenance decisions when system performance can be monitored cheaply at the system level but component-level inspections and maintenance are costly and time-consuming.

From a managerial perspective, the main insight is that operators do not need to adhere to rigid "inspect-all-components-at-every-epoch" routines. Instead, the bi-level IM policy offers an economically grounded rule for when to use only low-cost system-level monitoring and when to escalate to expensive component-level inspections and targeted maintenance. In particular, the optimal policy recommends (i) skipping high-cost component-level inspections when the belief about component health (updated from system-level observations and history) remains sufficiently favorable, and (ii) triggering component-level inspections and subsequent maintenance actions only when this belief has deteriorated to a level where the expected long-run profit justifies the additional inspection and maintenance effort. The photovoltaic-panel case study shows that following such a policy can markedly improve long-run system profitability and reduce unnecessary inspection and maintenance activities, while remaining computationally efficient compared with existing benchmark policies. These findings highlight that effective asset management in practice requires jointly exploiting both system-level and component-level information, rather than relying solely on either coarse system indicators or overly frequent detailed inspections.

Based on these findings, several potential future research directions enriching the proposed POSMDP framework are identified. First, future research could further investigate imperfect inspections and imperfect maintenance actions, building on the model extensions discussed in Section 4, and study their impact on the structure and robustness of the optimal bi-level IM policy. Second, it would be valuable to extend the model to allow component-level inspections to target only a subset of components at each decision epoch, for instance by imposing a capacity constraint on the maximum number of components that can be inspected and by prioritizing components according to their criticality or current deterioration level. Under such an extension, the bi-level IM policy would naturally generalize to a multi-level one, and we could then examine how specific selection rules (e.g., always inspecting the $k$ components with the lowest belief of being in a healthy state) affect computational tractability, the structure of the optimal multi-level IM policy, and the resulting system performance. Third, adaptive threshold-based macro-actions could be learned online, for example, by embedding a Bayesian update on deterioration rates so that skip-or-inspect decisions evolve with accumulating data.

**Competing Interests** The authors declare that they have no competing interests.

## References

Bellman R (1966). Dynamic programming. Science, 153(3731): 34–37

Bouslah B, Gharbi A, Pellerin R (2016). Integrated production, sampling quality control and maintenance of deteriorating production systems with AOQL constraint. Omega, 61: 110–126

Chen H, Wu M, Shi Y, Xiahou T, Chen J, Zhao Z, Liu Y (2025). A deep learning-enhanced active sampling approach to evidential uncertainty propagation. Journal of Mechanical Design, 148(4): 041703

Chen Y M, Liu Y, Xiahou T (2024). Dynamic inspection and maintenance scheduling for multi-state systems under time-varying demand: Proximal policy optimization. IISE Transactions, 56(12): 1245–1262

Chen Y M, Liu Y, Xiahou T F (2022). A deep reinforcement learning approach to dynamic loading strategy of repairable Multistate Systems. IEEE Transactions on Reliability, 71(1): 484–499

Cheng Y, Wei Y, Liao H T (2022). Optimal sampling-based sequential inspection and maintenance plans for a heterogeneous product with competing failure modes. Reliability Engineering & System Safety, 218: 108181

Dinh D H, Do P, Iung B (2022). Multi-level opportunistic predictive maintenance for multi-component systems with economic dependence and assembly/disassembly impacts. Reliability Engineering & System Safety, 217: 108055

Dui H Y, Wu X M, Wu S M, Xie M (2024). Importance measure-based maintenance strategy optimization: Fundamentals, applications and future directions in AI and IoT. Frontiers of Engineering Management, 11(3): 542–567

Fang H J, Zheng R, Xia X D, Hu C W (2025). Condition-based maintenance policy for a two-component balanced system with a multimode protective device. Reliability Engineering & System Safety, 262: 111195

Guo C H, Liang Z L (2022). A predictive Markov decision process for optimizing inspection and maintenance strategies of partially observable multi-state systems. Reliability Engineering & System Safety, 226: 108683

Hao S H, Yang J, Berenguer C (2020). Condition-based maintenance with imperfect inspections for continuous degradation processes. Applied Mathematical Modelling, 86: 311–334

Hashemi M, Asadi M (2021). Optimal preventive maintenance of coherent systems: A generalized Polya process approach. IISE Transactions, 53: 1266–1280

Hossain R R, Yin T Z, Du Y, Huang R K, Tan J, Yu W H, Liu Y, Huang Q H (2024). Efficient learning of power grid voltage control strategies via model-based deep reinforcement learning. Machine Learning, 113(5): 2675–2700

Hu J W, Sun Q Z (2026). A dynamic inspection and replacement policy for systems subject to degradation and periodic shocks. Reliability Engineering & System Safety, 266: 111765

Hu J W, Sun Q Z, Ye Z S (2021). Condition-based maintenance planning for systems subject to dependent soft and hard failures. IEEE Transactions on Reliability, 70(4): 1468–1480

Huang Z Y, Wei Y, Cheng Y (2025). A quantitative framework for performance-based reliability prediction for a multi-component system subject to dynamic self-reconfiguration. Reliability Engineering & System Safety, 262: 111188

Khaleghei A, Kim M J (2021). Optimal control of partially observable semi-Markovian failing systems: A *n* analysis using a phase methodology. Operations Research, 69(4): 1282–1304

Kou G, Liu Y Y, Xiao H, Peng R (2023). Optimal Inspection Policy for a Three-Stage System Considering the Production Wait Time. IEEE Transactions on Reliability, 72(3): 934–949

Kurniawati H, Hsu D, Lee W S (2008). SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. . In: Proceedings of Robotics: Science and Systems (RSS 2008), Zurich, Switzerland

Liu X C, Sun Q Z, Ye Z S, Yildirim M (2021). Optimal multi-type inspection policy for systems with imperfect online monitoring. Reliability Engineering & System Safety, 207: 107335

Liu Y, Zhang B Y, Jiang T, Xiahou T F (2020). Optimization of multi-level inspection strategy for nonrepairable multistate systems. IEEE Transactions on Reliability, 69(3): 968–985

Lv X L, Shi L X, He Y D, He Z, Lin D K J (2024). Joint optimization of production, maintenance, and quality control considering the product quality variance of a degraded system. Frontiers of Engineering Management, 11(3): 413–429

Mahmoudi M, Elwany A, Shahanaghi K, Gholamian M R (2017). A delay time model with multiple defect types and multiple inspection methods. IEEE Transactions on Reliability, 66(4): 1073–1084

Naderkhani ZG F, Makis V (2015). Optimal condition-based maintenance policy for a partially observable system with two sampling intervals. International Journal of Advanced Manufacturing Technology, 78(5-8): 795–805

Nguyen K T P, Do P, Huynh K T, Bérenguer C, Grall A (2019). Joint optimization of monitoring quality and replacement decisions in condition-based maintenance. Reliability Engineering & System Safety, 189: 177–195

Papakonstantinou K G, Andriotis C P, Shinozuka M (2018). POMDP and MOMDP solutions for structural life-cycle cost minimization under partial and mixed observability. Structure and Infrastructure Engineering, 14(7): 869–882

Pineau J, Gordon G, Thrun S (2006). Anytime point-based approximations for large POMDPs. Journal of Artificial Intelligence Research, 27: 335–380

Qiu Q, Cui L, Wu B (2020). Dynamic mission abort policy for systems operating in a controllable environment with self-healing mechanism. Reliability Engineering & System Safety, 203: 107069

Qiu Q, Kou M, Chen K, Deng Q, Kang F, Lin C (2021). Optimal stopping problems for mission oriented systems considering time redundancy. Reliability Engineering & System Safety, 205: 107226

Qiu Q, Li R, Zhao X (2025). Failure risk management: Adaptive performance control and mission abort decisions. Risk Analysis, 45(2): 421–440

Qiu Q, Maillart L M, Prokopyev O A, Cui L (2023). Optimal condition-based mission abort decisions. IEEE Transactions on Reliability, 72(1): 408–425

Song M Y, Liang Z L (2025). A hierarchical state ordering method for heterogeneous *k*-out-of-*n*: F systems with multi-state load-sharing components. Reliability Engineering & System Safety, 262: 111167

Sun Q Z, Hu T W, Ye Z S (2025). Optimal abort policy for mission-critical systems under imperfect condition monitoring. Operations Research, 73(5): 2396

Sun Q Z, Ye Z S, Chen N (2018). Optimal inspection and replacement policies for multi-unit systems subject to degradation. IEEE Transactions on Reliability, 67(1): 401–413

Sun Q Z, Ye Z S, Zhu X Y (2020). Managing component degradation in series systems for balancing degradation through reallocation and maintenance. IISE Transactions, 52(7): 797–810

Tan L, Wei F, Ma X, Peng R, Xiao H, Yang L (2025). Systemic condition-based maintenance optimization under inspection uncertainties: A customized multiagent reinforcement learning approach. IEEE Transactions on Reliability, 74(4): 5848–5862

Tang X, Xiao H, Kou G, Xiang Y (2024). Joint optimization of condition-based maintenance and spare parts ordering for a hidden multi-state deteriorating system. IEEE Transactions on Reliability, 74(2): 2503–2514

Vu H C, Do P, Barros A (2018). A study on the impacts of maintenance duration on dynamic grouping modeling and optimization of multi-component systems. IEEE Transactions on Reliability, 67(3): 1377–1392

Wang J J, Yang L, Ma X B, Peng R (2021). Joint optimization of multi-window maintenance and spare part provisioning policies for production systems. Reliability Engineering & System Safety, 216: 108006

Wang J T, Zhou S H, Peng R, Qiu Q A, Yang L (2023). An inspection-based replacement planning in consideration of state-driven imperfect inspections. Reliability Engineering & System Safety, 232: 109064

Wang W (2000). A model of multiple nested inspections at different intervals. Computers & Operations Research, 27(6): 539–558

Wang X Y, Ning R, Zhao X, Zhou J (2022a). Reliability analyses of *k*-out-of-*n*: F capability-balanced systems in a multi-source shock environment. Reliability Engineering & System Safety, 227: 108733

Wang X Y, Zhao X, Wang S Q, Sun L P (2020a). Reliability and maintenance for performance-balanced systems operating in a shock environment. Reliability Engineering & System Safety, 195: 106705

Wang X Y, Zhao X Y, Zhao X, Chen X, Ning R (2024). Reliability assessment for a generalized *k*-out-of-*n*: F system under a mixed shock model with multiple sources. Computers & Industrial Engineering, 196: 110459

Wang Z H, Xu Z G, Wang X L, Xie M (2022b). A temporal-spatial cleaning optimization method for photovoltaic power plants. Sustainable Energy Technologies and Assessments, 49: 101691

Wang Z H, Xu Z G, Zhang Y, Xie M (2020b). Optimal cleaning scheduling for photovoltaic systems in the field based on electricity generation and dust deposition forecasting. IEEE Journal of Photovoltaics, 10(4): 1126–1132

Wei Y, Cheng Y (2025). An optimal two-dimensional maintenance policy for self-service systems with multi-task demands and subject to competing sudden and deterioration-induced failures. Reliability Engineering & System Safety, 255: 110628

Wei Y, Cheng Y, Liao H T (2025a). Fleet service reliability analysis of self-service systems subject to failure-induced demand switching and a two-dmensional inspection and maintenance policy. IEEE Transactions on Automation Science and Engineering, 22: 10029–10044

Wei Y, Cheng Y, Liao H T (2025b). A quantitative maintenance policy development framework for a fleet of self-service systems. Naval Research Logistics, 72(5): 750–767

Wei Y, Li A C, Cheng Y, Li Y (2025c). An optimal multi-level inspection and maintenance policy for a multi-component system with a protection component. Computers & Industrial Engineering, 201: 110898

Wei Y, Liao H T, Cheng Y (2025d). Making an optimal inspection-free maintenance decision for a coherent system subject to hidden malfunctions. IISE Transactions

Xiahou T F, Zheng Y X, Liu Y, Chen H (2023). Reliability modeling of modular *k*-out-of-*n* systems with functional dependency: A case study of radar transmitter systems. Reliability Engineering & System Safety, 233: 109120

Xiao H, Yan Y M, Kou G, Wu S M (2023). Optimal inspection policy for a single-unit system considering two failure modes and production wait time. IEEE Transactions on Reliability, 72(1): 395–407

Xiao H, Yi K, Kou G, Xing L (2020). Reliability of a two-dimensional demand-based networked system with multistate components. Naval Research Logistics, 67: (6)453–468

Yang L, Chen Y, Ma X B (2023). A state-age-dependent opportunistic intelligent maintenance framework for wind turbines under dynamic wind conditions. IEEE Transactions on Industrial Informatics,19(10): 10434–10443

Yang L, Chen Y, Ma X B, Qiu Q G, Peng R (2024a). A prognosis-centered intelligent maintenance optimization framework under uncertain failure threshold. IEEE Transactions on Reliability, 73(1): 115–130

Yang L, Wei F P, Qiu Q G (2024b). Mission risk control via joint optimization of sampling and abort decisions. Risk Analysis, 44(3): 666–685

Zhang B Y, Liu Y, Xiahou T (2024a). Importance measure for multilevel inspections of multistate systems: A value of information perspective. IEEE Transactions on Reliability, 73(2): 885–901

Zhang P, Zhu X Y, Xie M (2021). A model-based reinforcement learning approach for maintenance optimization of degrading systems in a large state space. Computers & Industrial Engineering, 161: 107622

Zhang Q, Liu Y, Xiang Y S, Xiahou T (2024b). Reinforcement learning in reliability and maintenance optimization: A tutorial. Reliability Engineering & System Safety, 251: 110401

Zheng R, Li M M, Fang C, Wu K (2026). Optimization of an adaptive mission abort policy for a partially observable system. Reliability Engineering & System Safety, 267: 111945

Zheng R, Qian X F, Gu L D (2023). Group maintenance for numerical control machine tools: A case study. IEEE Transactions on Reliability, 72(4): 1407–1419

Zheng R, Zhou Y F (2022). A dynamic inspection and replacement policy for a two-unit production system subject to interdependence. Applied Mathematical Modelling, 103: 221–237

Zhao S Q, Wei Y, Cheng Y, Li Y (2025a). A state-specific joint size, maintenance, and inventory policy for a *k*-out-of-*n* load-sharing system subject to self-announcing failures. Reliability Engineering & System Safety, 257: 110855

Zhao S Q, Wei Y, Li Y, Cheng Y (2026). A multi-agent reinforcement learning (MARL) framework for designing an optimal state-specific hybrid maintenance policy for a series *k*-out-of-*n* load-sharing system. Reliability Engineering & System Safety, 265: 111587

Zhao X J, Chen P, Tang L C (2025b). Condition-based maintenance via Markov decision processes: A review. Frontiers of Engineering Management, 12(2): 330–342

Zhao X J, Wang Z Y (2022). Maintenance policies for two-unit balanced systems subject to degradation. IEEE Transactions on Reliability, 71(2): 1116–1126

Zhou X J, Huang K M, Xi L F, Lee J (2015). Preventive maintenance modeling for multi-component systems with considering stochastic failures and disassembly sequence. Reliability Engineering & System Safety, 142: 231–237