European Journal of Operational Research 315 (2024) 176–190

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Production, Manufacturing, Transportation and Logistics

Online reinforcement learning for condition-based group maintenance using factored Markov decision processes

Jianyu Xu [a], Bin Liu [b,*], Xiujie Zhao [c], Xiao-Lin Wang [d]

[a] International Business School Suzhou, Xi'an Jiaotong-Liverpool University, Suzhou, China
[b] Department of Management Science, University of Strathclyde, Glasgow, G1 1XQ, UK
[c] College of Management and Economics, Tianjin University, Tianjin, China
[d] Business School, Sichuan University, Chengdu 610065, China

ARTICLE INFO

Keywords:
Maintenance
Condition-based group maintenance
Factored Markov decision process
Factored value iteration
Online reinforcement learning

ABSTRACT

We investigate a condition-based group maintenance problem for multi-component systems, where the degradation process of a specific component is affected only by its neighbouring ones, leading to a special type of stochastic dependence among components. We formulate the maintenance problem into a factored Markov decision process taking advantage of this dependence property, and develop a factored value iteration algorithm to efficiently approximate the optimal policy. Through both theoretical analyses and numerical experiments, we show that the algorithm can significantly reduce computational burden and improve efficiency in solving the optimization problem. Moreover, since model parameters are unknown *a priori* in most practical scenarios, we further develop an online reinforcement learning algorithm to simultaneously learn the model parameters and determine an optimal maintenance action upon each inspection. A novel feature of this online learning algorithm is that it is capable of learning both transition probabilities and system structure indicating the stochastic dependence among components. We discuss the error bound and sample complexity of the developed learning algorithm theoretically, and test its performance through numerical experiments. The results reveal that our algorithm can effectively learn the model parameters and approximate the optimal maintenance policy.

1. Introduction

Maintenance serves as an essential measure to improve system reliability, sustain system operations, and reduce operating costs in various industries, including energy-generation, manufacturing, and transportation, among others. From the modelling perspective, maintenance models can be classified into two categories: maintenance models for single-unit systems and those for multi-component systems (de Jonge & Scarf, 2020). In the reliability and maintenance field, numerous studies have been devoted to single-unit systems, while maintenance problems for multi-component systems are generally much more complex, due to the presence of multiple components as well as various dependencies among components. In particular, dependencies among components are typically categorized into three types in the literature (Olde Keizer et al., 2017a): *structural dependence* (i.e., maintenance of a certain component requires the dismantling or maintenance of other components because of the system's physical structure), *stochastic dependence* (i.e., degradation or failure process of a certain component affects those of other components), and *economic dependence* (i.e., combining maintenance on multiple components is less expensive than maintaining each component separately). The recent literature has introduced two new types of dependence—*resource dependence* (Olde Keizer et al., 2017a) and *geographical dependence* (Nguyen et al., 2019). The former occurs when multiple components share a limited pool of maintenance resources (e.g., spares, tools, and crews), while the latter—concerning geographically dispersed systems—applies when jointly maintaining several components results in a smaller total travel distance/time than maintaining each component individually.

Group maintenance is widely adopted for multi-component systems; it prescribes the maintenance of a group of components together, thereby attaining economies of scale (Abbou & Makis, 2019; Wildeman et al., 1997). The rationale is that simultaneously maintaining multiple components can reduce the setup cost compared with maintaining them individually. Traditionally, group maintenance is conducted based on ages and failure distributions of the components, which is referred to as time-based maintenance. Recent advances in sensing technology enable monitoring system conditions in a low-cost manner, propelling

* Corresponding author.
E-mail addresses: b.liu@strath.ac.uk (B. Liu), xiaolinwang@scu.edu.cn (X.-L. Wang).

https://doi.org/10.1016/j.ejor.2023.11.039
Received 13 May 2023; Accepted 24 November 2023
Available online 28 November 2023

a shift in maintenance paradigm towards condition-based maintenance (CBM). Within the framework of CBM, optimal maintenance decisions are determined based on observed health conditions—obtained through either discrete inspection or continuous monitoring—of the systems and/or their components (Ahmad & Kamaruddin, 2012). However, developing a condition-based group maintenance policy based on component health conditions is rather complicated given the dependencies among the components.

Markov decision process (MDP), a well known stochastic control process, has been widely used to model CBM problems where a system is represented by a set of states that present random evolvement (Gámiz et al., 2023; Liu et al., 2021). MDP is an effective and flexible modelling tool for single-unit systems in the sense that it is able to evaluate and optimize maintenance policies for either a finite or an infinite horizon. However, for a multi-component system, any system state is a combination of the states of all the components. This leads to the so-called *curse of dimensionality* (i.e., the exponential explosion of the number of system states). Even for systems with moderate number of components (say, 15 to 20 ones), traditional MDPs would suffer from notorious computational complexity and cannot be directly applied. To relieve this issue, the factored MDP (FMDP) model is developed to represent large MDPs with factored structures (Talebi et al., 2021). In particular, FMDP separates the transitions and costs into their counterparts defined on small sets of elements in the state vector, which can reduce computational complexity in determining an optimal policy. FMDP is a promising approach to solving the group maintenance problem for multi-component systems, since degradation of a component might depend only on a small cluster of "neighbouring" components. In this case, maintenance cost can also be decomposed as the sum of the costs related to the individual or small sets of components (Zhou et al., 2018).

In practice, some large-scale, multi-component systems can be decomposed into a number of locally interactive components and thus modelled by an FMDP. A typical example is maintenance of railway tracks. A railway network consists of thousands of tracks, among which the tracks in a specific area are interdependent in the sense that they are operating in a common environment and under similar traffic loads. In this sense, the tracks are subject to location-based stochastic dependence; that is, neighbouring tracks are expected to have a higher level of dependence than those distant ones (Brown et al., 2022). When conducting maintenance activities on railway tracks, decision makers have to consider the effect of such dependencies so as to improve maintenance efficiency. Another example is maintenance of machines in a production line. In modern manufacturing scenarios, multiple machines in a production line work collectively to manufacture a product. Degradation of a certain machine reduces the quality of the parts produced, which might further affect the degradation of downstream machines. For example, the degradation process of cutting tools is affected by defective materials/parts from the previous production cell. In reality, such effect usually diminishes with the "distance" of machines. From the modelling perspective, this dependence relationship among components of a complex system can be modelled by a dynamic Bayesian network or its variants (Guestrin et al., 2003).

A practical issue to be considered when applying FMDP in the group maintenance problem is that the transition probabilities are generally unknown *a priori*. An implicit assumption adopted by most literature is that model parameters are fully known to the decision maker; this assumption is, unfortunately, not realistic in most practical scenarios, as the parameters often need to be estimated from historical data or suggested by domain experts. In this work, focusing on group maintenance for multi-component systems subject to economic and stochastic dependencies, we tackle this issue by developing an online reinforcement learning algorithm to simultaneously learn the transition probabilities and determine an optimal maintenance policy. To the best of our knowledge, this is the first work that presents an online reinforcement learning algorithm for CBM problems. In particular, the algorithm is devised by extending the existing factored Model-Based Interval Estimation (fMBIE) approach to an online learning scenario. The performance of this algorithm is investigated in both theoretical and numerical manners.

We summarize the contributions of this work in the following four aspects:

- We formulate a condition-based group maintenance problem for multi-component systems with stochastic and economic dependencies into an FMDP.
- We develop a modified factored value iteration algorithm to improve computational efficiency in calculating an optimal policy.
- We develop a novel online learning algorithm to learn the parameters and the dependence relationship and optimize the maintenance policy, simultaneously.
- We theoretically prove the bound of errors for the proposed online learning approach against the nominal model.

The remainder of the paper is organized as follows. Section 2 reviews relevant literature on CBM for multi-component systems and reinforcement learning for FMDPs. Section 3 formally describes the condition-based group maintenance problem and formulates the problem into an FMDP. For the nominal model with known parameters, Section 4 presents a modified factored value iteration algorithm to compute an optimal maintenance policy. Further considering the scenario in which model parameters are unknown *a priori*, Section 5 presents an online learning algorithm to learn the model parameters and evaluates the performance of this algorithm. Section 6 conducts numeral experiments to validate the developed approaches. Finally, Section 7 concludes the paper and suggests future research topics. All proofs can be found in Appendix A.

2. Related literature

We discuss two major research streams that are related to our work: (i) CBM for multi-component systems and (ii) reinforcement learning for FMDPs.

2.1. CBM for multi-component systems

Though most CBM-related studies are focused on single-unit systems (see, e.g., Chen et al., 2015; Deep et al., 2023; Drent et al., 2023; Elwany et al., 2011; Khaleghei & Kim, 2021), the CBM problems for multi-component systems have attracted an increasing attention in the literature (see, e.g., Olde Keizer et al., 2016; Tian & Liao, 2011; Zhu & Xiang, 2021). One can refer to Olde Keizer et al. (2017a) for a comprehensive overview on CBM for multi-component systems, in which the CBM policies are classified in terms of dependence type.

We confine our attention to MDP-based CBM studies for multi-component systems. Sun et al. (2017) develop a CBM model for multi-component systems with identical and independent components. Liu et al. (2021) investigate a CBM problem for two-component systems with heterogeneous and dependent components. They formulate the problem into a finite-horizon MDP and characterize the optimal preventive maintenance curve in terms of the components' degradation levels. Barlow et al. (2021) propose a performance-centred approach to optimizing maintenance of complex systems with multiple components and adopt a reinforcement learning algorithm to solve the complex optimization problem. Hoffman et al. (2022) develop an online improvement approach for CBM with Monte Carlo tree search. They develop a two-stage approach that first optimizes the static CBM policy and then uses Monte Carlo tree search to further improve the static policy. Olde Keizer et al. (2017b) develop a joint CBM and inventory model to reduce the maintenance and inventory losses through optimizing maintenance and spare ordering decisions based on the components' conditions. Wang and Zhu (2021) propose a joint CBM and inventory model that determines the optimal decisions based on the number of

components in each degradation state instead of the condition of each individual component. (Zheng et al., 2023) jointly optimize CBM and spare provisioning for a K-out-of-N system, where system degradation states are revealed upon periodic inspections that trigger the opportunities to replace components and order spare parts. When using MDP to model CBM for multi-component systems, a common yet burdensome issue is the computational complexity, since the number of system states increases exponentially with the number of components.

FMDP is an effective modelling approach to compactly representing the stochastic nature of degrading systems, which is, to some extent, helpful for resolving the curse of dimensionality. However, studies on FMDP-based maintenance optimization are rather limited. Zhou et al. (2016) formulate the maintenance problem of a multi-component system into an FMDP model and develop an improved approximate linear programming algorithm to solve the problem. Zhou et al. (2018) further investigate maintenance optimization of a series production system and develop a multi-agent FMDP model to select the maintenance actions in cooperation of different agents. Kıvanç et al. (2022) employ a factored partially observable MDP model to investigate the maintenance problem of a regenerative air heater system subject to stochastic and economic dependencies. Nevertheless, the aforementioned research implicitly assumes that model parameters and dependence structures are known to the decision maker in advance. This assumption is, unfortunately, not realistic as the parameters and structures usually need to be estimated. To address this issue, this work aims to develop an online learning algorithm for multi-component systems modelled by an FMDP to jointly learn the parameters and dependence structure, while determining optimal maintenance actions.

2.2. Reinforcement learning for FMDPs

In the literature, several attempts have been made on reinforcement learning for FMDPs. Kearns and Koller (1999) present an efficient and near-optimal algorithm for reinforcement learning in an FMDP framework, where the structure is modelled by a dynamic Bayesian network. Sallans and Hinton (2004) propose a novel approximation method to approximate the value function and select actions for MDPs with large state and action spaces. The approach enables determining the actions in large factored action spaces via Markov chain Monte Carlo sampling. Degris et al. (2006) develop a general framework that integrates FMDP-based incremental planning algorithms with supervised learning techniques to build structured representations of the reinforcement learning problem. Strehl (2007) extends the work of Kearns and Koller (1999) by employing the Interval Estimation approach for exploration, which outperforms traditional algorithms on most domains. Strehl et al. (2007) propose an efficient reinforcement learning algorithm to learn the unknown dynamic Bayesian network structure for an FMDP. Mahadevan and Maggioni (2007) develop a novel spectral framework to jointly learn the representations and the optimal policy for MDPs and FMDPs. Szita and Lörincz (2009) develop a factored optimistic algorithm to attain polynomial-time reinforcement learning in FMDPs. The work emphasizes the importance of initialization and proves that suitable initialization can lead to convergence and polynomial-time number of steps for near-optimal decisions. Osband and Van Roy (2014) report that it is possible to achieve regret that scales polynomially in the number of parameters encoding an FMDP. In addition, the work presents two algorithms that satisfy near-optimal regret bounds in this setting. Tian et al. (2020) investigate minimax optimal reinforcement learning for episodic FMDPs. By assuming that the factorization is known beforehand, two model-based algorithms that attain minimax optimal regret guarantees are proposed. Xu and Tewari (2020) develop oracle-efficient algorithms that achieve tighter regret bounds for non-episodic FMDPs. Deng et al. (2022) design the first polynomial-time algorithm for reinforcement learning in FMDPs that only requires a linear value function with a suitable local basis with respect to the factorization, permitting efficient variable elimination.

Though significant progresses have been made on reinforcement learning for FMDPs in a general context, up to now no efforts have been devoted to the specific condition-based group maintenance problem where the interactions among components are represented by location-based stochastic dependence. Different from most of the existing studies that focus on offline reinforcement learning algorithms, we develop a novel and more efficient online learning algorithm tailored to the condition-based group maintenance problem that is able to simultaneously learn the transition probabilities and the dependence relationship.

Notations

K, $\mathcal{K}$	number and set of all components
N	number of states for each component
$\mathcal{S}$	space of all state vectors of all components
x^k	state of each single component k
x	state vector of all components
$\mathcal{A}$	action space
a^k	action on each single component k
a	action vector on all components
A	maximum number of components maintained at each inspection
$P(\cdot)$	transition probability of system state between two successive inspections
P_R^k	instant transition of state of each component k upon an inspection (due to the "replacement" or "do nothing" action taken)
d^k	maximum distance a component being from the kth component
$\mathcal{U}_k$	set of neighbouring components whose state transitions depend on component k
P_D^k	transition of state of each component k within the degradation interval
P_D	transition of system state within the degradation interval
c	total maintenance cost for the whole system
c^k	maintenance cost for each component k
φ	cost factor reflecting economical dependence among components
π	a generic policy
π^*	the optimal policy

3. The condition-based group maintenance problem

In this section, we formally describe the condition-based group maintenance problem for a multi-component system in Section 3.1, and then formulate the problem into an FMDP in Section 3.2.

3.1. Problem description

We consider the maintenance problem for a multi-component system consisting of K non-identical components, where the health condition of each component deteriorates during operations. In reality, such health condition varies for different systems, such as crack sizes of railway tracks and charging/discharging rates of batteries. Moreover, the degradation process of a specific component is affected by only its neighbouring ones instead of all components; that is, there is a special type of *stochastic dependence* among the components (Olde Keizer et al., 2017a). The degradation process of the whole system, without interventions, is assumed to follow a Markov chain. To improve system reliability and sustain system operations, periodic inspections are carried out to reveal the system and component states. At each inspection, maintenance action might be implemented, depending on the observed component states. For each component, we only consider two actions: "do nothing" and "replacement", while the case involving imperfect maintenance actions of different depths is left for future research. The "replacement" action restores a component to an as-good-as-new state. The time duration required to complete a replacement

action is assumed to be negligible compared with the inspection interval, which is a common assumption in maintenance studies (see, e.g., Drent et al., 2023; Liu et al., 2021). We consider that maintenance on a group of components can reduce costs compared to separate maintenance on individual components, corresponding to the so-called *economic dependence* (Zhao et al., 2022). However, due to the limited availability of maintenance crews, only a set of components can be maintained at one time, reflecting the *resource dependence* among the components (Olde Keizer et al., 2017a).

In this problem, the transition of a component's state between any two successive inspections can be caused by maintenance at the former inspection and/or natural degradation during this interval. At inspection, the component state is determined immediately by the action (replacement or do nothing) applied. Specifically, if the replacement action is taken, then the component is instantly restored to an as-good-as-new state, given negligible replacement duration; otherwise, the component state remains unchanged. Within the interval between two successive inspections, each component gradually deteriorates to a worse state during operations. Because of the location-based stochastic dependence, the degradation process of each component is affected only by its neighbouring components.

Our objective is to determine an optimal maintenance policy so that the total discounted long-run cost is minimized. In doing so, we first study the optimal maintenance policy in a scenario where the degradation parameters (i.e., transition probabilities of the Markov chain and the parameters associated with the stochastic dependence) are pre-known, and then extend our attention to a more typical online maintenance scenario where the parameters are unknown *a priori*.

3.2. Model formulation

Based on the previous description, we now formulate the condition-based group maintenance problem into an FMDP. As mentioned earlier, FMDP separates the transitions and costs into their counterparts defined on small sets of elements in the state vector. As a result, FMDP can reduce the computational complexity in determining an optimal maintenance policy for multi-component systems.

Let $\mathcal{K} \triangleq \{1, \dots, K\}$ be the set of all components. Degradation of the components is described by a controlled Markov process $(\mathcal{S}, \mathcal{A}, P)$. The term "controlled" means that the degradation paths of the components can be influenced by the decision maker through maintenance actions. Specifically, $\mathcal{S} \triangleq \{1, \dots, N\}^K$ is the set of all possible state vectors of all components. For a state vector $x = (x^1, \dots, x^K) \in \mathcal{S}$, x^k is the state of an individual component $k \in \mathcal{K}$; a higher value of state indicates a healthier condition. $\mathcal{A} \subset \{0, 1\}^K$ is the action set with $a = (a^1, \dots, a^K)$ being a generic element thereof. In particular, a^k represents the action taken on component $k \in \mathcal{K}$, with $a^k = 1$ and $a^k = 0$ representing "replacement" and "do nothing", respectively. We impose a restriction upon $\mathcal{A}$ to reflect the limited availability of maintenance resources, particularly the crews executing maintenance actions. At each inspection, due to the limited maintenance crews, only a portion of components can be maintained; specifically, for all $a = (a^1, \dots, a^K) \in \mathcal{A}$, we impose $\sum_{k\in\mathcal{K}} a^k \le A$, where A is a pre-specified constant satisfying $A \ll K$. It is worth noting that in extreme cases A can be as big as K, especially when the system scale is not large. Nonetheless, our approach well accommodates the case of $A = K$.

Furthermore, P is the controlled transition matrix of the states, where $P(y \mid x, a)$ is the transition probability from state $x \in \mathcal{S}$ to state $y \in \mathcal{S}$ under action $a \in \mathcal{A}$. As discussed earlier, the transition of a component's state between two successive inspections can be attributed to maintenance at inspection and/or natural degradation during the inspection interval. Specifically, suppose that action a is taken at inspection. Then, the state of each component $k \in \mathcal{K}$ makes an immediate transition according to $\{P_R^k(z^k \mid x^k, a^k)\}_{k\in\mathcal{K}}$, $\forall z \in \mathcal{S}$, where $P_R^k(z^k \mid x^k, a^k)$ characterizes the transition probability, for component k, from state x^k to state z^k under action a^k. In particular, when the "do nothing" action is taken (i.e., $a^k = 0$),

$$P_R^k(z^k \mid x^k, 0) = \begin{cases} 1, & z^k = x^k, \\ 0, & z^k \in \{1, \dots, N\} \setminus \{x^k\}. \end{cases}$$

Upon the next inspection, the system transitions from state z right after the previous inspection and maintenance, if any, to a new state y according to $P_D(y \mid z)$, $\forall y \in \mathcal{S}$, which reflects the transition of system state under natural degradation during the inspection interval. We consider that degradation of each component is affected only by a small set (relative to $\mathcal{K}$) of its neighbouring components, which can be represented in a rigorous way as follows. Here and thereafter, $\forall \mathcal{K}' \subset \mathcal{K}$, let $z(\mathcal{K}')$ (resp. $\mathcal{S}(\mathcal{K}')$) denote the elements of $z \in \mathcal{S}$ (resp. $\mathcal{S}$) with indices in $\mathcal{K}'$. For each $k \in \mathcal{K}$, there exists an integer $1 \le d^k \le K$ such that if we define $\mathcal{U}_k = \{k' \in \mathcal{K} : |k' - k| \le d^k\}$, then

$$P_D(y \mid z) = \Pi_{k\in\mathcal{K}} P_D^k(y^k \mid z(\mathcal{U}_k)),\ \forall z,\ y \in \mathcal{S},$$

where $P_D^k : \mathcal{S}(\mathcal{U}_k) \times \mathcal{A}$ is the local state transition probability of a single component k. Following the paradigm of dynamic Bayesian network, we refer to each $\mathcal{U}_k$ as the parent set of the kth component.

Combining P_R^k and P_D^k, $\forall k$, the transition function P can be represented by

$$P(y \mid x, a) = \sum_{z\in\mathcal{S}} \left(\Pi_{k\in\mathcal{K}} P_R^k\left(z^k \mid x^k, a^k\right)\right) \cdot \left(\Pi_{k\in\mathcal{K}} P_D^k\left(y^k \mid z\left(\mathcal{U}_k\right)\right)\right). \quad (1)$$

After an action $a^k \in \{0, 1\}$ is carried out on a single component $k \in \mathcal{K}$ at state $x^k \in \mathcal{S}$, a normalized instant cost $c^k(x^k, a^k)$ is incurred, where $c^k : \{1, \dots, N\} \times \{0, 1\} \to [0, 1]$ is an increasing function of a^k for given x^k. However, when multiple components are maintained simultaneously, there might be significant economic dependence between adjacent components. To be specific, if two components are adjacent (say, k and $k+1$), then the cost of maintaining them together should be less than that of maintaining them individually. To characterize such effect, we define a penalty cost on top of the marginal maintenance costs of each individual components, and the total cost is defined as the combination of both. In particular, we say that components $k_1 \in \mathcal{K}$ and $k_2 \in \mathcal{K}$ are adjacent if $|k_1 - k_2| = 1$. For any nonempty subsets $\mathcal{K}_1 \subset \mathcal{K}$ and $\mathcal{K}_2 \subset \mathcal{K}$, we say that $\mathcal{K}_1$ and $\mathcal{K}_2$ are attached if $\mathcal{K}_1 \ne \mathcal{K}_2$ and there exist components $k_1 \in \mathcal{K}_1$ and $k_2 \in \mathcal{K}_2$ such that k_1 and k_2 are adjacent; otherwise, we say that $\mathcal{K}_1$ and $\mathcal{K}_2$ are detached. For any $a \in \mathcal{A}$, let $\{k \in \mathcal{K} : a^k = 1\}$ be the set of components that need active maintenance under action a; further suppose that $\{k \in \mathcal{K} : a^k = 1\}$ is the union of exactly $D(a)$ mutually detached sets of components, none of which can be further separated into multiple detached sets. We can then define a cost function $\varphi : \mathcal{A} \to \mathbb{R}^+$ as $\varphi(a) = \rho \cdot D(a)$, where ρ is an instance-free constant. We let $c(x, a) \triangleq \sum_{k\in\mathcal{K}} c^k(x^k, a^k) + \varphi(a)$ denote the total cost of applying action a when the state vector is x. This formulation implies that maintenance crews can maintain detached sets within the capacity constraint A; however, a penalty cost $\varphi(a)$ would be incurred to reflect the additional efforts needed to maintain detached sets that are distanced.

Denote $\pi : \mathcal{S} \to \mathcal{A}$ as a policy and Π as the set of all such policies. At each inspection epoch $t = 1, 2, \dots$, the decision maker observes a state vector x_t and takes an action $a_t = \pi(x_t)$ that incurs cost $c(x_t, a_t)$. In the context of sequential decision making, the long-term performance of a policy π given an initial state $x \in \mathcal{S}$ is measured by the following value function:

$$V^\pi(x) = \mathbf{E}_\pi\left(\sum_{t=1}^{\infty} \gamma^{t-1} c(x_t, a_t) \,\middle|\, x_1 = x\right),$$

where $0 < \gamma < 1$ is the discount factor. The objective is to find an optimal policy π^* that minimizes the value function, namely,

$$\pi^*(x) = \arg\min_{\pi\in\Pi} V^\pi(x),\ \ \forall x \in \mathcal{S}.$$

The optimal policy can be evaluated through the classical value iteration approach (Puterman, 2014). In particular, the optimal value function under π^*, denoted by V^* for brevity, can be evaluated by

$$V^*(x) = \min_{a\in\mathcal{A}} \left\{ c(x,\, a) + \gamma \sum_{y\in S} P(y \mid x,\, \pi(x)) V^*(y) \right\},\ \forall x \in S. \tag{2}$$

In addition, the optimal policy can be represented in a compact way using the state–action value function $Q^* : S \times \mathcal{A}$ of π^* defined as

$$\begin{aligned} Q^*(x,\, a) &= c(x,\, a) + \gamma \sum_{y\in S} P(y \mid x,\, a) V^*(y) \\ &= c(x,\, a) + \gamma \sum_{y\in S} P(y \mid x,\, a) \min_{a'\in\mathcal{A}} Q^*(y,\, a'),\ \forall x \in S,\, a \in \mathcal{A} \end{aligned}$$

and $\pi^*(x) = \arg\min_{a\in\mathcal{A}} Q^*(x,\, a)$ for any $x \in S$.

Though the optimal policy can be numerically calculated by Eq. (2), the dynamic programme suffers from high computational complexity due to the curse of dimensionality, and is thus considered to be computationally intractable even for moderate-scale problems. However, in many real-world maintenance problems, there can be hundreds of components and the exponentially large state space prohibits a feasible computation process. In what follows, we take advantage of the crucial structural properties of the problem to design an effective algorithm that can significantly reduce the computational difficulty.

4. Modified factored value iteration algorithm

The factorization properties of FMDP, if well exploited, can significantly reduce computational complexity in determining an optimal policy. (Szita & Lörincz, 2008) propose the factored value iteration (FVI) approach by combining the factorization properties with the classical value iteration (Guestrin et al., 2003). FVI has been proven to be an efficient method for FMDPs. In this section, we first modify the original FVI approach to construct an efficient planning algorithm that can obtain an approximate optimal policy. We then prove the efficiency of the constructed algorithm from the perspectives of both computational complexity and approximation bias.

The FVI approach first approximates the real value function by a linear combination of a set of basis functions $h_m : S \to \mathbb{R}$, $m = 1, \ldots, M$, where h_m is specified such that it relies only on a small set of elements in S. That is, for all $m = 1, \ldots, M$, there exists $\mathcal{V}_m \subset \{1, \ldots, K\}$ such that $h_m(x)$ relies only on $x(\mathcal{V}_m)$, $\forall x \in S$. For notational convenience, we use $h_m(x(\mathcal{V}_m))$ to characterize the dependence. Define H as an $|S| \times M$ matrix with entries $H_{x,\,m} = h_m(x(\mathcal{V}_m))$ for all $x \in S$ and $m = 1, \ldots, M$. The objective of FVI is to determine a weight vector $w \in \mathbb{R}^M$ such that $H \cdot w$ is close to V^* under some metric. Let c^a be an $|S|$-dimensional cost vector with entries $c^a_x = c(x,\, a)$ and P^a be an $|S| \times |S|$ transition matrix with entries $P^a_{x,\,y} = P(y \mid x,\, a)$. For any $a \in \mathcal{A}$, the optimal weight w^* can be obtained by solving the following fixed-point problem:

$$w^* = G\left[\min_{a\in\mathcal{A}} \left(\sum_{k=1}^{K} c^{a,\,k} + \varphi(a) + \gamma B^a \cdot w^* \right)\right], \tag{3}$$

where G is a linear operator from the space of value functions to the linear space $\mathcal{L}(H)$ expanded by $h_1, \ldots, h_M$, represented by a matrix in $\mathbb{R}^{M\times|S|}$ that satisfies the following non-expansion property:

$$\left\| HGv - HGv' \right\|_\infty \le \left\| v - v' \right\|_\infty,\ \forall v,\, v' \in \mathbb{R}^{|S|},$$

with $\|\cdot\|_\infty$ being the sup-norm; $c^{a,\,k}$ is an $|S|$-dimensional local cost vector with entries $c^{a,\,k}_x = c^k(x^k,\, a^k)$; B^a is a matrix with entries

$$\begin{aligned} B^a_{x,\,m} = \sum_{y(\mathcal{V}_m)\in S(\mathcal{V}_m)} &\left[\left(\Pi_{k\in\cup_{k'\in\mathcal{V}_m}\mathcal{U}_{k'}} P^k_R\left(z^k \mid x^k,\, a^k\right) \right) \right. \\ &\left. \cdot \left(\Pi_{k\in\mathcal{V}_m} P^k_D\left(y^k \mid z(\mathcal{U}_k)\right) \right) \right] h_m\left(\mathcal{V}_m\right). \end{aligned}$$

Then, a sampling technique is used to cope with the computational complexity caused by the scales of the matrices in Eq. (3), which are $O(|S|)$ and can still be prohibitively large. We sample a subset of state vectors $\hat{S}$ from S and confine the calculation to $\hat{S}$, to formulate an approximation of w^*. It is apparent that the sample size $|\hat{S}|$ influences the efficiency of approximation. Though there is no universal approach to specifying the sample size across different problems, there are indeed some routines to follow in practice. An important routine is that the sample size should be sufficiently large while keeping polynomial in K, so that the approximation accuracy and the computational efficiency can be well balanced. We denote by $\hat{G}$, $\hat{c}^{a,\,k}$, and $\hat{B}^a$ the sub-matrices of G, $c^{a,\,k}$, and B^a, respectively, with rows corresponding to $\hat{S}$. An approximation of w^* (i.e., $\hat{w}^*$) can be evaluated by

$$\hat{w}^* = \hat{G}\left[\min_{a\in\mathcal{A}} \left(\sum_{k=1}^{K} \hat{c}^{a,\,k} + \varphi(a) + \gamma \hat{B}^a \cdot \hat{w}^* \right)\right]. \tag{4}$$

Since the scales of $\hat{S}$ and $\mathcal{A}$ are both polynomial in K, Eq. (4) becomes computationally tractable. The procedures of producing an approximate optimal policy are summarized in Algorithm 1.

We now derive a bound on the approximation error, in terms of a bound on the difference between the value function of the optimal policy (i.e., V^*) and that of the approximation (i.e., $H\hat{w}^*$). To this end, we make Assumption 1 on G. By this assumption, we shall show in Theorem 1 that the bias can be bounded using only a sampled set $\hat{S}$ with a carefully determined size which is polynomial in K.

Assumption 1. Let $\mathcal{W}_1, \ldots, \mathcal{W}_E \subset \mathcal{K}$ be exclusive sets of component indices such that $\mathcal{W}_1 \cup \cdots \cup \mathcal{W}_E = \mathcal{K}$. We assume that G can be separated as $G = \sum_{e=1}^{E} G_e$, where each G_e is an $M \times |S|$ and $\mathcal{W}_e$-scope matrix, namely, each row of G_e, considered as a function on S, relies only on $S(\mathcal{W}_e)$. Moreover, we suppose that $K_W = \max_{e\in\mathcal{E}} |\mathcal{W}_e| \ll K$.

We denote by π_0 the greedy policy using Hw^*, and c^{π_0} and P^{π_0} the cost function and transition matrix induced by π_0. Further define a matrix B^{π_0} along with B^a, with entries

$$\begin{aligned} B^{\pi_0}_{x,\,m} = \sum_{y(\mathcal{V}_m)\in S(\mathcal{V}_m)} &\left[\left(\Pi_{k\in\cup_{k'\in\mathcal{V}_m}\mathcal{U}_{k'}} P^k_R\left(z^k \mid x^k,\, \pi_0(x)^k\right) \right) \right. \\ &\left. \cdot \left(\Pi_{k\in\mathcal{V}_m} P^k_D\left(y^k \mid z^k\right) \right) \right] h_m\left(\mathcal{V}_m\right). \end{aligned}$$

Following the definition of $B^{\pi_0}_{x,\,m}$, Eq. (3) can be rewritten as

$$w^* = G\left[c^{\pi_0} + \gamma B^{\pi_0} \cdot w^*\right].$$

Moreover, we can separate matrix B^{π_0} as

$$B^{\pi_0} = \sum_{m=1}^{M} B^{\pi_0,\,m}, \tag{5}$$

Algorithm 1 Modified factored value iteration

Input: S, $\mathcal{A}$, M, φ, G, c^k, P^k_R, P^k_D, $\forall k \in \mathcal{K}$, $0 < \epsilon < 1$.
Initialization: Generate H and randomly sample $\hat{S}$ from S.
Calculate $\hat{G}$, $\hat{c}^{a,\,k}$, $\hat{B}^a$, $\forall k \in \mathcal{K}$, $\forall a \in \mathcal{A}$.
$n \leftarrow 0$, $w_0 \leftarrow \vec{0}$, $\Delta \leftarrow 1$.
while $\Delta > \epsilon$ **do**
  Calculate w_{n+1} by

$$w_{n+1} = \hat{G}\left[\min_{a\in\mathcal{A}} \left(\sum_{k=1}^{K} \hat{c}^{a,\,k} + \varphi(a) + \gamma \hat{B}^a \cdot w_n \right)\right].$$

  $\Delta \leftarrow \left\| w_{n+1} - w_n \right\|_\infty$.
  $n \leftarrow n + 1$
end while
Output: Policy $\pi(x)$ defined by

$$\pi = \arg\min_{a\in\mathcal{A}} \left(\sum_{k=1}^{K} \hat{c}^{a,\,k} + \varphi(a) + \gamma \hat{B}^a \cdot w_n \right).$$

where, for any $m = 1, \ldots, M$, $B^{\pi_0, m}$ is the product of P^{π_0} with an $|S| \times M$ matrix that keeps the mth column of H and sets other entries to 0. Similar to the entries of matrix B^a, entries of $B^{\pi_0, m}$ is $B^{\pi_0, m}_{x, m} = B^{\pi_0}_{x, m}$ and $B^{\pi_0, m}_{x, m'} = 0$ if $m' \neq m$. It is easy to verify that each $B^{\pi_0, m}$ is a $\cup_{k \in \mathcal{V}_m} \mathcal{U}_k$-local scope matrix. Under Assumption 1 and using Eq. (5), the following theorem provides performance guarantee of Algorithm 1.

Theorem 1. *For any $0 < \delta < 1$ and $\varepsilon > 0$, when the size of $\hat{S}$ satisfies*

$$|\hat{S}| \geq \frac{2\left(1 + \gamma \cdot \|w^*\|_\infty\right)^2 \|H\|_\infty^2 M^2}{(1-\gamma)^2 \varepsilon^2} \log \frac{4M^2}{\delta} \max\left\{\Psi_1, \Psi_2\right\}, \tag{6}$$

where

$$\Psi_1 = \left[E \cdot K \cdot N^{K_W} \max_{e=1,\ldots,E} \|G_e\|_\infty \max_{k, x^k, a^k} c^k(x^k, a^k)\right]^2,$$

and

$$\Psi_2 = \left[E \cdot K \cdot N^{\max\{K_W, \max_{k \in \mathcal{K}} d^k + \max_{m=1,\ldots,M} |\mathcal{V}_m|\}} \max_{e=1,\ldots,E} \|G_e\|_\infty \max_{m=1,\ldots,M} \|B^{\pi_0, m}\|_\infty\right]^2,$$

with probability at least $1 - \delta$, we have

$$\left\|H\hat{w}^* - V^*\right\|_\infty \leq \left\|Hw^* - V^*\right\|_\infty + \varepsilon. \tag{7}$$

Theorem 1 provides a lower bound on the sample size $|\hat{S}|$, which guarantees that Algorithm 1 approximates the true optimal value function V^* (and the action-value function Q^*) well enough. It is worth noting that the lower bound is not necessarily polynomial in K, because it still relies on $\max_e \|G_e\|_\infty$ and $\max_m \|B^{\pi_0, m}\|_\infty$, with scale $M \times |S|$. This issue can be addressed by carefully choosing matrix H. Nevertheless, Algorithm 1 is based on classical techniques for high-dimensional MDPs, while specifying an appropriate H can be flexible yet challenging in different problems. One can follow some general routines for choosing H. For example, one may choose H such that the value of each h_m relies only on a very limited number of elements in the state space. In our simulation study, we define each h_m as a categorical function on a single component. Existing research has shown that such basis functions deliver high computational efficiency and low approximation error. In Section 6, we shall show that such choice of H ensures that $\max_e \|G_e\|_\infty$ and $\max_m \|B^{\pi_0, m}\|_\infty$ are delicately bounded, so that $|\hat{S}|$ is small enough and Algorithm 1 is computationally tractable when $\max_{k \in \mathcal{K}} d^k$ and $\max_{m=1,\ldots,M} |\mathcal{V}_m|$ are relatively small. In the subsequent sections, we always admit a projection matrix G that satisfies the restrictions in our previous discussions.

5. An online learning perspective

The modified FVI algorithm developed previously is based on an implicit assumption that state transitions of the system are fully known to the decision maker, which is usually not the case in real applications. In this section, we tackle the condition-based group maintenance problem of interest from an online learning perspective in which the exact transitions are unknown *a priori*. In this setting, the decision maker determines an optimal maintenance policy upon the arrival of a new inspection data, while simultaneously learning the true model parameters using historical observations. This leads to the so-called *exploration-exploitation tradeoff* in reinforcement learning (Xu et al., 2021).

In the online group maintenance problem, we need to learn both the transition probabilities of the components and the structure $\{d^k\}_{k \in \mathcal{K}}$ of the transitions. We thus introduce an additional assumption on the structure. Specifically, we assume that a uniform upper bound κ on $\{d^k\}_{k \in \mathcal{K}}$, instead of their true values, is known; that is, we have $\kappa \geq \max_{k \in \mathcal{K}} d^k$. This implies that the decision maker has a crude knowledge on the maximum number of neighbouring components that can affect the degradation of any specific component, which is fairly reasonable in practical scenarios. Such an upper bound can be established by expert judgements or estimated from historical data, if available.

Developing learning algorithms for FMDPs under unknown transition structure has been an active research topic. A recent and significant progress has been made by Rosenberg and Mansour (2021), who propose a novel approach to finding exact positions of parent sets with fixed and known sizes. Our problem setting differs from that of Rosenberg and Mansour (2021) in two aspects: First, in our setting the positions of parent sets are known but the sizes $\{d^k\}_{k \in \mathcal{K}}$ are unknown. Second, our model uses the discounted total cost as the objective function, while Rosenberg and Mansour (2021) focus on the average cost. Nevertheless, their approach provides the foundation upon which modification can be made to solve our maintenance problem. On the other hand, value iteration-type learning algorithms have been proven to be efficient for discounted MDPs by Strehl (2007), albeit known transition structure is assumed therein. In particular, Strehl (2007) develops the fMBIE method that can effectively address the exploration-exploitation tradeoff for model-based reinforcement learning. In this work, we develop an online algorithm to approximate the optimal maintenance policy π^* by combining the online approach of learning transition structure (see Rosenberg & Mansour, 2021) and the fMBIE approach for discounted MDPs (see Strehl, 2007).

To this end, we first introduce a performance metric for online learning algorithms. The objective of an online algorithm is to gradually approximate some optimal policy; therefore, the necessary samples (time steps) for the algorithm to generate some policy that is close enough to the optimal policy is crucial for the algorithm's performance. In particular, an efficient online algorithm is supposed to generate policies with value functions ε-close to that of the optimal policy with high probability within a time at most polynomial in $1/\varepsilon$ and some other parameters of the underlying model. An online algorithm that satisfies this property is called an efficient *Probably Approximate Correct* (PAC) algorithm. A formal definition of efficient PAC algorithms for FMDPs can be given based on the sample complexity defined below. We assume here that $\{h_m\}_{m=1}^M$ and $\{\mathcal{V}_m\}_{m=1}^M$ are specified beforehand.

Definition 1 (*Sample Complexity*)**.** For any $\varepsilon > 0$, the sample complexity of an online algorithm for FMDP $(S, \mathcal{A}, P, \kappa, \gamma, c)$ is the number of time steps such that the sequence of policies generated by the algorithm, denoted by $\{\pi_t\}_{t=1}^\infty$, satisfies $V^{\pi_t}(x_t) < V^{\pi^*}(x_t) + \varepsilon$.

An online algorithm for FMDP $(S, \mathcal{A}, P, \kappa, \gamma, c)$ is called an efficient PAC learning algorithm if for any $\varepsilon > 0$ and $0 < \delta < 1$, the per-step computational complexity and sample complexity can be bounded by some polynomial in the relevant parameters $(1/\varepsilon, 1/\delta, 1/(1-\gamma), K, N^{2\kappa+1}, |\mathcal{A}|)$ with probability at least $1 - \delta$. It should be highlighted that the sample complexity is required to be polynomial in $N^{2\kappa+1}$, instead of $|S| = N^K$ required for efficient PAC algorithms for general MDPs. This is because FMDP has a factored structure, so that an efficient online algorithm can collect sufficient samples from each factor to approximate the true model well enough.

5.1. An online PAC learning algorithm

We are now in a position to construct an online learning algorithm to support condition-based group maintenance decision making, which is proven to be an efficient PAC algorithm for our FMDP model. At each inspection, the proposed algorithm proceeds in two steps. In the first step, the algorithm leverages the value iteration method to update state–action value functions and generate a policy to recommend an action (replacement or do nothing). In the second step, the algorithm utilizes previous samples to evaluate the transitions as well as the exact value of $\{d^k\}_{k \in \mathcal{K}}$.

To proceed, we list below the necessary quantities observed or calculated by the algorithm up to epoch $t \geq 1$ during execution.

- $n_t(x^k, a^k)$: number of times action a^k is taken on component k when it is in state $x^k \in \{1, \ldots, N\}$;

- $n_t(x^k, a^k, y^k)$: number of times component k transitions from state $x^k \in \{1, \dots, N\}$ to state $y^k \in \{1, \dots, N\}$ when action a^k is taken;
- $n_t(x(\mathcal{U}))$: number of times a set $\mathcal{U} \subset \mathcal{K}$ of components are observed to be in state vector $x(\mathcal{U}) \in S(\mathcal{U})$ immediately after actions are taken;
- $n_t(x(\mathcal{U}), y^k)$: number of times component k is observed to be in state $y^k \in \{1, \dots, N\}$ upon the next inspection, given that a set $\mathcal{U} \subset \mathcal{K}$ of components are observed to be in state vector $x(\mathcal{U}) \in S(\mathcal{U})$ immediately after actions are taken;
- $\hat{P}^k_{R,t}(y^k \mid x^k, a^k) \triangleq n_t(x^k, a^k, y^k)/\max\{n_t(x^k, a^k), 1\}$: estimated transition probability of component k under action a^k;
- $\hat{P}^k_{D,t}(y^k \mid x(\mathcal{U})) \triangleq n_t(x(\mathcal{U}), y^k)/\max\{n_t(x(\mathcal{U})), 1\}$: estimated transition probability of component k during the degradation interval (when considering $\mathcal{U} \subset \mathcal{K}$ to be $\mathcal{U}_k$);

Since the exact value of d^k in the online setting is unknown, we consider $\mathcal{U}_k$ as a function $\mathcal{U}_k(d^k)$ of d^k, $k \in \mathcal{K}$. The algorithm retains a current estimate d^k_t with initial value $d^k_1 = 0$. According to the fMBIE approach, the algorithm only uses the first τ_R samples of each state–action pair (x^k, a^k) and the first τ_D samples of each factored state $x(\mathcal{U}_k(\kappa))$. The algorithm stops recording the observations of (x^k, a^k) or $x(\mathcal{U}_k(\kappa))$, when some $n_t(x^k, a^k)$ reaches τ_R or $n_t(x(\mathcal{U}))$ reaches τ_D, whichever happens first.[1]

In the first step of epoch $t \geq 1$, the algorithm calculates estimated transition probabilities $\hat{P}^k_{R,t}(y^k \mid x^k, a^k)$ and $\hat{P}^k_{D,t}(y^k \mid x(\mathcal{U}(d^k_t)))$ for all $k \in \mathcal{K}$, x^k, y^k, x, and y using historical data. Note that our algorithm follows a standardized online reinforcement learning pattern, in which historical data contain the sequence of system states and the corresponding actions taken in preceding epochs. Such data are available in most practical maintenance scenarios. After a sufficient amount of samples, as indicated in Theorem 1, have been collected, the algorithm will consider the model to be fully learned and stop learning anymore. However, if no historical data is available, then domain knowledge would be needed to estimate the transition probabilities.

Then, the algorithm calculates the estimated overall transition probabilities, denoted by $\hat{P}_t$, via

$$\hat{P}_t(y \mid x, a) = \sum_{z \in S} \left(\Pi_{k \in \mathcal{K}} \hat{P}^k_{R,t}\left(z^k \mid x^k, a^k\right) \right) \cdot \left(\Pi_{k \in \mathcal{K}} \hat{P}^k_{D,t}(y^k \mid z(\mathcal{U}_k(d^k_t))) \right), \ \forall x, y \in \mathcal{S}, a \in \mathcal{A}, \tag{8}$$

and the state–action value function $\hat{Q}_t$ by

$$\hat{Q}_t(x, a) = c(x, a) + \gamma \sum_{y \in S} \hat{P}_t(y \mid x, a) \min_{a' \in \mathcal{A}} \hat{Q}_t(y, a') - \beta_t(x, a), \ \forall x \in \mathcal{S}, a \in \mathcal{A}, \tag{9}$$

where $\beta_t : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is an exploration bonus to balance the exploitation-exploration tradeoff. We note here that $\beta_t(\cdot, \cdot)$ should be determined such that it is factored and $\hat{Q}_t$ can be effectively solved by Algorithm 1. Next, the algorithm takes an action greedily on the current state vector x_t; that is, $a_t = \arg\min_{a \in \mathcal{A}} \hat{Q}_t(x_t, a)$.

In the second step of epoch $t \geq 1$, the algorithm utilizes historical observations to make a judgement if $d^k_t < d^k$ holds with high probability. Specifically, the algorithm checks in each epoch $t \geq 1$, $\forall k \in \mathcal{K}$, if there exists $d^k_t < d \leq \kappa$ such that the following condition is satisfied:

$$\left\| \hat{P}^k_{D,t}(\cdot \mid x(\mathcal{U}_k(d))) - \hat{P}^k_{D,t}(\cdot \mid x(\mathcal{U}_k(\kappa))) \right\|_1 > 2\sqrt{\frac{2\ln\left\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_D/\delta\right\} + 2N^{2\kappa+1}\ln(2)}{\max\{n_t(x(\mathcal{U}_k(\kappa))), 1\}}}, \ \forall x \in \mathcal{S}, \tag{10}$$

where $\|\cdot\|_1$ is the L_1-norm. If condition (10) holds for some $k \in \mathcal{K}$ and let $d^k_t < d \leq \kappa$ be the largest value that makes (10) valid, then the algorithm updates $d^k_{t+1} \leftarrow d+1$ and proceeds to the next epoch $t+1$. The procedures are summarized in Algorithm 2, and the algorithm breaks the ties at any time.

[1] The exact values of τ_R and τ_D shall be specified in Theorem 2.

Algorithm 2 Efficient online PAC learning algorithm

Input: N, A, K, c, γ.
Initialization: $t = 1$, initial state x_1, $d^k_1 = 0$;
$\forall k \in \mathcal{K}$, x^k, y^k and $a \in \mathcal{A}$, $n_1(x^k, a^k) = 0$, $n_1(x^k, a^k, y^k) = 0$;
$\forall d \leq \kappa$, $n_1(x_1(\mathcal{U}_k(d))) = 1$, $n_1(x_1(\mathcal{U}), y^k) = 1$, $\forall x(\mathcal{U}_k(d)) \neq x_1(\mathcal{U}_k(d))$, $n_1(x(\mathcal{U}_k(d))) = 0$, $n_1(x(\mathcal{U}), y^k) = 0$;
$\forall k \in \mathcal{K}$, x, y^k and $a \in \mathcal{A}$, $\hat{P}^k_{R,1}(y^k \mid x^k, a^k) = 0$, $\hat{P}^k_{D,1}(y^k \mid x(\mathcal{U}_k(d))) = 0$.
while (1) **do**
  Step 1: Apply Algorithm 1 to calculate $\hat{Q}_t$ and take action $a_t = \arg\min_{a \in \mathcal{A}} \hat{Q}_t(x_t, a)$.
  Observe state vector y_t immediately after action a_t is taken.
  for $k \in \mathcal{K}$ **do**
    if $n_t(x^k_t, a^k_t) < \tau_R$ **then**
      $n_t(x^k_t, a^k_t) \leftarrow n_{t-1}(x^k_t, a^k_t)+1$, $n_t(x^k_t, a^k_t, y^k_t) \leftarrow n_t(x^k_t, a^k_t, y^k_t)+1$.
    end if
    $n_t(y_t(\mathcal{U}_k(d))) \leftarrow \min\{n_{t-1}(y_t(\mathcal{U}_k(d))) + 1, \tau_D\}$
  end for
  Step 2: Observe x_{t+1} upon the next inspection.
  for $k \in \mathcal{K}$ **do**
    for $d^k_t \leq d \leq \kappa$ **do**
      if $n_t(y_t(\mathcal{U}_k(d))) < \tau_D$ **then**
        $n_{t+1}(y_t(\mathcal{U}_k(d)), x^k_{t+1}) \leftarrow n_t(y_t(\mathcal{U}_k(d)), x^k_{t+1}) + 1$.
      end if
    end for
  end for
  for $k \in \mathcal{K}$ **do**
    if There exists $d^k_t < d \leq \kappa$ such that (10) holds and d is the largest of such values **then**
      Update $d^k_{t+1} \leftarrow d+1$.
    end if
  end for
  Calculate $\hat{P}^k_{R,t+1}(y^k \mid x^k, a^k) = \frac{n_{t+1}(x^k, a^k, y^k)}{\max\{n_t(x^k, a^k), 1\}}$, $\forall k \in \mathcal{K}$, x^k, y^k.
  Calculate $\hat{P}^k_{D,t+1}(y^k \mid x(\mathcal{U}_k(d^k_{t+1}))) = \frac{n_t(x(\mathcal{U}_k(d^k_{t+1})), y^k)}{\max\{n_t(x(\mathcal{U}_k(d))), 1\}}$, $\forall k \in \mathcal{K}$, x^k, y^k.
  $t \leftarrow t+1$.
end while

5.2. Sample complexity

We now derive an upper bound on the sample complexity of Algorithm 2, which is polynomial in the relevant model parameters; thus, we can claim that the proposed algorithm is an efficient PAC algorithm. The main theoretical result on the sample complexity is presented below.

Theorem 2. *Given* $\varepsilon > 0$ *and* $0 < \delta < 1$, *if the exploration bonus* β_t *is chosen as*

$$\beta_t(x, a) = \frac{(1+\rho)A \cdot \gamma}{1-\gamma} \left(K \cdot \max_{k \in \mathcal{K}} \cdot \max_{(x^k, a^k)} \times \sqrt{\frac{2\ln\left\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_R/\delta\right\} + 4N\ln(2)}{\max\{n_t(x^k, a^k), 1\}}} + N^A \cdot K \cdot \max_{k \in \mathcal{K}} \max_{x(\mathcal{U}_k(\kappa))} \times \sqrt{\frac{2\ln\left\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_D/\delta\right\} + 2N^{2\kappa+1}\ln(2)}{\max\{n_t(x(\mathcal{U}_k(\kappa))), 1\}}} \right),$$

and τ_D *and* τ_R *are chosen as*

$$\tau_R = \frac{128(2-\gamma)^2\gamma^2(1+\rho)^2A^2K^2}{(1-\gamma)^6\varepsilon^2} \times \left[2\ln\left\{K\cdot N\left[N^{2\kappa+1}+2\right]/\delta\right\} + 4\ln\left(\frac{16(2-\gamma)\gamma(1+\rho)A\cdot K}{(1-\gamma)^3\varepsilon}\right) + 4N\right],$$

and

$$\tau_D = \frac{128(2-\gamma)^2\gamma^2(1+\rho)^2A^2N^{2A}K^2}{(1-\gamma)^6\varepsilon^2}\left[2\ln\left\{K\cdot N\left[N^{2\kappa+1}+2\right]/\delta\right\} + 4\ln\left(\frac{16(2-\gamma)\gamma(1+\rho)A\cdot N^AK}{(1-\gamma)^3\varepsilon}\right) + 2N^{2\kappa+1}\right],$$

then the sample complexity of Algorithm 2 can be bounded by

$$O\left(\frac{N\cdot K\cdot\tau_R + N^{2\kappa+1}\cdot K\cdot\tau_D}{\varepsilon(1-\gamma)^2}\ln\frac{1}{\delta}\ln\frac{1}{\varepsilon(1-\gamma)}\right),$$

with probability at least $1-\delta$.

Here we provide the sketch of the proof (detailed proof can be found in Appendix B). We first construct in Lemma 1 a "good event" in that the estimation biases of $\hat{P}_R^k$ and P_R^k shrink with high probability as the sample size increases. Then, restricted to this event, we derive a bound on the estimation bias of the total transition P_t. This bound can be directly transferred to a bound on the estimation bias of the state–action value function Q. Finally, we integrate all these results and follow the existing techniques to complete the proof.

We argue that Algorithm 2 follows the traditional paradigm of PAC learning. The coefficients τ_R and τ_D are pre-determined thresholds based on some known model parameters. During the execution of Algorithm 2, for each single component, only the first τ_R samples under the same action taken at inspection and the first τ_D samples in the degradation interval are observed and utilized. After that, Algorithm 2 considers the model to be fully learned and neglects any new data. The exploration bonus β_t is set to balance between exploiting the "optimal" decision based on the latest data (exploitation) and checking if there are less-explored decisions that can be better identified (exploration).

6. Numerical studies

In this section, we present numerical studies to demonstrate the effectiveness of the proposed algorithms. For this purpose, we consider a hypothetical system with multiple non-identical components. The components are periodically inspected, and the degradation level of each component is discretized into three states for illustration. Specifically, state 3 represents the perfect state in which no maintenance action is needed; state 1 is the failure state that induces corrective maintenance; while the decision maker needs to decide whether to implement a preventive maintenance if a component is in state 2.

To implement our approach, the first step is to specify the matrix H of basis functions in Algorithm 1. Plenty of work has examined different basis functions for such problems, and a set of basis functions given below have been proven to be both simple and efficient (Guestrin et al., 2003; Osband & Van Roy, 2014; Xu & Tewari, 2020):

$$h_k(x) = h_k(x^k) = \begin{cases} 0, & x^k = 3,\\ \frac{1}{2}, & x^k = 2,\\ 1, & x^k = 1,\end{cases} \quad k = 1,\dots,K.$$

Specifically, a set of K (i.e., $M = K$) basis functions are defined and each basis function h_k relies only on x^k. Inspired by the discussion in Szita and Lörincz (2008), the following operator G is compatible with the modified FVI approach and easy to calculate in the concerned scenario:

$$G_{i,j} = \frac{H^+_{i,j}}{9K}, \tag{11}$$

where H^+ is the Moore–Penrose inverse matrix of H. We claim that the above-defined G satisfies the requirements as stated in the following proposition, and thus can be used as the projection operator in our approach.

Proposition 1. *The operator* G *defined in* (11) *satisfies the non-expansion property. Moreover, there exist* $G_1,\dots,G_K$ *such that* $G = \sum_{k=1}^K G_k$ *and* $\|G_k\|_\infty$ *can be easily bounded by* $\|G_k\|_\infty < 1/(3K+2)$.

The next step is to determine the sample size $|\hat{S}|$. This requires us to evaluate the values of Ψ_1 and Ψ_2 in Theorem 1, respectively. For B^{π_0} in Theorem 1, we have $B^{\pi_0} = \sum_{k=1}^K B^{\pi_0,k}$, where $B^{\pi_0,k}$ is the product of P^{π_0} with an $|S|\times K$ matrix that keeps the kth column of H while setting other entries to 0. As the row sum of P^{π_0} is 1 and the elements of H are either 0 or 1, the row sum of $B^{\pi_0,k}$ should be bounded by 1; that is, $\|B^{\pi_0,k}\|_\infty \le 1$. We note that all quantities needed to determine the values of Ψ_1 and Ψ_2 are either known or upper bounded. Specifically, the values of E, K, N, K_W, and $c^k(x^k, a^k)$'s are known, and $\|B^{\pi_0,k}\|_\infty$'s and $\|G_k\|_\infty$'s are bounded. Therefore, by substituting all quantities into Ψ_1 and Ψ_2 in Theorem 1, we have

$$\Psi_1 \le \frac{9K^4}{(3K+2)^2}\left[\max_{k,x^k,a^k} c^k(x^k,a^k)\right]^2, \quad \Psi_2 \le \frac{9\cdot 4^{2\kappa+1}K^4}{(3K+2)^2}.$$

We let $C_0 \triangleq \max_{k,x^k,a^k} c^k(x^k,a^k)$. Since Hw^* is the projection of V^* on $\mathcal{L}(H)$, $w^*(k)\cdot\|h_k\|_\infty \le \|V^*\|_\infty$, $\forall k\in\mathcal{K}$. Thus, we have

$$w^*(k)\cdot\|h_k\|_\infty = w^*(k)\cdot\frac{|S|}{2} \le \|V^*\|_\infty \le |S|\cdot\frac{\max_{k,x^k,a^k} c^k(x^k,a^k)}{1-\gamma} = |S|\cdot\frac{C_0}{1-\gamma},$$

which implies that

$$\|w^*\|_\infty = \sum_{k\in\mathcal{K}} w^*(k) \le \frac{2K\cdot C_0}{1-\gamma}.$$

According to Theorem 1 and the trivial fact that $\|H\|_\infty \le K$, the sample size $|\hat{S}|$ required by Algorithm 1 can be calculated as

$$\frac{18\left(1-\gamma+2\gamma\cdot K\cdot C_0\right)^2K^6M^2}{(1-\gamma)^3\varepsilon^2(3K+2)^2}\log\frac{4M^2}{\delta}\max\left\{4^{2\kappa+1}, C_0^2\right\}.$$

In what follows, we first demonstrate the preciseness of the modified FVI approach in Section 6.1, and then examine the superiority of Algorithm 2 through comparison studies in Section 6.2.

6.1. Modified FVI

We first implement Algorithm 1 presented in Section 4 and illustrate the performance of this approach. Though our algorithm is designed for large-scale maintenance problems, we choose a moderate problem size for illustrative purposes. This facilitates comparing the value function of the approximate policy generated by Algorithm 1 and that of the true optimal policy, where the latter can only be efficiently evaluated with a small or moderate problem size due to the computational complexity. First, we run Algorithm 1 on problems under different levels of K, namely, $K \in \{4, 5, 6, 7, 8, 9, 10\}$. We let components connected as a circle so that no component is positioned at the boundary. Because usually a limited number of neighbouring components are correlated in practical maintenance scenarios, without loss of generality, we fix $d^1 = \cdots = d^K = 2$ under each value of K.

Tables 1 and 2 present the transition probabilities that are specified according to the following considerations. First, the degradation process of each component leads to worse health conditions. Therefore, when $x_k = 3$, the possible values for y_k after degradation are $\{3, 2, 1\}$, whereas when $x_k = 2$, the possible values for y_k after degradation are only $\{2, 1\}$. Second, practical evidence shows that a component is more likely to degrade to a state adjacent to the current state; see, for example, the degradation process of railway tracks (Sadeghi & Askarinejad, 2010). Third, when a component is found failed upon inspection, an instant corrective maintenance would be executed to restore it back to state 3.

Table 1
State transition parameters $P_D(y^k \mid x(\mathcal{U}_k^-))$ under $x_k = 3$.

(x_{k-1}, x_k, x_{k+1})	(1, 3, 1)	(1, 3, 2)	(2, 3, 1)	(2, 3, 2)	(1, 3, 3)	(3, 3, 1)	(2, 3, 3)	(3, 3, 2)	(3, 3, 3)
$P_D(y^k = 3)$	0.5	0.6	0.6	0.7	0.7	0.7	0.8	0.8	0.9
$P_D(y^k = 2)$	0.3	0.3	0.3	0.15	0.2	0.2	0.15	0.15	0.1
$P_D(y^k = 1)$	0.2	0.1	0.1	0.15	0.1	0.1	0.05	0.05	0

Table 2
State transition parameters $P_D(y^k \mid x(\mathcal{U}_k^-))$ under $x_k = 2$.

(x_{k-1}, x_k, x_{k+1})	(1, 2, 1)	(1, 2, 2)	(2, 2, 1)	(2, 2, 2)	(1, 2, 3)	(3, 2, 1)	(2, 2, 3)	(3, 2, 2)	(3, 2, 3)
$P_D(y^k = 2)$	0.5	0.6	0.6	0.7	0.65	0.65	0.8	0.8	0.9
$P_D(y^k = 1)$	0.5	0.4	0.4	0.3	0.35	0.35	0.2	0.2	0.1

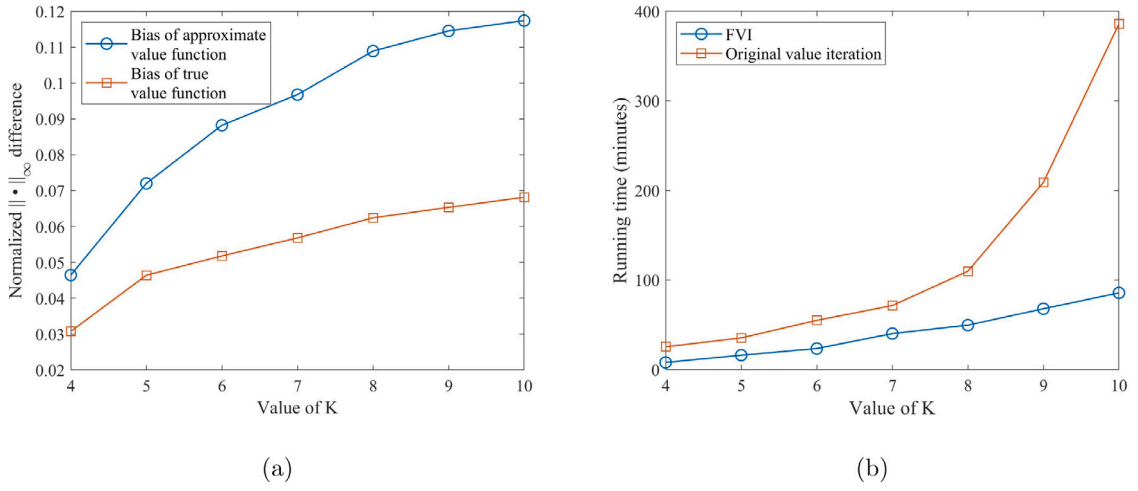

Fig. 1. The approximate error and running time of the modified FVI.

Preventive maintenance is only feasible when a component is at state 2 upon inspection. That is, $a^k \in \{0, 1\}$ if $x^k = 2$; otherwise, $a^k = 0$. We set the transition P_R^k and cost c as

$$P_R^k(z^k = 3 \mid x^k = 2,\ a^k = 1) = 0.8,\ P_R^k(z^k = 2 \mid x^k = 2,\ a^k = 1) = 0.2,$$

and

$$c^k(3, 0) = 0,\ c^k(2, 0) = 0,\ c^k(2, 1) = 2,\ c^k(1, 0) = 10,\ \forall k \in \mathcal{K}.$$

Since there are at most 10 components in the numerical example, we do not use the sampling technique. Other parameters are arbitrarily set as $\gamma = 0.8$, $\rho = 3$, and $A = 3$. In particular, 0.8 is a commonly used value for discount factor γ in the learning literature, $\rho = 3$ imposes a moderate penalty on maintaining detached components, and $A = 3$ indicates that at most 3 components are allowed to be maintained at each inspection.

Under each value of K, we evaluate the true optimal policy using the traditional value iteration method and the approximate optimal policy through Algorithm 1. The L_∞-difference between the optimal value function V^* and the approximate optimal value function Hw^* is calculated and normalized as $\|V^* - Hw^*\|_\infty / \|V^*\|_\infty$. Meanwhile, the L_∞-difference between the V^* and the true value function V^{π_0} of the greedy policy π_0 is also calculated and normalized as $\|V^* - V^{\pi_0}\|_\infty / \|V^*\|_\infty$. Since this is the bias of total return caused by using π_0 in practice, it can also be used to show the preciseness of the approximate optimal policy. The results are presented in Fig. 1(a). We can see that the bias for the approximate value function Hw^* is below 12%, which is acceptable in many practical scenarios. Meanwhile, the bias grows in a sublinear manner with the number of components K, implying that the performance of the modified FVI is robust to the problem scale. In addition, Fig. 1(a) shows that the bias for the true value function of π_0 is smaller than that for Hw^*, indicating that when used in practice, the bias of value function induced by π_0 is even smaller than estimated.

The superiority of the modified FVI lies in its low computational complexity compared with classical iterative algorithms. We thus compare in Fig. 1(b) the running times of the modified FVI and the original value iteration algorithm under different values of K. Significant advantage of the modified FVI over its counterpart can be observed under each value of K. Moreover, the running time of the modified FVI grows in a polynomial manner as K increases, whereas that of the original value iteration grows exponentially. This implies that the modified FVI is still feasible when K is large, whereas the original value iteration algorithm may fail due to the high computational cost.

6.2. Simulation study of Algorithm 2

We now examine the performance—in particular, the sample complexity—of the online learning algorithm through simulation experiments. As discussed previously, one of our novelties in developing Algorithm 2 is that we incorporate an online method to evaluate the dependencies among components (i.e., adaptively detecting the value of $\{d^k\}_{k\in\mathcal{K}}$). Hence, we compare the performance of Algorithm 2 with that of the common practice that assumes known value of $\{d^k\}_{k\in\mathcal{K}}$ in advance.

In the simulation, we use the same model setting as before; that is, each component has 3 states and the cost function is identical. The same basis functions as specified in Section 6.1 are used here. We set other parameters as follows: $\gamma = 0.8$ and $A = 5$. To generate the transitions, we define two additional functions $\tilde{p}_1 : \{1, 2, 3\} \to [0, 1]$ and $\tilde{p}_2 : \{1, 2, 3\} \times \{1, 2, 3\} \to [0, 1]$ as

$$\tilde{p}_1(x) = \begin{cases} 0.95, & x = 3, \\ 0.90, & x = 2, \\ 0.85, & x = 1, \end{cases} \qquad \tilde{p}_2(x, y) = \begin{cases} 0.90, & (x, y) = (3, 2) \text{ or } (2, 1), \\ 0.80, & (x, y) = (3, 1), \\ 0, & \text{otherwise.} \end{cases}$$

Table 3
Average number of detached sets of components under different values of ρ.

Value of ρ	1	2	3	4	5	6	7	8	9	10	15	20	30	50
Average number of detached sets	4	4	4	4	4	3	3	3	3	3	3	3	2	1

Then, we set the degradation transition as

$$P_D(y^k \mid x(\mathcal{U}_k)) = \tilde{p}_2(x^k, y^k) \cdot \Pi_{k'=k-2}^{k+2} \tilde{p}_1(x^k).$$

Here we focus on the power of Algorithm 2 for moderate- or large-scale systems. We first study the influence of economic dependence on the optimal solution. For this purpose, we fix $K = 30$ and examine the number of detached sets of the optimal solution produced by Algorithm 2. Recall that $D(a)$ is the number of detached sets of components selected for maintenance under action a and a_t is the action taken by Algorithm 2 at epoch $t \geq 1$. In particular, we are interested in the average value $round(\sum_{t=1}^{T} D(a_t)/T)$ under different values of ρ, where $round(\cdot)$ returns the nearest integer to any input real number. We choose 14 different levels of ρ, that is, $\rho \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 30, 50\}$. The time horizon is fixed at 10^5 to ensure that the output policy by Algorithm 2 is stable. The results are presented in Table 3.

We can observe from Table 3 that the value of ρ has a significant influence on the number of detached sets of components. More specifically, when ρ becomes larger, the economic dependence among components becomes stronger and the benefit of maintaining adjacent components together becomes higher. This implies that the optimal maintenance policy prefers to maintain neighbouring components in each single epoch. This is consistent with our motivation of introducing the penalty coefficient ρ in the cost function.

Next, we conduct a comparison study to demonstrate the superiority of our proposed algorithm. To this end, we fix $\rho = 3$ as in Section 6.1 and vary the value of K from 20 to 30, namely, $K \in \{20, 21, \ldots, 30\}$. It should be noted that very few existing approaches can solve problems with such a large scale in the learning environment, especially when the parameters $\{d^k\}_{k\in\mathcal{K}}$ are unknown (Dann et al., 2017; Rosenberg & Mansour, 2021; Strehl, 2007). A commonly adopted routine is to set values of $\{d^k\}_{k\in\mathcal{K}}$ in advance and use this fixed value thereafter. Following this routine, we choose two maintenance strategies that both use fixed values of $\{d^k\}_{k\in\mathcal{K}}$ for performance comparison, which are suboptimal but make sense in practice. In particular, the first strategy is to ignore the dependence among different components and consider that each component evolves independently, or equivalently, fix $d^1 = \cdots = d^K = 0$ throughout the whole maintenance process. In the second strategy, only dependence between neighbouring components are considered, namely, $d^1 = \cdots = d^K = 1$ is fixed all the time. The true values of all d^k's are set to 2. The two strategies can be easily realized by excluding the learning process for each d^k in Algorithm 2 and fixing d^k at 0 and 1, respectively. In this way, we are comparing our algorithm (that considers the stochastic dependence and learns the dependence) with the strategy that does not consider stochastic dependence, in order to show the effectiveness of our method.

An important note to make is that the theoretical guarantee in Theorem 2 holds under the premise that the true optimal policy can be achieved each time. However, the true optimal policy, as we discussed in Section 4, is not computationally tractable in many real cases. As a result, we first illustrate the sample complexity of Algorithm 2 with the modified FVI approximation method. Because calculating the true optimal policy is no longer feasible, we can simply record the total number of samples Algorithm 2 uses to update the model parameters as its sample complexity. In Fig. 2(a), we illustrate the sample complexity of Algorithm 2 under different values of K. We can see that the sample complexity of Algorithm 2 increases in a polynomial pattern in K. This is different from the sample complexity of a learning algorithm for general MDPs, which grows exponentially in K. The result verifies the feasibility of Algorithm 2 in solving a large-scale maintenance problem. In Fig. 2(b), we compare the value functions of the three maintenance strategies. For a problem with over 20 components, the true value function of any policy is no longer tractable, thus we use the approximate value function instead of the true one in the experiment. In Fig. 2(b), we illustrate the approximate value functions of the three policies based on the strategies we select previously. The results in Fig. 2(b) show that the proposed algorithm outperforms the other two without learning processes.

7. Concluding remarks

In this work, we study a condition-based group maintenance problem for multi-component systems subject to multiple types of dependencies among components. The problem is modelled by an FMDP taking advantage of a specific location-based stochastic dependence among components. We first examine this problem from a traditional perspective in which model parameters are assumed to be fully known in advance. To reduce the computational burden, we develop a modified FVI algorithm to efficiently approximate the optimal maintenance policy and also provide an upper bound on the approximation error of this algorithm. Subsequently, we turn to an online learning environment in which model parameters are unknown *a priori*, and develop an online reinforcement learning algorithm to learn the model parameters and determine an optimal maintenance policy, simultaneously. The algorithm is capable of learning transition probabilities and the system structure (indicating the stochastic dependence among components) from previous observations. Moreover, it outperforms the other existing approaches in that it can generate computationally tractable and approximately optimal maintenance policies even under a large problem scale. A key point here is that we properly incorporate the dependence properties into the design of our learning algorithm. By doing so, our algorithm is able to effectively mitigate computational complexities and burdens associated with online maintenance problems, while still retaining a good performance.

We believe that our model and algorithms are not restricted to the specific setting concerned in this work. First, the location-based stochastic dependence assumption can be relaxed beyond neighbouring components, as long as the dependence structure can be gradually learned from previous observations. Second, the modified FVI with sampling technique in our framework can be replaced by any effective approximation method to solve FMDPs. Third, from a more conceptual perspective, the scheme of the proposed algorithm, which learns the dependence structure among components while evaluating optimal maintenance policies, can be modified and extended to more extensive maintenance problems for large-scale systems with some structural features among components.

However, this paper presents several limitations that deserve further research efforts. First, an implicit assumption adopted in this work is perfect inspection; that is, an inspection can reveal each component's actual state without errors. In reality, an inspection may be imperfect caused by measurement errors or sensor deterioration. Generalizing our modelling framework to involve imperfect inspections and developing appropriate algorithms to solve the associated maintenance problems is an open question. Second, we assume that at inspection, there are only two actions (replacement or do nothing) to be taken for each component. Considering imperfect maintenance actions of different depths is an interesting research topic. Finally, conducting a real-world case study to calibrate our FMDP model with real data (collected from, e.g., energy-generation, transportation, or manufacturing systems) and compare the performance of our algorithms with relevant benchmarks is highly valuable.

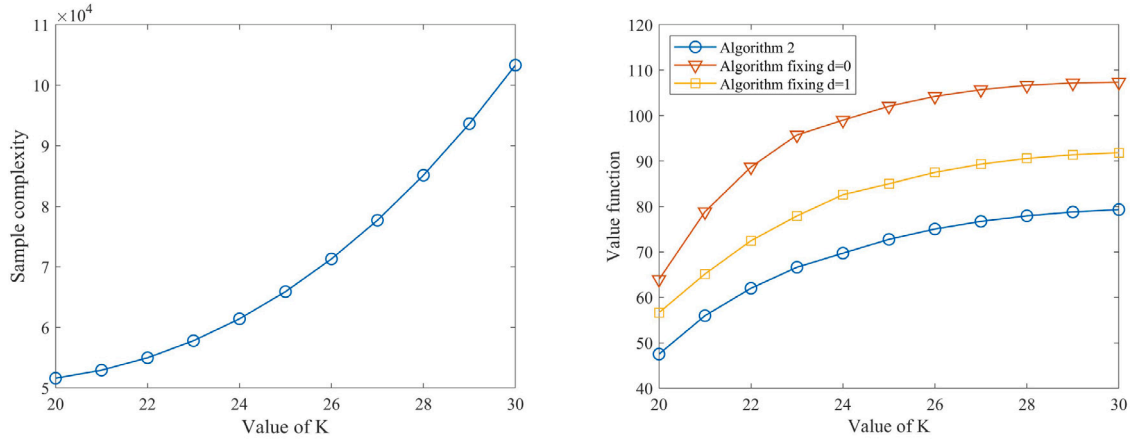

Fig. 2. The numerical performance of Algorithm 2.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (grant numbers 72201180, 72371182, 72002149) and in part by the Ministry of Education of China under the Humanities and Social Sciences Youth Foundation (grant number 22YJC630142). The authors are grateful to the editor and the anonymous reviewers for their constructive comments and suggestions that helped improve the paper significantly.

Appendix A. Proof of Theorem 1

The proof is based on reorganizing and refining the results in Szita and Lörincz (2008).

Proof of Theorem 1. The error $\|H\hat{w}^* - V^*\|_\infty$ is separated as

$$\left\|H\hat{w}^* - V^*\right\|_\infty \le \left\|Hw^* - V^*\right\|_\infty + \left\|Hw^* - H\hat{w}^*\right\|_\infty. \tag{A.1}$$

The rest of the proof is to bound $\|Hw^* - H\hat{w}^*\|_\infty$. Based on the discussions in the proof of Theorem 1 in Szita and Lörincz (2008), we have

$$\begin{aligned}\|Hw^* - H\hat{w}^*\|_\infty &\le \|H\|_\infty \|w^* - \hat{w}^*\|_\infty \\ &\le \frac{\|H\|_\infty}{1-\gamma}\left(\left\|Gc^{\pi_0} - \hat{G}\hat{c}^{\pi_0}\right\|_\infty + \gamma\left\|GB^{\pi_0} - \hat{G}\hat{P}^{\pi_0}\hat{H}\right\|_\infty \|w^*\|_\infty\right).\end{aligned} \tag{A.2}$$

If we define $c_k^{\pi_0}$ to be a K-dimensional vector with the kth component being $c^k(x^k, \pi_0(x)^k)$ and other components being 0, then $c^{\pi_0} = \sum_{k\in\mathcal{K}} c_k^{\pi_0}$. Meanwhile, because $G = \sum_{e=1}^{E} G_e$ and $B^{\pi_0} = \sum_{m=1}^{M} B^{\pi_0,m}$, according to Szita and Lörincz (2008, Lemma 5), we have for any $\epsilon_0 > 0$, with probability at least $1-\delta/2$,

$$\left\|Gc^{\pi_0} - \hat{G}\hat{c}^{\pi_0}\right\|_\infty \le \epsilon', \tag{A.3}$$

if $|\hat{S}| \ge 2\Psi_1 M^2/\epsilon_0^2 \cdot \log(4M^2/\delta)$, and with probability at least $1-\delta/2$,

$$\left\|GB^{\pi_0} - \hat{G}\hat{P}^{\pi_0}\hat{H}\right\|_\infty \le \epsilon_0, \tag{A.4}$$

if $|\hat{S}| \ge 2\Psi_2 M^2/\epsilon_0^2 \cdot \log(4M^2/\delta)$. If we let

$$\epsilon_0 = \frac{(1-\gamma)\epsilon}{\|H\|_\infty}\left(1+\gamma\left\|w^*\right\|_\infty\right),$$

then by combining (A.2), (A.3), and (A.4), we have, with probability at least $1-\delta$,

$$\left\|Hw^* - H\hat{w}^*\right\|_\infty \le \epsilon, \tag{A.5}$$

if

$$\begin{aligned}|\hat{S}| &\ge \frac{2M^2}{\epsilon_0^2}\log\left(\frac{4M^2}{\delta}\right)\max\{\Psi_1, \Psi_2\} \\ &\ge \frac{\left(1+\gamma\|w^*\|_\infty\right)^2\|H\|_\infty^2 M^2}{(1-\gamma)^2\epsilon^2}\log\left(\frac{4M^2}{\delta}\right)\cdot\max\{\Psi_1, \Psi_2\}.\end{aligned}$$

This completes the proof. □

Appendix B. Proof of Theorem 2

The proof follows the process we sketch in Section 5.2. In the following Lemma, we construct an event where the estimated transitions are close to the true transitions, and prove that the event holds with a high probability.

Lemma 1. *We define $\mathscr{E}$ to be an event where for all $t \ge 1$ during the execution of Algorithm Theorem 2 and all $k \in \mathcal{K}$, $x^k \in \{1,\dots,N\}$, $a^k \in \{0,1\}$, $x \in S$ and $d_k \le d \le \kappa$, relations*

$$\begin{aligned}&\left\|\hat{P}_{R,t}^k(\cdot \mid x^k, a^k) - P_R^k(\cdot \mid x^k, a^k)\right\|_1 \\ &\le \sqrt{\frac{2\ln\left\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_R/\delta\right\} + 4N\ln(2)}{\max\{n_t(x^k, a^k), 1\}}},\end{aligned} \tag{B.1}$$

and

$$\begin{aligned}&\left\|\hat{P}_{D,t}^k(\cdot \mid x(\mathcal{U}_k(d))) - P_D^k(\cdot \mid x(\mathcal{U}_k(d^k)))\right\|_1 \\ &\le \sqrt{\frac{2\ln\left\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_D/\delta\right\} + 2N^{2\kappa+1}\ln(2)}{\max\{n_t(x(\mathcal{U}_k(d))), 1\}}},\end{aligned} \tag{B.2}$$

hold simultaneously. Then $\mathscr{E}$ holds with probability at least $1-\delta$.

Proof. Since the total number of state–action pair (x^k, a^k) is $2N$ for each $k \in \mathcal{K}$, according to Weissman et al. (2003, Theorem 2.1), for any given $k \in \mathcal{K}$, $x^k \in \{1,\dots,N\}$ and $a^k \in \{0,1\}$, relation

$$\begin{aligned}&\left\|\hat{P}_{R,t}^k(\cdot \mid x^k, a^k) - P_R^k(\cdot \mid x^k, a^k)\right\|_1 \\ &\le \sqrt{\frac{2\ln\left\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_R/\delta\right\} + 4N\ln(2)}{\max\{n_t(x^k, a^k), 1\}}},\end{aligned}$$

holds with probability at least $1 - \delta/\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_R\}$. Meanwhile, since $d \ge d_k$ and $|x(\mathcal{U}_k(d))| = N^d$, for any given $k \in \mathcal{K}$, $x \in S$, and $d_k \le d \le \kappa$, relation

$$\left\|\hat{P}_{D,t}^k(\cdot \mid x(\mathcal{U}_k(d))) - P_D^k(\cdot \mid x(\mathcal{U}_k(d^k)))\right\|_1$$

$$\leq \sqrt{\frac{2\ln\left\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_D/\delta\right\}+N^{2d+1}\ln(2)}{\max\{n_t(x(\mathcal{U}_k(d))),\,1\}}}$$
$$\leq \sqrt{\frac{2\ln\left\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_D/\delta\right\}+2N^{2\kappa+1}\ln(2)}{\max\{n_t(x(\mathcal{U}_k(d))),\,1\}}},$$

holds with probability at least $1-\delta/\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_D\}$. Using the trick of the union probability bound, we combine all state–action pairs $(x^k,\,a^k)$'s and $X(\mathcal{U}_k(d))$'s, and all possible samples $1,\ldots,\tau_R$ for state–action pairs $n_t(x^k,\,a^k)$'s and all possible samples $1,\ldots,\tau_D$ for $x(\mathcal{U}_k(d))$'s to conclude that

$$\begin{aligned}Pr(\mathscr{E}) &\geq 1-\left(\frac{\delta\cdot\sum_{t=1}^{\tau_R}\sum_{k\in\mathcal{K}}\sum_{d=d_k}^{\kappa}N^{2d+1}}{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_R}+\frac{\delta\cdot\sum_{t=1}^{\tau_D}\sum_{k\in\mathcal{K}}2N}{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_D}\right)\\ &\geq 1-\left(\frac{\delta\cdot\sum_{t=1}^{\tau_R}\sum_{k\in\mathcal{K}}\sum_{d=1}^{\kappa}N^{2d+1}}{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_R}+\frac{\delta\cdot\tau_D K\cdot 2N}{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_D}\right)\\ &= 1-\left(\frac{\delta\cdot\tau_R\cdot K\cdot N^{2\kappa+1}(N-1)}{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_R}+\frac{\delta\cdot\tau_D K\cdot 2N}{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_D}\right)\\ &\geq 1-\delta.\end{aligned}$$

Thus, we conclude the proof. □

The corollary below follows directly from Lemma 1, namely, for all $d^k\leq d\leq\kappa$, Condition (10) does not hold.

Corollary 1. *Restricted to event $\mathscr{E}$, for all $t\geq 1$ during the execution of Algorithm 2, the relation $d_t^k\leq d^k$ holds.*

Proof. According to Lemma 1, $\forall d\geq d^k$, we have

$$\begin{aligned}&\left\|\hat{P}_{D,t}^k(\cdot\mid x(\mathcal{U}_k(d)))-\hat{P}_{D,t}^k(\cdot\mid x(\mathcal{U}_k(\kappa)))\right\|_1\\ &\leq\left\|\hat{P}_{D,t}^k(\cdot\mid x(\mathcal{U}_k(d)))-P_D^k(\cdot\mid x(\mathcal{U}_k(d^k)))\right\|_1\\ &\quad+\left\|\hat{P}_{D,t}^k(\cdot\mid x(\mathcal{U}_k(\kappa)))-P_D^k(\cdot\mid x(\mathcal{U}_k(d^k)))\right\|_1\\ &\leq\sqrt{\frac{2\ln\left\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_D/\delta\right\}+2N^{2\kappa+1}\ln(2)}{\max\{n_t(x(\mathcal{U}_k(d))),\,1\}}}\\ &\quad+\sqrt{\frac{2\ln\left\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_D/\delta\right\}+2N^{2\kappa+1}\ln(2)}{\max\{n_t(x(\mathcal{U}_k(\kappa))),\,1\}}}\\ &\leq 2\sqrt{\frac{2\ln\left\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_D/\delta\right\}+2N^{2\kappa+1}\ln(2)}{\max\{n_t(x(\mathcal{U}_k(\kappa))),\,1\}}},\end{aligned}$$

where the last inequality is because of $d\leq\kappa$, which implies $n_t(x(\mathcal{U}_k(\kappa)))\leq n_t(x(\mathcal{U}_k(d_t^k)))$. Thus, the update of d_t^k only happens when $d_t^k<d^k$, by which we conclude the proof. □

In the next lemma, we show that restricted to event $\mathscr{E}$, the L_1-divergence between the estimated transition $\hat{P}_t$ that Algorithm 2 uses each time and the true transition P can be bounded.

Lemma 2. *Restricted to event $\mathscr{E}$, the following relation*

$$\begin{aligned}&\left\|\hat{P}_t(\cdot\mid x,\,a)-P_t(\cdot\mid x,\,a)\right\|_1\\ &\leq K\cdot\max_{k\in\mathcal{K}}\cdot\max_{(x^k,\,a^k)}\sqrt{\frac{2\ln\left\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_R/\delta\right\}+4N\ln(2)}{\max\{n_t(x^k,\,a^k),\,1\}}}\\ &\quad+N^A\cdot K\cdot\max_{k\in\mathcal{K}}\max_{x(\mathcal{U}_k(\kappa))}\sqrt{\frac{2\ln\left\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_D/\delta\right\}+2N^{2\kappa+1}\ln(2)}{\max\{n_t(x(\mathcal{U}_k(\kappa))),\,1\}}},\end{aligned}$$

holds for all $t\geq 1$, $x\in\mathcal{S}$ and $a\in\mathcal{A}$.

Proof. For notational convenience, for all $k\in\mathcal{K}$, we define

$$\mu_{1,t}^k\triangleq\max_{(x^k,\,a^k)}\sqrt{\frac{2\ln\left\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_R/\delta\right\}+4N\ln(2)}{\max\{n_t(x^k,\,a^k),\,1\}}},$$

and

$$\mu_{2,t}^k\triangleq\max_{x(\mathcal{U}_k(\kappa))}\sqrt{\frac{2\ln\left\{2K\left[N^{2\kappa+1}(N-1)+2N\right]\tau_D/\delta\right\}+2N^{2\kappa+1}\ln(2)}{\max\{n_t(x(\mathcal{U}_k(\kappa))),\,1\}}}.$$

According to Strehl (2007, Corollary 1), $\forall x^{k'},\,a^{k'}$, we have

$$\begin{aligned}&\left\|\Pi_{k\in\mathcal{K}}\hat{P}_{R,t}^k\left(\cdot\mid x^k,\,a\right)-\Pi_{k\in\mathcal{K}}P_R^{k'}\left(\cdot\mid x^{k'},\,a^{k'}\right)\right\|_1\\ &\leq K\cdot\max_{k\in\mathcal{K}}\left\|\hat{P}_{R,t}^k\left(\cdot\mid x^k,\,a^k\right)-P_R^k\left(\cdot\mid x^k,\,a^k\right)\right\|_1\leq\kappa\cdot\mu_{1,t}^k,\end{aligned}\tag{B.3}$$

and for all $x(\mathcal{U}_k(d_t^k))$,

$$\begin{aligned}&\left\|\left(\Pi_{k\in\mathcal{K}}\hat{P}_{D,t}^k(\cdot\mid z(\mathcal{U}_k(d_t^k)))\right)-\left(\Pi_{k\in\mathcal{K}}P_D^k(\cdot\mid z(\mathcal{U}_k(d^k)))\right)\right\|_1\\ &\leq K\cdot\max_{z(\mathcal{U}_k(d_t^k))\in\mathcal{S}(\mathcal{U}_k(d_t^k))}\left\|\hat{P}_{D,t}^k(\cdot\mid z(\mathcal{U}_k(d_t^k)))-P_D^k(\cdot\mid z(\mathcal{U}_k(d^k)))\right\|_1\leq K\cdot\mu_{2,t}^k,\end{aligned}\tag{B.4}$$

where the last inequality is based on Corollary 1. Meanwhile, $\forall x^{k'},\,z^{k'},\,a^{k'}$, we have

$$\begin{aligned}&\left\|\left(\Pi_{k\in\mathcal{K}}\hat{P}_{R,t}^k\left(z^k\mid x^k,\,a^k\right)\right)\cdot\left(\Pi_{k\in\mathcal{K}}\hat{P}_{D,t}^k(\cdot\mid z(\mathcal{U}_k(d_t^k)))\right)\right.\\ &\quad\left.-\left(\Pi_{k\in\mathcal{K}}P_R^k\left(z^k\mid x^k,\,a^k\right)\right)\cdot\left(\Pi_{k\in\mathcal{K}}P_D^k(\cdot\mid z(\mathcal{U}_k(d^k)))\right)\right\|_1\\ &\leq\left\|\left(\Pi_{k\in\mathcal{K}}\hat{P}_{R,t}^k\left(z^k\mid x^k,\,a^k\right)\right)\cdot\left(\Pi_{k\in\mathcal{K}}\hat{P}_{D,t}^k(\cdot\mid z(\mathcal{U}_k(d_t^k)))\right)\right.\\ &\quad-\left(\Pi_{k\in\mathcal{K}}\hat{P}_{R,t}^k\left(z^k\mid x^k,\,a^k\right)\right)\cdot\left(\Pi_{k\in\mathcal{K}}P_D^k(\cdot\mid z(\mathcal{U}_k(d^k)))\right)\\ &\quad+\left(\Pi_{k\in\mathcal{K}}\hat{P}_{R,t}^k\left(z^k\mid x^k,\,a^k\right)\right)\cdot\left(\Pi_{k\in\mathcal{K}}P_D^k(\cdot\mid z(\mathcal{U}_k(d^k)))\right)\\ &\quad\left.-\left(\Pi_{k\in\mathcal{K}}P_R^k\left(z^k\mid x^k,\,a^k\right)\right)\cdot\left(\Pi_{k\in\mathcal{K}}P_D^k(\cdot\mid z(\mathcal{U}_k(d^k)))\right)\right\|_1\\ &\leq\left(\Pi_{k\in\mathcal{K}}P_R^k\left(z^k\mid x^k,\,a^k\right)\right)\\ &\quad\cdot\left\|\Pi_{k\in\mathcal{K}}\hat{P}_{D,t}^k(\cdot\mid z(\mathcal{U}_k(d_t^k)))-\Pi_{k\in\mathcal{K}}P_D^k(\cdot\mid z(\mathcal{U}_k(d^k)))\right\|_1\\ &\quad+\left|\left(\Pi_{k\in\mathcal{K}}\hat{P}_{R,t}^k\left(z^k\mid x^k,\,a^k\right)\right)-\left(\Pi_{k\in\mathcal{K}}P_R^k\left(z^k\mid x^k,\,a^k\right)\right)\right|\\ &\quad\cdot\left\|\Pi_{k\in\mathcal{K}}P_D^k(\cdot\mid z(\mathcal{U}_k(d^k)))\right\|_1\\ &\leq\left\|\Pi_{k\in\mathcal{K}}\hat{P}_{D,t}^k(\cdot\mid z(\mathcal{U}_k(d_t^k)))-\Pi_{k\in\mathcal{K}}P_D^k(\cdot\mid z(\mathcal{U}_k(d^k)))\right\|_1\\ &\quad+\left|\left(\Pi_{k\in\mathcal{K}}\hat{P}_{R,t}^k\left(z^k\mid x^k,\,a^k\right)\right)\right.\\ &\quad\left.-\left(\Pi_{k\in\mathcal{K}}P_R^k\left(z^k\mid x^k,\,a^k\right)\right)\right|.\end{aligned}\tag{B.5}$$

Recall that $\forall a\in\mathcal{A}$, we have $\sum_{k\in\mathcal{K}}a^k\leq A\ll K$, so under any $a\in\mathcal{A}$, there are at most a total number of N^A components of $z\in\mathcal{S}$ such that $\Pi_{k\in\mathcal{K}}\hat{P}_{R,t}^k\left(z^k\mid x^k,\,a^k\right)>0$. Therefore, $\forall x,\,y\in\mathcal{S}$, $a\in\mathcal{A}$,

$$\begin{aligned}&\left\|P(\cdot\mid x,\,a)-\hat{P}_t(\cdot\mid x,\,a)\right\|_1\\ &=\left\|\sum_{z\in\mathcal{S}}\left[\left(\Pi_{k\in\mathcal{K}}\hat{P}_{R,t}^k\left(z^k\mid x^k,\,a^k\right)\right)\cdot\left(\Pi_{k\in\mathcal{K}}\hat{P}_{D,t}^k(\cdot\mid z(\mathcal{U}_k(d_t^k)))\right)\right]\right.\\ &\quad\left.-\sum_{z\in\mathcal{S}}\left(\Pi_{k\in\mathcal{K}}P_R^k\left(z^k\mid x^k,\,a^k\right)\right)\cdot\left(\Pi_{k\in\mathcal{K}}P_D^k(\cdot\mid z(\mathcal{U}_k(d^k)))\right)\right\|_1\\ &\leq N^A\cdot\max_{z\in\mathcal{S}}\left\|\left(\Pi_{k\in\mathcal{K}}\hat{P}_{D,t}^k(\cdot\mid z(\mathcal{U}_k(d_t^k)))\right)-\left(\Pi_{k\in\mathcal{K}}P_D^k(\cdot\mid z(\mathcal{U}_k(d^k)))\right)\right\|_1\\ &\quad+\sum_{z\in\mathcal{S}}\left|\left(\Pi_{k'\in\mathcal{U}_k(d_t^k)}\hat{P}_{R,t}^{k'}\left(z^{k'}\mid x^{k'},\,a^{k'}\right)\right)-\left(\Pi_{k'\in\mathcal{U}_k(d_t^k)}P_R^{k'}\left(z^{k'}\mid x^{k'},\,a^{k'}\right)\right)\right|\\ &\leq N^A\cdot K\cdot\max_{k\in\mathcal{K}}\max_{z(\mathcal{U}_k(d_t^k))\in\mathcal{S}(\mathcal{U}_k(d_t^k))}\left\|\hat{P}_{D,t}^k(\cdot\mid z(\mathcal{U}_k(d_t^k)))-P_D^k(\cdot\mid z(\mathcal{U}_k(d^k)))\right\|_1\\ &\quad+\left\|\Pi_{k\in\mathcal{K}}\hat{P}_{R,t}^k\left(\cdot\mid x^k,\,a^k\right)-\Pi_{k\in\mathcal{K}}P_R^k\left(\cdot\mid x^k,\,a^k\right)\right\|_1\\ &\leq N^A\cdot K\cdot\max_{k\in\mathcal{K}}\mu_{2,t}^k+K\cdot\max_k\mu_{1,t}^k,\end{aligned}\tag{B.6}$$

where the last two inequalities follow from (B.3) and (B.4). Substituting $\mu_{1,t}^k$ and $\mu_{2,t}^k$ into (B.6), we finally conclude the proof. □

Based on Lemma 1, Corollary 1, and Lemma 2, we present the detailed proof of Theorem 2 below.

Proof of Theorem 2. The main target of the proof is to show that the three conditions in Strehl et al. (2006, Proposition 1) are satisfied by our algorithm, then use the results therein to construct an upper bound on the sample complexity of our algorithm and conclude the proof. We

first define a set of "known" state–action pairs at each time $t \geq 1$ as

$$\Phi_t \triangleq \left\{ (x, a) \in S \times \mathcal{A} \,\middle|\, \text{for all } k \in \mathcal{K},\ a \in \mathcal{A},\ y \in S, n_t(x^k, a^k) = \tau_R,\ n_t(y(\mathcal{U}_k(\kappa))) = \tau_D \right\},$$

and we also define a "known" action-value function as

$$Q^{K_t}(x, a) \triangleq \begin{cases} c(x, a) + \sum_{y \in S} P(y \mid x, a) \min_{a \in \mathcal{A}} Q^{K_t}(y, a), \ \forall (x, a) \in \Phi_t, \\ c(x, a) + \sum_{y \in S} \hat{P}_t(y \mid x, a) \min_{a \in \mathcal{A}} Q^{K_t}(y, a) - \beta_t(x, a), \ \forall (x, a) \notin \Phi_t. \end{cases}$$

According to Strehl et al. (2006, Proposition 1), to conclude the bound on the sample complexity in Theorem 2, we need to prove that for all $\varepsilon > 0$, $0 < \delta < 1$, $t \geq 1$ and all $(x, a) \in S \times \mathcal{A}$, the following conditions hold with probability at least $1 - \delta/2$: (1) $\min_{a \in \mathcal{A}} \hat{Q}_t(x, a) \leq \min_{a \in \mathcal{A}} Q^*(x, a) + \varepsilon/4$; (2) $|\min_{a \in \mathcal{A}} \hat{Q}_t(x, a) - \min_{a \in \mathcal{A}} Q^{K_t}(x, a)| \leq \varepsilon/4$; (3) The number of time steps when some $(x_t, a_t) \notin \Phi_t$ is observed can be bounded by $N \cdot K \cdot \tau_R + N^{2\kappa+1} \cdot K \cdot \tau_D$. In the rest of the proof, we verify these three conditions one by one. Note that we always restrict our discussions to event $\mathscr{E}$.

***Step* 1. Proving** $\min_{a \in \mathcal{A}} \hat{Q}_t(x, a) \leq \min_{a \in \mathcal{A}} Q^*(x, a) + \varepsilon/4$.

We consider the value iteration equation for solving Q_t below:

$$\hat{Q}_t^{i+1}(x, a) = c(x, a) + \sum_{y \in S} \hat{P}_t(y \mid x, a) \min_{a \in \mathcal{A}} \hat{Q}_t^i(y, a) - \beta_t(x, a), \forall x \in S,\ a \in \mathcal{A},$$

with $\hat{Q}_t^i$ being the ith iterated value function ($i \geq 0$) and $\hat{Q}_t^0(x, a) = (1 + \rho)A/(1 - \gamma)$. Next, we prove

$$\min_{a \in \mathcal{A}} \hat{Q}_t(x, a) \leq \min_{a \in \mathcal{A}} Q^*(x, a), \ \forall x \in S, \tag{B.7}$$

by induction on i. Because $c^k(x^k, a^k) \in [0, 1]$, and $\sum_{k \in \mathcal{K}} a^k \leq A$ for all a implies $D(a) \leq A$, we have $Q^*(x, a) \leq (1 + \rho)A/(1 - \gamma)$, therefore (B.7) holds for $i = 0$. Suppose that (B.7) holds for some i, then for all $x \in S$ and $a \in \mathcal{A}$,

$$\begin{aligned}
&\hat{Q}_t^{i+1}(x, a) - Q^*(x, a) \\
=&\gamma \sum_{y \in S} \hat{P}_t(y \mid x, a) \min_{a \in \mathcal{A}} Q_t^{i-1}(y, a) - \gamma \sum_{y \in S} P(y \mid x, a) \min_{a \in \mathcal{A}} Q^*(y, a) - \beta_t(x, a) \\
\leq&\gamma \sum_{y \in S} \hat{P}_t(y \mid x, a) \min_{a \in \mathcal{A}} Q^*(y, a) - \gamma \sum_{y \in S} P(y \mid x, a) \min_{a \in \mathcal{A}} Q^*(y, a) - \beta_t(x, a) \\
\leq&\frac{(1 + \rho)A \cdot \gamma}{1 - \gamma} \sum_{y \in S} \left(\hat{P}_t(y \mid x, a) - P(y \mid x, a) \right) - \beta_t(x, a) \\
\leq&\frac{(1 + \rho)A \cdot \gamma}{1 - \gamma} \left\| \hat{P}_t(\cdot \mid x, a) - P_t(\cdot \mid x, a) \right\|_1 - \beta_t(x, a) \leq 0,
\end{aligned}$$

where the last inequality follows from Lemma 2. Thus, (B.7) is proven, which implies $\min_{a \in \mathcal{A}} \hat{Q}_t(x, a) \leq \min_{a \in \mathcal{A}} Q^*(x, a) + \varepsilon/4$.

***Step* 2. Proving** $\left| \min_{a \in \mathcal{A}} \hat{Q}_t(x, a) - \min_{a \in \mathcal{A}} Q^{K_t}(x, a) \right| \leq \varepsilon/4$.

According to Strehl and Littman (2008, Lemma 1), we have

$$\begin{aligned}
&\left| \min_{a \in \mathcal{A}} \hat{Q}_t(x, a) - \min_{a \in \mathcal{A}} Q^{K_t}(x, a) \right| \leq \max_{a \in \mathcal{A}} \left| \hat{Q}_t(x, a) - Q^{K_t}(x, a) \right| \\
\leq& \frac{1}{(1 - \gamma)^2} \max_{(x, a) \in \Phi_t} \beta_t(x, a) + \frac{(1 + \rho)A \cdot \gamma}{(1 - \gamma)^2} \max_{(x, a) \in \Phi_t} \left\| \hat{P}_t(\cdot \mid x, a) - P_t(\cdot \mid x, a) \right\|_1 \\
=& \frac{(2 - \gamma)\gamma(1 + \rho)A}{(1 - \gamma)^3} \max_{(x, a) \in \Phi_t} \left\| \hat{P}_t(\cdot \mid x, a) - P_t(\cdot \mid x, a) \right\|_1.
\end{aligned} \tag{B.8}$$

Noting that $n_t(x^k, a^k) = \tau_R$ and $n_t(x(\mathcal{U}_k(\kappa))) = \tau_D$, based on Lemma 2, a set of sufficient conditions for the right-hand side of (B.8) to be no larger than $\varepsilon/4$ is

$$\begin{cases} \sqrt{\frac{2 \ln \left\{ 2K \left[N^{2\kappa+1}(N-1) + 2N \right] \tau_R / \delta \right\} + 4N \ln(2)}{\tau_R}} \leq \frac{(1-\gamma)^3 \varepsilon}{8(2-\gamma)\gamma(1+\rho)A \cdot K}, \\ \sqrt{\frac{2 \ln \left\{ 2K \left[N^{2\kappa+1}(N-1) + 2N \right] \tau_D / \delta \right\} + 2N^{2\kappa+1} \ln(2)}{\tau_D}} \leq \frac{(1-\gamma)^3 \varepsilon}{8(2-\gamma)\gamma(1+\rho)A \cdot N^A \cdot K}. \end{cases} \tag{B.9}$$

Using the inequality $\ln x \leq x/\sigma + \ln \sigma - 1$, $\forall x > 0$, $\sigma > 0$, we have

$$\begin{aligned}
&2 \ln \left\{ 2K \left[N^{2\kappa+1}(N-1) + 2N \right] \tau_R / \delta \right\} \\
&\leq 2 \ln \left\{ 2K \left[N^{2\kappa+1}(N-1) + 2N \right] / \delta \right\} \\
&+ \frac{(1-\gamma)^6 \varepsilon}{128(2-\gamma)^2 \gamma^2 (1+\rho)^2 A^2 K^2} \cdot \tau_R \\
&+ 4 \ln \left(\frac{16(2-\gamma)\gamma(1+\rho)A \cdot K}{(1-\gamma)^3 \varepsilon} \right) - 2 + 4N \ln(2),
\end{aligned}$$

and

$$\begin{aligned}
&2 \ln \left\{ 2K \left[N^{2\kappa+1}(N-1) + 2N \right] \tau_D / \delta \right\} \\
&\leq 2 \ln \left\{ 2K \left[N^{2\kappa+1}(N-1) + 2N \right] / \delta \right\} \\
&+ \frac{(1-\gamma)^6 \varepsilon}{128(2-\gamma)^2 \gamma^2 (1+\rho)^2 A^2 N^{2A} K^2} \cdot \tau_D \\
&+ 4 \ln \left(\frac{16(2-\gamma)\gamma(1+\rho)AN^A \cdot K}{(1-\gamma)^3 \varepsilon} \right) - 2 + 2N^{2\kappa+1} \ln(2).
\end{aligned}$$

So, we claim the following sufficient conditions for (B.9):

$$\begin{cases} 2 \ln \left\{ 2K \left[N^{2\kappa+1}(N-1) + 2N \right] / \delta \right\} + 4 \ln \left(\frac{16(2-\gamma)\gamma(1+\rho)A \cdot K}{(1-\gamma)^3 \varepsilon} \right) \\ \quad + 4N \leq \frac{(1-\gamma)^6 \varepsilon^2}{128(2-\gamma)^2 \gamma^2 (1+\rho)^2 A^2 K^2} \cdot \tau_R, \\ 2 \ln \left\{ 2K \left[N^{2\kappa+1}(N-1) + 2N \right] / \delta \right\} + 4 \ln \left(\frac{16(2-\gamma)\gamma(1+\rho)A \cdot N^A \cdot K}{(1-\gamma)^3 \varepsilon} \right) \\ \quad + 4N \leq \frac{(1-\gamma)^6 \varepsilon^2}{128(2-\gamma)^2 \gamma^2 (1+\rho)^2 A^2 N^{2A} K^2} \cdot \tau_D. \end{cases} \tag{B.10}$$

One can easily verify that (B.10) can be satisfied by the choices of τ_R and τ_D in Theorem 2.

***Step* 3. Bounding the number of epochs when some** $(x_t, a_t) \notin \Phi_t$ **is observed.**

Note that each time some $(x_t, a_t) \notin \Phi_t$ is observed, at least one of $n_t(x_t^k, a_t^k)$'s or $n_t(x_t(\mathcal{U}_k(\kappa)))$ is updated. Since Algorithm 2 only uses the first τ_D observations of each (x_t^k, a_t^k) and the first τ_R observations of $x_t(\mathcal{U}_k(\kappa))$, the number of times at least one of $n_t(x_t^k, a_t^k)$'s and $n_t(x_t(\mathcal{U}_k(\kappa)))$ is updated, is at most $\max\{N \cdot K \cdot \tau_r,\ N^{2\kappa+1} \cdot K \cdot \tau_D\}$. Therefore, the number of epochs when some $(x_t, a_t) \notin \Phi_t$ is observed can be bounded by $\max\{N \cdot K \cdot \tau_r,\ N^{2\kappa+1} \cdot K \cdot \tau_D\}$.

We finally claim that all the three conditions presented at the beginning of the proof are satisfied, and thus complete the whole proof. □

Appendix C. Proof of Proposition 1

The proof is directly based on mathematical deductions.

Proof of Proposition 1. Based on our choice of H, for each $k \in \mathcal{K}$, we can choose $x \in S$ such that $x^k = 1$ and $x^{k'} = 3$ for all $k' \neq k$. Then, we have $h_k(x) = 1$ and $h_{k'} = 0$ for all $k' \neq k$. Therefore, we conclude that H has linearly independent columns. Thus, H^+ can be calculated by $H^+ = \left(H^T H\right)^{-1} H^T$. Next, we have

$$h_k^T \cdot h_{k'} = \begin{cases} \frac{5|S|}{12} = \frac{5 \times 3^{K-1}}{4}, & k = k', \\ \frac{|S|}{4} = 3^K, & k \neq k', \end{cases}$$

for all $k, k' = 1, \ldots, K$. So,

$$H^T H = \begin{pmatrix} \frac{5 \times 3^{K-1}}{4}, & 3^K, & \cdots, & 3^K \\ 3^K, & \frac{5 \times 3^{K-1}}{4}, & \cdots, & 3^K \\ \vdots & \vdots & \ddots & \vdots \\ 3^K, & \cdots, & 3^K, & \frac{5 \times 3^{K-1}}{4} \end{pmatrix},$$

and

$$\left(H^T H\right)^{-1} = \begin{pmatrix} -\frac{2(3K-1)}{(3K+2)3^{K-1}}, & -\frac{2}{(3K+2)3^{K-2}}, & \cdots, & -\frac{2}{(3K+2)3^{K-2}} \\ -\frac{2}{(3K+2)3^{K-2}}, & -\frac{2(3K-1)}{(3K+2)3^{K-1}}, & \cdots, & -\frac{2}{(3K+2)3^{K-2}} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{2}{(3K+2)3^{K-2}}, & \cdots, & -\frac{2}{(3K+2)3^{K-2}}, & -\frac{2(3K-1)}{(3K+2)3^{K-1}} \end{pmatrix}.$$

Therefore,

$$H^+ = \frac{2}{(3K+2)3^{K-2}}\left((K-1/3)\cdot h_1 - \sum_{k\neq 1} h_k, \ldots, (K-1/3)\cdot h_K - \sum_{k\neq K} h_k\right)^T. \tag{C.1}$$

Next, we have

$$HH^+ = \frac{2}{(3K+2)3^{K-2}}\left((K-1/3)\sum_{k=1}^{K} h_k\cdot h_k^T - \sum_{k\neq k'} h_k\cdot h_{k'}^T\right). \tag{C.2}$$

For all $k', k' \in \mathcal{K}$, we denote by $\left(h_k\cdot h_{k'}^T\right)_{x_1,x_2}$, $x_1, x_2 \in \mathcal{S}$, the element of $h_k\cdot h_{k'}^T$. Since $\left(h_k\cdot h_{k'}^T\right)_{x_1,x_2}$ is the row of h_k corresponding to x multiplies the row of $h_{k'}$ corresponding to x_2, we have

$$\left(h_k\cdot h_{k'}^T\right)_{x_1,x_2} = \begin{cases} 1, & x_1^k = 1,\ x_2^{k'} = 1,\\ 1/2, & x_1^k = 1,\ x_2^{k'} = 2 \text{ or } x_1^k = 2,\ x_2^{k'} = 1,\\ 1/4, & x_1^k = 2,\ x_2^{k'} = 2,\\ 0, & \text{otherwise.}\end{cases} \tag{C.3}$$

For all $x \in \mathcal{S}$, if we define

$$\Omega_1(x) \triangleq \left\{k \in \mathcal{K} :\ x^k = 1\right\},\ \ \Omega_2(x) \triangleq \left\{k \in \mathcal{K} :\ x^k = 2\right\},$$

then according to (C.2) and (C.3), by simple deduction, we have

$$\begin{aligned} HH^+_{x_1,x_2} =& \frac{2}{(3K+2)3^{K-2}}\left[\left(K+\frac{2}{3}\right)\left(\left|\Omega_1(x_1)\cap\Omega_1(x_2)\right| \right.\right.\\ &+ \frac{1}{2}\left|\Omega_1(x_1)\cap\Omega_2(x_2)\right| + \frac{1}{2}\left|\Omega_2(x_1)\cap\Omega_1(x_2)\right|\\ &\left.+ \frac{1}{4}\left|\Omega_2(x_1)\cap\Omega_2(x_2)\right|\right) - \left|\Omega_1(x_1)\right|\cdot\left|\Omega_1(x_2)\right|\\ &- \frac{1}{2}\left(\left|\Omega_2(x_1)\right|\cdot\left|\Omega_1(x_2)\right| + \left|\Omega_1(x_1)\right|\cdot\left|\Omega_2(x_2)\right|\right)\\ &\left.- \frac{1}{4}\left|\Omega_2(x_1)\right|\cdot\left|\Omega_2(x_2)\right|\right]. \end{aligned}$$

If we define

$$\begin{aligned} \Delta_1 \triangleq& \left(\left|\Omega_1(x_1)\cap\Omega_1(x_2)\right| + \frac{1}{2}\left|\Omega_1(x_1)\cap\Omega_2(x_2)\right|\right.\\ &\left.+ \frac{1}{2}\left|\Omega_2(x_1)\cap\Omega_1(x_2)\right| + \frac{1}{4}\left|\Omega_2(x_1)\cap\Omega_2(x_2)\right|\right),\\ \Delta_2 \triangleq& \left|\Omega_1(x_1)\right|\cdot\left|\Omega_1(x_2)\right| + \frac{1}{2}\left(\left|\Omega_2(x_1)\right|\cdot\left|\Omega_1(x_2)\right| + \left|\Omega_1(x_1)\right|\cdot\left|\Omega_2(x_2)\right|\right)\\ &+ \frac{1}{4}\left|\Omega_2(x_1)\right|\cdot\left|\Omega_2(x_2)\right|. \end{aligned}$$

Then, by the relations $0 \le \left|\Omega_1(x)\right|, \left|\Omega_2(x)\right| \le K$, and $0 \le \left|\Omega_1(x)\right| + \left|\Omega_2(x)\right| \le K$, we have

$$\Delta_1 \le \frac{3}{2}\left|\Omega_1(x_1)\right| + \frac{3}{4}\left|\Omega_2(x_1)\right| \le \frac{3}{4}\left|\Omega_1(x_1)\right| + \frac{3}{4}\left(\left|\Omega_1(x_1)\right| + \left|\Omega_2(x_1)\right|\right) \le \frac{3K}{2},$$

and

$$\begin{aligned} \Delta_2 \le& \left|\Omega_1(x_1)\right|\cdot\left|\Omega_1(x_2)\right| + \frac{1}{2}\left[\left(K-\left|\Omega_1(x_1)\right|\right)\cdot\left|\Omega_1(x_2)\right| + \left(K-\left|\Omega_1(x_2)\right|\right)\cdot\left|\Omega_1(x_1)\right|\right]\\ &+ \frac{1}{4}\left(K-\left|\Omega_1(x_1)\right|\right)\left(K-\left|\Omega_1(x_2)\right|\right)\\ \le& \frac{K^2}{4} + \frac{K}{4}\left(\left|\Omega_1(x_1)\right| + \left|\Omega_2(x_2)\right|\right) + \frac{1}{4}\left|\Omega_1(x_1)\right|\cdot\left|\Omega_1(x_2)\right|\\ \le& \frac{3K^2}{4}. \end{aligned}$$

Therefore,

$$\begin{aligned} HH^+_{x_1,x_2} &\le \frac{2}{(3K+2)3^{K-2}}\max\left\{\left(K+\frac{2}{3}\right)\Delta_1,\ \Delta_2\right\}\\ &\le \frac{2}{(3K+2)3^{K-2}}\max\left\{\left(K+\frac{2}{3}\right)\frac{3K}{2},\ \frac{3K^2}{4}\right\}\\ &= \frac{K}{3^{K-2}}. \end{aligned}$$

Therefore, the row sum of each row of HH^+ can be bounded by $|S|\cdot K/3^{K-2} = 9K$, which means that the row sum of each row of HG is no more than 1. This implies that G satisfies the non-expansion property.

In addition, by (C.1), we have

$$\begin{aligned} H^+ =& \frac{2}{(3K+2)3^{K-2}}\left[\left(K+\frac{2}{3}\right)(h_1,\ldots,h_K)^T - \sum_{k=1}^{K}(h_k,\ldots,h_k)^T\right]\\ =& \frac{2}{(3K+2)3^{K-2}}\left[\sum_{k=1}^{K}\left(K+\frac{2}{3}\right)(0,\ldots,h_k,\ldots,0)^T - \sum_{k=1}^{K}(h_k,\ldots,h_k)^T\right]\\ =& \frac{2}{(3K+2)3^{K-2}}\sum_{k=1}^{K}\left(-h_k,\ldots,\left(K-\frac{1}{3}\right)\cdot h_k,\ldots,-h_k\right)^T. \end{aligned} \tag{C.4}$$

Thus, if we let

$$G_k \triangleq \frac{2}{9K(3K+2)3^{K-2}}\left(-h_k,\ldots,\left(K-\frac{1}{3}\right)\cdot h_k,\ldots,-h_k\right)^T,$$

then G_k is a $\{k\}$-scope matrix and $G = \sum_{k=1}^{K} G_k$ is a separation that satisfies Assumption 1 with $K_W = 1$. In addition, $\|G_k\|_\infty$ can be easily bounded by

$$\|G_k\|_\infty \le \frac{|S|}{2}\cdot\left(K-\frac{1}{3}\right)\cdot\frac{2}{9K(3K+2)3^{K-2}} = \frac{3K-1}{3K(3K+2)} < \frac{1}{3K+2}.$$

This completes the proof. □

References

Abbou, A., & Makis, V. (2019). Group maintenance: A restless bandits approach. *INFORMS Journal on Computing*, *31*(4), 719–731.

Ahmad, R., & Kamaruddin, S. (2012). An overview of time-based and condition-based maintenance in industrial application. *Computers & Industrial Engineering*, *63*(1), 135–149.

Barlow, E., Bedford, T., Revie, M., Tan, J., & Walls, L. (2021). A performance-centred approach to optimising maintenance of complex systems. *European Journal of Operational Research*, *292*(2), 579–595.

Brown, B., Liu, B., McIntyre, S., & Revie, M. (2022). Reliability analysis of load-sharing systems with spatial dependence and proximity effects. *Reliability Engineering & System Safety*, *221*, Article 108284.

Chen, N., Ye, Z.-S., Xiang, Y., & Zhang, L. (2015). Condition-based maintenance using the inverse Gaussian degradation model. *European Journal of Operational Research*, *243*(1), 190–199.

Dann, C., Lattimore, T., & Brunskill, E. (2017). Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning. *Advances in Neural Information Processing Systems*, *30*, 5713–5723.

de Jonge, B., & Scarf, P. A. (2020). A review on maintenance optimization. *European Journal of Operational Research*, *285*(3), 805–824.

Deep, A., Zhou, S., Veeramani, D., & Chen, Y. (2023). Partially observable Markov decision process-based optimal maintenance planning with time-dependent observations. *European Journal of Operational Research*, *311*(2), 533–544.

Degris, T., Sigaud, O., & Wuillemin, P.-H. (2006). Learning the structure of factored Markov decision processes in reinforcement learning problems. In *Proceedings of the 23rd international conference on machine learning* (pp. 257–264).

Deng, Z., Devic, S., & Juba, B. (2022). Polynomial time reinforcement learning in factored state MDPs with linear value functions. In *International conference on artificial intelligence and statistics* (pp. 11280–11304). PMLR.

Drent, C., Drent, M., Arts, J., & Kapodistria, S. (2023). Real-time integrated learning and decision making for cumulative shock degradation. *Manufacturing & Service Operations Management*, *25*(1), 1–369.

Elwany, A. H., Gebraeel, N. Z., & Maillart, L. M. (2011). Structured replacement policies for components with complex degradation processes and dedicated sensors. *Operations Research*, *59*(3), 684–695.

Gámiz, M. L., Limnios, N., & del Carmen Segovia-García, M. (2023). Hidden Markov models in reliability and maintenance. *European Journal of Operational Research*, *304*(3), 1242–1255.

Guestrin, C., Koller, D., Parr, R., & Venkataraman, S. (2003). Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, *19*, 399–468.

Hoffman, M., Song, E., Brundage, M. P., & Kumara, S. (2022). Online improvement of condition-based maintenance policy via Monte Carlo tree search. *IEEE Transactions on Automation Science and Engineering*, *19*(3), 2540–2551.

Kearns, M., & Koller, D. (1999). Efficient reinforcement learning in factored MDPs. In *Proceedings of the 16th international joint conference on artificial intelligence, vol. 2* (pp. 740–747).

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

Kıvanç, İ., Özgür-Ünlüakın, D., & Bilgiç, T. (2022). Maintenance policy analysis of the regenerative air heater system using factored POMDPs. *Reliability Engineering & System Safety*, *219*, Article 108195.

Liu, B., Pandey, M. D., Wang, X., & Zhao, X. (2021). A finite-horizon condition-based maintenance policy for a two-unit system with dependent degradation processes. *European Journal of Operational Research*, *295*(2), 705–717.

Mahadevan, S., & Maggioni, M. (2007). Proto-value functions: A Laplacian framework for learning representation and control in Markov decision processes. *Journal of Machine Learning Research*, *8*, 2169–2231.

Nguyen, H. S. H., Do, P., Vu, H.-C., & Iung, B. (2019). Dynamic maintenance grouping and routing for geographically dispersed production systems. *Reliability Engineering & System Safety*, *185*, 392–404.

Olde Keizer, M. C. A., Flapper, S. D. P., & Teunter, R. H. (2017). Condition-based maintenance policies for systems with multiple dependent components: A review. *European Journal of Operational Research*, *261*(2), 405–420.

Olde Keizer, M. C. A., Teunter, R. H., & Veldman, J. (2016). Clustering condition-based maintenance for systems with redundancy and economic dependencies. *European Journal of Operational Research*, *251*(2), 531–540.

Olde Keizer, M. C. A., Teunter, R. H., & Veldman, J. (2017). Joint condition-based maintenance and inventory optimization for systems with multiple components. *European Journal of Operational Research*, *257*(1), 209–222.

Osband, I., & Van Roy, B. (2014). Near-optimal reinforcement learning in factored MDPs. *Advances in Neural Information Processing Systems*, *27*, 604–612.

Puterman, M. L. (2014). *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons.

Rosenberg, A., & Mansour, Y. (2021). Oracle-efficient regret minimization in factored MDPs with unknown structure. *Advances in Neural Information Processing Systems*, *34*.

Sadeghi, J., & Askarinejad, H. (2010). Development of improved railway track degradation models. *Structure and Infrastructure Engineering*, *6*(6), 675–688.

Sallans, B., & Hinton, G. E. (2004). Reinforcement learning with factored states and actions. *Journal of Machine Learning Research*, *5*, 1063–1088.

Strehl, A. L. (2007). Model-based reinforcement learning in factored-state MDPs. In *2007 IEEE international symposium on approximate dynamic programming and reinforcement learning* (pp. 103–110).

Strehl, A. L., Diuk, C., & Littman, M. L. (2007). Efficient structure learning in factored-state MDPs. In *Proceedings of the twenty-second AAAI conference on artificial intelligence, vol. 7* (pp. 645–650).

Strehl, A. L., Li, L., & Littman, M. L. (2006). Incremental model-based learners with formal learning-time guarantees. In *Proceedings of the twenty-second conference on uncertainty in artificial intelligence* (pp. 485–493).

Strehl, A. L., & Littman, M. L. (2008). An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, *74*(8), 1309–1331.

Sun, Q., Ye, Z.-S., & Chen, N. (2017). Optimal inspection and replacement policies for multi-unit systems subject to degradation. *IEEE Transactions on Reliability*, *67*(1), 401–413.

Szita, I., & Lörincz, A. (2008). Factored value iteration converges. *Acta Cybernetica*, *18*(4), 615–635.

Szita, I., & Lörincz, A. (2009). Optimistic initialization and greediness lead to polynomial time learning in factored MDPs. In *Proceedings of the 26th annual international conference on machine learning* (pp. 1001–1008).

Talebi, M. S., Jonsson, A., & Maillard, O. (2021). Improved exploration in factored average-reward MDPs. In *International conference on artificial intelligence and statistics* (pp. 3988–3996). PMLR.

Tian, Z., & Liao, H. (2011). Condition based maintenance optimization for multi-component systems using proportional hazards model. *Reliability Engineering & System Safety*, *96*(5), 581–589.

Tian, Y., Qian, J., & Sra, S. (2020). Towards minimax optimal reinforcement learning in factored Markov decision processes. *Advances in Neural Information Processing Systems*, *33*, 19896–19907.

Wang, J., & Zhu, X. (2021). Joint optimization of condition-based maintenance and inventory control for a k-out-of-n: F system of multi-state degrading components. *European Journal of Operational Research*, *290*(2), 514–529.

Weissman, T., Ordentlich, E., Seroussi, G., Verdu, S., & Weinberger, M. J. (2003). *Inequalities for the L_1 deviation of the empirical distribution: Tech. Rep.*, Hewlett-Packard Labs.

Wildeman, R. E., Dekker, R., & Smit, A. C. J. M. (1997). A dynamic policy for grouping maintenance activities. *European Journal of Operational Research*, *99*(3), 530–551.

Xu, J., Liu, B., Mo, H., & Dong, D. (2021). Bayesian adversarial multi-node bandit for optimal smart grid protection against cyber attacks. *Automatica*, *128*, Article 109551.

Xu, Z., & Tewari, A. (2020). Reinforcement learning in factored MDPs: Oracle-efficient algorithms and tighter regret bounds for the non-episodic setting. *Advances in Neural Information Processing Systems*, *33*, 18226–18236.

Zhao, X., Liang, Z., Parlikad, A. K., & Xie, M. (2022). Performance-oriented risk evaluation and maintenance for multi-asset systems: A Bayesian perspective. *IISE Transactions*, *54*(3), 251–270.

Zheng, M., Lin, J., Xia, T., Liu, Y., & Pan, E. (2023). Joint condition-based maintenance and spare provisioning policy for a K-out-of-N system with failures during inspection intervals. *European Journal of Operational Research*, *308*(3), 1220–1232.

Zhou, Y., Guo, Y., Lin, T. R., & Ma, L. (2018). Maintenance optimisation of a series production system with intermediate buffers using a multi-agent FMDP. *Reliability Engineering & System Safety*, *180*, 39–48.

Zhou, Y., Lin, T. R., Sun, Y., & Ma, L. (2016). Maintenance optimisation of a parallel-series system with stochastic and economic dependence under limited maintenance capacity. *Reliability Engineering & System Safety*, *155*, 137–146.

Zhu, Z., & Xiang, Y. (2021). Condition-based maintenance for multi-component systems: Modeling, structural properties, and algorithms. *IISE Transactions*, *53*(1), 88–100.