Combinatorial Q-Learning for Condition-Based Infrastructure Maintenance

AKIRA TANIMOTO
NEC, Kanagawa 211-0011, Japan
e-mail: a.tanimoto@nec.com

ABSTRACT Infrastructure maintenance planning is a large-scale optimization problem of planning when and on which components to carry out maintenance so as to keep the whole infrastructure in good condition with minimal maintenance cost. Recent advances in condition monitoring techniques have enabled timely maintenance in response to the condition of each part regardless of age. In addition to the condition, the spatial structure is also important for cost-efficiency in infrastructure maintenance since traveling costs and/or setup costs can be saved by simultaneous maintenance of neighboring components, which is called economic dependency. This optimization problem naively has a high computational complexity of $O(2^nH)$, where $n$ is the number of components and $H$ is the planning horizon, and the predictive modeling of degradation is also a big issue. To solve this problem efficiently at scale, our proposed method utilizes two kinds of dynamic programming for temporal and spatial scalability and consequently enjoys $O(n)$ complexity at each time step. For temporal scalability, we utilize a direct modeling approach for the action value of maintenance instead of modeling degradation, namely, Q-learning. For spatial scalability, we exploit locality in economic dependency by means of a reasonable approximation of the Q-function. A typical baseline approach is to divide the whole infrastructure into fixed groups of neighboring components beforehand and determine if maintenance should be performed for all the components in each group at each time step. In contrast, our scalable method enables fully combinatorial optimization for each component at each time step. We demonstrate the advantage of our method in a simulated environment, and the resulting maintenance history intuitively illustrates the benefit of our dynamic grouping approach. We also show that our method has a kind of interpretability in the optimization at each time step.

INDEX TERMS Predictive maintenance, artificial intelligence, decision support systems, reinforcement learning, combinatorial optimization, dynamic programming.

I. INTRODUCTION

We consider an infrastructure maintenance planning problem for the road surfaces of highways; water, oil, and gas pipelines; and so on. At each discretized time step, the maintenance decision-maker considers which components, or the small patches of the road surface, should be maintained on the basis of the regularly observed condition of each component. If a number of patches are almost deteriorated and geospatially neighboring, simultaneous maintenance (as shown in Fig. 1) is economical. In highway maintenance, for example, the traveling cost of a maintenance team to the site and the setup costs associated with putting up lane restrictions are incurred once for the simultaneous maintenance of a larger section consisting of contiguous small patches. Similarly, in underground pipeline maintenance, the cost of drilling vertically is incurred only once for the simultaneous maintenance of a larger section, while the cost of drilling horizontally is incurred for each patch [1].

A huge maintenance cost is paid to keep the infrastructure in good condition since its condition is critical in terms of safety, conformity, and the prevention of economic loss caused by emergent corrective maintenance or availability loss. We focus on reducing the total cost, i.e., the sum of the maintenance and condition cost (risk) caused by deteriorated infrastructure.

Maintenance planning for minimizing the total cost has been extensively investigated in prior work [2]. For multi-component systems, i.e., those that have multiple maintenance targets, the so-called economic dependency of the targets and group-based maintenance is often discussed [3], [4]. Infrastructure maintenance can also be regarded as multi-component maintenance by considering small patches
as components. In road maintenance, for example, cost savings can be achieved by maintaining larger sections instead of small patches [4]. In [1], a maintenance optimization technique for an infrastructure network was proposed. They formalized a special type of economic dependency for an infrastructure network, namely, the network topology dependency (NTD), and proposed an optimization method under the benefit of maintenance for each component given. The NTD assumption reflects the locality of the economic dependency in infrastructure maintenance; i.e., the cost reduction is achieved only when the neighboring components are maintained simultaneously. To consider complex economic dependency such as NTD, combinatorial optimization is required, and the computational complexity is high. The proposed optimization method in [1] exploits the submodularity in NTD for computational efficiency. We also consider such locality in economic dependency. That is, we can assume that simultaneous maintenance is beneficial only when the maintenance target is spatially neighboring.

These maintenance optimization methods for multi-component systems are mostly built on the basis of time-based maintenance (TBM), in which each component has a predefined lifetime. Thus, the benefit of maintenance for each component can be calculated, but the uncertainty in the deterioration process is not considered. On the other hand, recent developments in health condition monitoring technologies have enabled the actual condition of each component of an infrastructure to be observed in a timely manner. Examples of such technologies include image processing [5] and sensor networks [6] for road surfaces, and fiber optic sensing for pipelines [7], [8]. These sensing technologies contribute to cost savings since only deteriorated components are maintained regardless of their age through a policy known as condition-based maintenance (CBM). Note that CBM includes a wide range of maintenance concepts, which are characterized as predictive maintenance aided by condition monitoring technologies.

These capabilities for health monitoring pose challenges to the subsequent stages of the information processing pipeline, i.e., analyzing the data and making a decision [9]. In particular, optimization for multi-component CBM is not straightforward due to the economic dependency and the uncertainty in condition degradation. The optimization for this setting is computationally more challenging than TBM when taking the uncertainty into account. Studies for CBM of large-scale multi-component systems such as those for infrastructures are limited. Existing work in this context [10]–[12] for systems such as those for heavy vehicles assumes simple economic dependency, i.e., constant maintenance costs or cost reductions, regardless of the number of components or which components are to be maintained. Since infrastructures are geospatially distributed systems with large numbers of components, the locality of economic dependency such as NTD should be considered.

A simple heuristic approach to avoid the whole combinatorial optimization with respect to locality is to divide the whole infrastructure into larger local sections in advance, which is called a fixed section-based maintenance policy (illustrated in Fig. 1(b)). However, this simplified approach lacks flexibility in optimization, which leads to limited performance.

To fully consider the local economic dependency and optimize large-scale maintenance actions efficiently, we utilize two dynamic programming techniques for temporal and spatial scalability. For temporal scalability, we implement the direct modeling approach of a cost-benefit evaluator, that is, Q-learning [13]. Q-learning aims to learn the total cost-benefit in the long run under the observed conditions as the state-action value function (known as the Q-function), \( Q(s, a) \). Once the Q-function is learned, the maintenance action can be easily evaluated without assessing the uncertain future degradation. For the spatial scalability of the combinatorial optimization of actions, we propose an approximated Q-function model and a linear-time optimization algorithm that exploits the locality in the economic dependency. The scalable action optimization is also necessary for learning the Q-function, since the Q-learning requires the optimal value \( \min_a Q(s, a) \) in each learning iteration. Although our Q-function is simple, the dynamic grouping of neighboring maintenance targets (shown in Fig. 1(c)) was significantly better than that of the fixed section-based approach.

In addition to the performance, our proposed method also provides an interpretation of the solution. Since maintenance decision-makers are often responsible for safety, the interpretability of an optimized solution matters. In our parameterized Q-function, the maintenance benefits for each component and cost are separated. Thus, the estimated benefit and condition for each component can be shown in the same figure, which enables the decision-makers to assess the cost-benefit tradeoff. A detailed discussion is provided in Section V.
In our experiments, we compare our dynamic grouping approach with the fixed section-based approach, since the independent maintenance policy shown in Fig. 1(a) is included in the fixed section-based maintenance policy where the section length (window width) is set to one. The optimized maintenance history provides an intuitive explanation of the advantage of determining groups dynamically.

For the geospatial structure of the maintenance targets, we focus on one-dimensional (1-D) cases such as highways and pipelines, which is the simplest way to demonstrate the advantage of our approach. In addition, most parts of a highway, for example, are 1-D. For the highway network, combining a fixed section-based policy for intersections and branching parts and dynamic grouping for the remaining parts would be effective.

## II. RELATED WORK

Condition-based infrastructure maintenance planning at scale has yet to be fully investigated. We introduce some related work and clarify the differences from our setting.

### A. MULTI-COMPONENT MAINTENANCE PLANNING

The most related area is multi-component maintenance planning. In [4], various types of component dependency including economic dependency are reviewed. In particular, NTD in [1] is related the most to our local economic dependency. However, TBM is assumed, in which the maintenance benefit is given or is easily calculated since the aging process is deterministic. That is, we have to estimate the benefit of maintenance (as explained in Section 10), which is assumed to be independent and explicitly given in [1]. To the best of our knowledge, condition-based multi-component maintenance at scale is a novel setting.

### B. CONDITION-BASED MAINTENANCE PLANNING

Both TBM and CBM aim at proactive maintenance to extend the lifetime of the entire system or to reduce accidents, downtime, and emergency maintenance costs due to unexpected failures [14]. While TBM policies tend to be too conservative with failures, resulting in high maintenance costs, CBM policies are more economical because health monitoring enables unnecessary maintenance to be controlled [14]. In the problem of maintenance planning based on CBM, it is basically assumed that the condition is measured regularly at a sufficient frequency or even continuously, except in a few studies that include the optimization of inspection policies in the problem setting [15], and methods for prognosis and decision-making based on the measured current condition and historical data are discussed [9]. Research on CBM-based maintenance planning can be broadly divided into two types of policies. The model-based approach, which optimizes after prognosis with respect to the condition, is reviewed in Section II-C, and the model-free (reinforcement learning) approach, which optimizes decision policies without explicit modeling of the deterioration process, is reviewed in Section II-D.

### C. MODEL-BASED PREDICTIVE CONTROL FOR MAINTENANCE PLANNING

While we adopt a model-free approach, Q-learning, model-based approaches have also been studied. In a model-based approach, the transition model $s_t = M(s_{t-1}, a)$ is first estimated, and then, on the basis of the estimated model, the action optimization and future prediction to a prediction horizon are iteratively performed. Since this approach is computationally complex, existing work [10]–[12] assumes simple economic dependencies. In railway infrastructure maintenance, applying the model-predictive control (MPC) is discussed [16]–[18], which is computationally expensive and does not scale to a massive number of components. In MPC, the future degradation up to the prediction horizon is predicted by using the estimated transition model, and then the maintenance action of not only the current but also the future maintenance plan up to the planning horizon is jointly optimized in each time step. In addition, uncertainty of the model estimation should be considered in this approach. In [16], [17], the chance-constrained optimization approach is proposed. They impose a constraint to be satisfied with high probability with respect to the model uncertainty. To evaluate the constraint, they have to make multiple predictions (called “scenarios”) with parameters sampled from the posterior probability of the transition model. Even though we only consider unconstrained optimization of the expected total cost, such uncertainty evaluation is generally necessary in model-based optimization as long as the cost (risk) evaluation has non-linearity with respect to the condition. On the other hand, our method has an advantage in that the evaluation of uncertainty is included in the training of the model, i.e., the objective function of the optimization is modeled directly, so that the evaluation of uncertainty, such as based on scenario sampling, is not necessary during training or testing. We further discuss this point in Section IV-A. In other areas related to maintenance, rebalancing in bike-sharing is considered a maintenance task in that the bike inventory in each 2-D distributed station is maintained to be sufficiently stocked [19]. In [19], combinatorial optimization is based on predicted values for such a problem; however, stations are clustered in advance. The advantage of our approach is that maintenance groups are determined dynamically, i.e., combinatorial optimization is performed at every time step.

### D. (DEEP) REINFORCEMENT LEARNING FOR MAINTENANCE PLANNING

The application of model-free reinforcement learning (RL) to maintenance has been explored recently. Examples include on-policy RL (e.g., SARSA algorithm [20], [21]) proposed for a petroleum industry production system [22], for opportunistic maintenance of a fleet of military trucks [23], for minimizing the forced outage in gas turbine maintenance [24], for minimizing the average inventory level and the average number of backorders by optimizing production/maintenance policy in manufacturing [25],
and for minimizing the maintenance cost and downtime in manufacturing [26].

In addition, especially since the successes of the deep Q-network (DQN) [27], applying off-policy (deep) reinforcement learning has been actively studied. The method corresponding to an off-policy configuration is superior in that it can utilize historical data of past maintenance by human experts to implement an optimized decision-making policy that is different from the policy in the past history immediately after offline training. DQN applications to maintenance include road pavement maintenance [28], bridge maintenance [29], and general multi-component condition-based maintenance [30]. In [30], stochastic and economic dependencies among multiple components are taken into account by DQN. DQN takes the same approach as ours in terms of Q-learning, and while its model is flexible enough to fully capture these dependencies, it is too complex to scale with respect to the number of components. The number of components assumed in [30] is around ten, while we assume up to thousands or more. DQN utilizes a multi-head neural network that outputs Q-values for each combination of actions (thus it has $2^n$ heads for the number of components $n$), while we have too many components ($n = 1000$ or more) to apply this approach in terms of statistical and computational complexity. Although DQN was successfully applied to bridge maintenance in [29], in which a large number of components ($k = 263$) are encoded as independent Q values (thus the network only has $k \times |A|$ heads, where $A$ denotes the candidate maintenance actions for each component), it would be suboptimal when the action is optimized independently for each component (as in [29]) and when the true Q value (e.g., the maintenance cost) has high dependency among actions for each component, as in our setting presented in Section III-C.

One possible approach for maintaining the combinatorial optimization of components is the actor-critic (AC) algorithm as in [31], in which an actor network that outputs an approximately optimal combinatorial action, as well as a critic network (single-head Q-network) with the action as its input, are trained. Although AC provides an approximated solution for the action optimization after training, training an actor is another big issue in terms of computation, and thus its scalability with respect to the number of components is limited. Also, these approaches face difficulties in terms of interpretability. Our approach combines a simple Q-function with dynamic programming-based optimization to resolve the scalability and interpretability issues.

An additional important possibility in the application of RL to maintenance is the integrated optimization of an inspection policy, [15] formulated maintenance decision-making as a partially observable Markov decision process (POMDP) and proposed evaluating the value of inspection, i.e., observing latent states (conditions). Although we assume regular inspection with sufficient frequency in this paper, which leads to the Markov decision process (MDP) without unobserved conditions, this would be an important direction for future research.

III. PROBLEM
A. PROBLEM DESCRIPTION
The problem can be described as optimizing which components (small patches of a 1-D structured infrastructure) to be maintained for minimizing the sum of the maintenance and condition cost (or risk) caused by deteriorated components in the long run based on the current observed condition of each component. The condition cost is a predefined non-decreasing function of the condition (degree of deterioration). The deterioration speed varies in every small patch, and thus it is inefficient to maintain by large sections, as in Fig. 1(b). This implies the need to divide the whole infrastructure into small patches in sufficient spatial resolution, which leads to a large number of components as a whole. Then, each component is a small patch, and the cost of sending a maintenance team and setting up (traveling/setup cost) is relatively higher than the cost of maintaining each component (working cost), which indicates the efficiency of the dynamic group-based maintenance in Fig. 1(c) compared to the independent maintenance in Fig. 1(a). Traveling cost is assumed to occur once for neighboring components maintained simultaneously at the same time step and the working cost is proportional to the number of components maintained. The consideration of traveling cost incurs the economic dependency, i.e., the total cost of maintenance cannot be written as the summation of independent maintenance costs for each component.

We address these problems, namely, minimizing the economically dependent maintenance cost and the condition cost in the long run. For the other points, we make the following simplifying assumptions.

- Complete maintenance by replacement: the condition is fully recovered after maintenance.
- Stochastic independence: each component deteriorates independently from other components.
- Regular (real-time) inspection: the latest condition is always observed for each component.

B. MARKOV DECISION PROCESS
Our problem, i.e., sequential maintenance decision-making aimed at long-term cost minimization under imperfect knowledge of condition degradation, can be modeled as a reinforcement learning problem. The Markov decision process (MDP) is a formalism of reinforcement learning to describe a discrete-time decision-making process with a stochastic environment. At each time step $t$, the decision-maker observes the state $s_t$ (the condition of each component) and decides on an action $a_t$ (which components to perform maintenance). At the same time, the decision-maker receives a reward (cost) $R(s_t, a_t)$, which consists of the condition cost and the maintenance cost. The state (condition) transits to the next state $s_{t+1}$ stochastically according to an (unknown) conditional probability $p(s_{t+1}|s_t, a_t)$ depending on the current state $s_t$ and the action $a_t$.

Formally, MDP consists of the following five parts.

- $S$ is a set of states of the environment.
• $A$ is a set of actions that can be taken as a result of decision-making.
• $p(s_{t+1} \mid s_t, a_t)$ is the state transition probability that means the action $a_t \in A$ in the state $s_t \in S$ will lead to the next state $s_{t+1} \in S$.
• $R(s_{t}, a_{t})$ is the immediate reward function (for maximization problem) or cost function (for minimization problem) of the action $a_t$ in the state $s_t$.
• $\beta \in [0, 1]$ is the discount parameter of future rewards.

The aim is to optimize a (deterministic) policy $\pi$ that maximizes or minimizes the discounted total reward (cost) in the long run, i.e.,

$$\arg \max_\pi \mathbb{E}_{(a_t \mid s_t), p(s_{t+1} \mid s_t, a_t)} \left[ \sum_{t=1}^{T} \beta^{t-i} R(s_t', \pi(a_t' \mid s_t')) \right].$$

(1)

The state is what determines the reward (cost) along with the action, i.e., the condition of the components. Here, an important assumption in MDP is the Markov property for the transition, i.e., the next state only depends on the current state (condition) and the action $p(s_{t+1} \mid s_t, a_t) = p(s_{t+1} \mid s_t, a_t)$. We assume that the state (condition) is representative of the entire past information for both the future states and rewards.

**C. PROBLEM FORMULATION**

We determine when and which maintenance targets (small patches of road or pipeline) should be maintained to minimize the cumulative cost including future maintenance cost and condition cost. We assume the current cost is given explicitly as the cost function $\text{Cost}(s, a)$, where $s = \{s_t\}, s_t \in \mathbb{R}$ is the state (condition) and $a = \{a_t\}, a_t \in \{0, 1\}$ is the action taken at each time step ($a_t = 1$ represents that the maintenance is performed for the $i$th patch).

The final goal is as follows. At each time step $t$, given the observed states (or the condition) $s_t \in \mathbb{R}^n$, where $n$ is the number of maintenance targets, we determine which targets are to be maintained to minimize the expected (discounted) total cost in the long run with regard to future actions assumed to be optimized. Thus, the optimal action for the time step $t$ is

$$a^*_t = \arg \min_{a \in \Gamma(t) \subseteq \{0, 1\}^n} \left\{ \text{Cost}(s_t, a) + \min_{a_{t+1}} \sum_{t'=t+1}^{t+H} \beta^{t'-t} \mathbb{E}_{s_t \mid s_t, a_t^*, \ldots, a_{t'-1}} [\text{Cost}(s_{t'}, a_{t'})] \right\},$$

(2)

where $\beta \in [0, 1]$ is the discount parameter, $H \in \mathbb{N} \cup \{\infty\}$ is the prediction horizon, and $\Gamma(t)$ is the feasible set of actions. $a_{t,i}$ is the maintenance action for the $i$th target at $t$. In the following sections, we assume $\Gamma(t) = \{0, 1\}^n$.

The cost function can be separated into maintenance (action) cost and condition (state) cost; namely,

$$\text{Cost}(s, a) = C_a(a) + C_s(s).$$

(3)

The local economic dependency in the action cost is formalized as

$$C_a(a) := a_1(c_w + c_t) + \sum_{i=2}^{n} a_i(c_w + (1 - a_{i-1})c_t),$$

(4)

where $c_t$ and $c_w$ are given constants that represent the travel costs occurring once for neighboring patches maintained simultaneously and the working costs for each patch, respectively. Fig. 2 illustrates the calculation of the action cost. The interaction term $-a_ia_{i-1}c_t$ represents the local economic dependency, which comes from the traveling cost savings, i.e., the traveling cost is incurred only once for the contiguous section maintained at the same time. Although only the dependency of one-neighboring components is modeled in (4), the length of the locality considered can easily be extended, i.e., the maintenance cost is assumed to be decomposed as $C_a(a) = \sum f_i(a_{i-k}, \ldots, a_i)$, where $(f_i)$ is a set of (possibly nonlinear) functions and $k$ denotes the width of locality considered. The benefit of simultaneous maintenance is considered to have such locality ($k \ll n$), which is the key assumption that we exploit to achieve the computationally efficient algorithm described in Section IV. By assuming this locality, we can exploit the dynamic programming by memorizing the optimal subaction costs not for each full combination of sub-actions $\{a_1, \ldots, a_{i-1}\} \subseteq \{0, 1\}^{i-1}$ but only for each combination of local actions $\{a_{i-k}, \ldots, a_{i-1}\} \subseteq \{0, 1\}^k$ to compute the optimal subaction costs for the $1, \ldots, i$th components, which results in the computational complexity of $O(n2^k)$. In this paper, we assume $k = 1$ for simplicity. Other global nonlinearity in the maintenance cost $C_a$ is ignored, such as the workload capacity in each time step $[4]$, which might matter when the resources are not sufficient.

For the state (condition) cost function, we assume the independence of each component. The dependent state cost setting has also been studied as a stochastic dependency in [12], although here we focus on economic dependency. For the state cost of each component, it is reasonable to assume a non-decreasing function. In our experiment, we set the following hinge cost:

$$C_s(s) := c_s \sum_{i} (s_i - \alpha)_+, \quad \text{(5)}$$

where $(x)_+ := \max\{x, 0\}, c_s$, and $\alpha$ are given constants.
A. Tanimoto: Combinatorial Q-Learning for Condition-Based Infrastructure Maintenance

Algorithm 1 Fitted-Q for Maintenance Optimization

Input: \( D = \{s_t, a_t, r_t, s_{t+1}\}, \beta, \text{Cost}(\cdot, \cdot) \).

Initialize \( \theta \).

while No convergence is met do

\( s \leftarrow \text{Get} \left( s_t, a_t, s_{t+1} \right) \) from \( D \) in random order.

Calculate the empirical target \( y \) with minimizing \( Q_\theta \) by Algorithm 2:

\[
y \leftarrow C_\theta(a_t) + \beta C_\beta(s_{t+1}) + \min_{a'} Q_\theta(s_{t+1}, a')
\]

Then, we have an optimal substructure:

\[
\tilde{Q}(s_t, a_t) = C_\theta(a_t) + \beta \mathbb{E}_{s_{t+1}|a_t, \theta} \left[ C_\beta(s_{t+1}) \right] \\
\quad + \beta \mathbb{E}_{s_{t+1}|a_t, \theta} \left[ \min_{a'} \tilde{Q}(s_{t+1}, a_{t+1}) \right].
\]

The first term represents the cost of maintenance at time \( t \) and the rest represent the benefit of maintenance, i.e., if we perform maintenance at time \( t \), the condition cost at time \( t+1 \) (the second term), the need for maintenance and the condition costs afterwards (the third term) will decrease.

Our adopted fitted-Q learning [32] minimizes the empirical inconsistency between both sides of Eq. (9) in terms of MSE (called the mean-squared Bellman error), which is the objective function \( L_\theta \) in Algorithm 1. This consequently enables (approximate) minimization of (2) through minimizing the learned Q-function \( Q_\theta \) as a proxy. Although there is no rigorous guarantee that the estimated Q function using the Bellman error will converge to the true Q function (except in special cases [13]), empirical evidence shows success in many fields [33].

Note that the Q-learning approach also handles the uncertainty in future condition degradation. In the model-based approach described in Section II-C, a state transition model \( M(s_t, a_t) \) is trained to predict the future state \( \hat{s}_{t+1} \), and the uncertainty in the future state \( \hat{s}_{t+1} \) has to be considered for unbiased estimation of the expectations in (9) due to the non-linearity in \( C_\beta \) and \( \min \tilde{Q} \). That is, even if the state prediction \( \hat{s}_{t+1} \) is an unbiased estimator of the expected future state \( \mathbb{E}[s_{t+1}] \), a simple plug-in estimation \( C_\beta(\hat{s}_{t+1}) \) is biased for the term \( \mathbb{E}[C_\beta(s_{t+1})] \) when \( C_\beta \) is nonlinear. This is why the model-based approach needs to take uncertainty into account explicitly. In contrast, our \( q \) function is trained to approximate the expectation terms directly, and thus we can simply minimize \( Q_\theta \) as an empirical estimate of (9).

B. Q-FUNCTION APPROXIMATION BY DECOMPOSED COST AND COMPONENT-WISE BENEFIT

We approximate the \( \tilde{Q} \) function (9) with a parametric model \( \tilde{Q}_\theta \). For \( \tilde{Q}_\theta \), we approximately assume the component-wise independence for these benefit terms in (9) (as assumed in [1]), which enables the fast optimization. The second term is component-wise independent under the component-wise transition (i.e., \( s_{t+1,i} = M(s_{t,i}, a_{t,i}) \)), that is, the second term can be decomposed into the sum of functions of \( a_{t,i} \) as \( \sum_i \mathbb{E}[C_i(M(s_{t,i}, a_{t,i}))] \). Therefore, the approximation of component-wise independence corresponds to ignoring the dependencies in the third term. This approximation is accurate when \( \beta \) is sufficiently small.

When \( \beta \) is not small enough, the future rewards are taken into account and approximated to be independent for each component. There would be some planning ability lost through this approximation,
e.g., clustering the degraded components left not maintained close together so that they can be maintained together in the future. On the other hand, it does not lose opportunistic planning ability in the sense of maintaining components that are likely to deteriorate in the near future.

After summing up terms that are not involved in optimizing $a$ as a constant, we have the following parametric Q-function that represents the cost-benefit tradeoff of maintenance:

$$Q_\theta(s_t, a_t) := C_\alpha(a_t) + \sum_{i} (1 - a_{t,i}) q(s_{t,i}; \theta) + \theta_0, \quad (10)$$

where the component-wise function $q$ represents the benefit for performing maintenance of each component, i.e., the cost of not performing maintenance. We will discuss the specific design of the component-wise benefit $q$ in Section IV-D.

This component-wise separation of Q function also contributes to the interpretability in optimization. In this formulation, the value $q(s_i)$ can be interpreted as the priority of performing maintenance on the $i$-th component. The detailed discussion is in Section VI-A.

Here, the constant $\theta_0$ in (10) represents a baseline cost. It does not directly affect the optimization; nonetheless, it contributes to the learning phase. Considering that the $Q_\theta$ function approximates the expected total cost in the long run (9), there may remain other terms besides the maintenance cost and benefit (the cost of not performing maintenance). That is the future cost that remains even when the maintenance is performed. Let us consider an extreme case where the condition cost $C_s$ is so high or the maintenance operation is so imperfect that it is optimal to perform maintenance for almost all components in every time step. Since the immediate maintenance cost is the same in both $Q_\theta$ and (9), the remaining terms in (9) would be the (expected) future condition and maintenance costs and those in (10) would be the benefit $q$ and the constant term $\theta_0$. Without the constant term ($\theta_0 = 0$), we need to express all the future costs by the benefit terms of the few components that are not maintained, which causes over-estimation of the maintenance benefit. In other words, $\theta_0$ is the constant that summarizes terms that are not involved in the current action optimization with respect to the cost-benefit tradeoff.

### C. Q-FUNCTION OPTIMIZATION BY DYNAMIC PROGRAMMING

Our approximated Q-function can be optimized with respect to the action in linear time by means of dynamic programming. This is because the locality of economic dependency enables the optimal action of a patch to depend only on the optimal action of the neighbors; i.e., it has an optimal substructure property, as shown below.

First, we define the partial value function $v_i(a)$ as

$$v_i(a_{t,i}) := a_{t,i}(c_w + c_t) + (1 - a_{t,i}) q(s_{t,i}) \quad (11)$$

and for $i = 2, \ldots, n$,

$$v_i(a_{t,i}) := \min_{a_{t,i-1}} \left\{ a_{t,i}(c_w + c_t) + \sum_{i' = 2}^{i} a_{t,i'}(c_w + (1 - a_{t,i'})c_t) + \sum_{i' = 1}^{i-1} (1 - a_{t,i'})q(s_{t,i'}) \right\}. \quad (12)$$

Note that, the minimization of $v_n(a_{t,n})$ is equivalent to that of the whole Q-function.

$$\min_{a_{t,n}} v_n(a_{t,n}) + \theta_0 = \min_{a_{t,1}, \ldots, a_{t,n}} Q_\theta(a_t, s_t) \quad (13)$$

The partial value $v_i(a_{t,i})$ depends on the combination of actions $(a_{t,i})_i$ only through the neighboring partial values $\{v_{i-1}(a_{t,i-1})\}_{a_{t,i-1}}$; namely,

$$v_i(a_{t,i} = 0) = \min_{a_{t,i-1}} \{v_{i-1}(a_{t,i-1} = 0) + q(s_{t,i})\},$$

$$v_i(a_{t,i} = 1) = \min_{a_{t,i-1}} \{v_{i-1}(a_{t,i-1} = 1) + c_t + c_w\} \quad (14)$$

This property means that we only have to calculate the partial values $\{v_i(a_{t,i} = 1), v_i(a_{t,i} = 0)\}_{i \in [n]}$ to obtain the optimal action $a^*_t$, which takes only linear time with respect to the number of components $n$. The detailed algorithm is described in Algorithm 2.

### D. MODELING $q_i$: THE MAINTENANCE PRIORITy of \textit{i}-TH TARGET

The component-wise value $q_i = q(s_{t,i}; \theta)$ in (10) represents the priority (or the benefit) of performing maintenance for the $i$-th component. In this section, we design the hypothesis space of the function $q$ specifically using domain knowledge of desirable properties as a benefit function.

First, the benefit of maintenance should be non-negative, i.e., $q(s_{t,i}) \geq 0$ should hold. Since the state cost $C_s(s_{t,i})$ is non-decreasing in $s_{t,i}$ and the condition does not improve (at least without maintenance), $q(s_{t,i})$ should also be non-decreasing. Considering these properties, we utilize the following parameterization for $q$, which is an extension of the softplus (smoothed ReLU) function [34].

$$q(s_{t,i}; \theta) := \frac{\theta_1}{\theta_1} \log (1 + \exp(\theta_1(s_{t,i} - \theta_2))). \quad (16)$$

The parameter $\theta_1$ controls the smoothness. Since we adopt non-linear parameterization for $q$, convergence is not guaranteed [35]. Thus, we try several initial parameters for $\theta$.

Here, we explain how to design the $q$ function given the condition cost $C_s$ in (5). Since the third term in (9) is greater than 0, the $q$ function should be greater than the expected condition cost in the next time step, i.e., $q(s_{t,i}) \geq \beta \mathbb{E} [C_s(s_{t+1,i})|s_{t,i}]$ should hold.
For the cost function, we used the state cost function in (4) with the parameters \(\alpha = 50\), \(c_s = 1\), and \(c_t = 10\).

For the cost function, we used the state cost function \(C_s\) in (5) with the parameters \(\alpha = 50\), \(c_s = 1\) and the action cost function \(C_a\) in (4) with the parameters \(c_w = 2\), \(c_t = 10\).

### V. EXPERIMENT

We investigated the effectiveness of this approach with experiments in a simulated environment.

#### A. ENVIRONMENT SETTINGS

Since degradation proceeds at an accelerated rate, a log-linear model is often assumed \([36]–[38]: s_i = e^{\beta t + a} = e^{\beta s_{i-1}}\).

This represents that the degraded condition itself causes further degradation. Also, we consider a stochastic degradation model with heteroscedastic noise, i.e., the degradation rate depends on its location (component) \(i\). The heteroscedasticity of road pavement, for example, is caused by the difference in traffic conditions, material properties, construction quality, and other geometric conditions \([39], [40]\). This difference in the degradation rate is the very reason CBM, in which the component to be maintained is determined in accordance with its degradation condition, is superior to TBM, which assumes a pre-determined lifetime. We also take into account the skewness of the degradation rate distribution \([41]\), i.e., several components show very fast degradation rates and need frequent maintenance. To reproduce these conditions, we use the following transition models \(\{M_i\}\) for each component (position) \(i\) as the environment:

\[
M_i(s_{t,i}, a_{t,i}) = \begin{cases} 1.1s_{t,i} + \epsilon \Delta_i & (a_{t,i} = 0) \\ 1.0 & (a_{t,i} = 1), \end{cases}
\]

where \(\Delta_i\) is the characteristic excess degradation rate for the \(i\)-th target, which is generated from the log-normal distribution \(\Delta_i^{(\text{base})} \sim \exp(N(0, 1.3))\) followed by the application of a Gaussian filter (std = 2) for smoothness. We fixed the average degradation rates \(\{\Delta_i\}\) once after sampling; thus the average frequency of maintenance needed for the \(i\)-th component is constant for the entire training and test periods. The resulting degradation rates \(\{\Delta_i\}\) are shown in Fig. 3 and the condition history for a specific component is shown in Fig. 4.

For the cost function, we used the state cost function \(C_s\) in (5) with the parameters \(\alpha = 50\), \(c_s = 1\) and the action cost function \(C_a\) in (4) with the parameters \(c_w = 2\), \(c_t = 10\).

#### B. TRAINING AND TESTING SETTINGS

We set the number of targets \(n = 1000\) and the training and testing periods \(T_{\text{train}} = \{0, \ldots, 1000\}, T_{\text{test}} = \{1001, \ldots, 2000\}\), respectively. To generate the training data, we adopted the fixed section-based policy in (20) with the parameters \(w = 10, \theta_t = 45\). The random values \(\Delta_i\) and \(\epsilon_{t,i}\) are the same for all policies tested, i.e., CBM with various parameters and the proposed policy. We set the discount parameter \(\beta = 0.9\), since a too-large discount parameter causes divergence. After training, we fixed the estimated Q-function during the test phase and ran a simulation. The test evaluation was done by the total cost in the entire test period \(T_{\text{test}}\).
A. Tanimoto: Combinatorial Q-Learning for Condition-Based Infrastructure Maintenance

Figure 4. Condition and maintenance history of specific target generated from (17) and the fixed section-based CBM policy (20). Condition degrades gradually, then returns to a good condition when maintenance is performed.

Table 1. Initial parameters tested.

| Parameter          | Values                      |
|--------------------|-----------------------------|
| $\theta_0$ (Baseline cost) | {0.1, 1}                    |
| $\theta_1$ (Smoothness)    | {0.05, 0.1, 0.2, 0.5, 1, 2} |
| $\theta_2$ (Threshold)     | {20, 25, 30, 35, 40, 45, 50}|
| $\theta_3$ (Slope)         | {0.1, 0.3, 1, 2}            |

We tried the initial parameters of the Cartesian product of the candidate shown in Table 1 and selected the best parameter that minimizes the training objective $\sum_{t \in T_{\text{train}}} L_{\theta}$. These initial parameters were selected considering the environment to ensure that we had a good parameter near one of the initial parameters. Let us consider a greedy policy as a baseline that considers only the action cost and condition cost in the next step, i.e., one that ignores the third term in (9). Further suppose that the expectation and the cost function $C_s$ in the second term can be approximately exchangeable, i.e.,

$$E_{s_{t+1} \mid a_t, s_t} [C_s(s_{t+1})] \simeq C_s(E[s_{t+1} \mid a_t, s_t])$$

$$= \sum_i (1 - a_i) C_s(1.1s_{t,i}).$$

Then, (9) can be expressed by our model with the parameters $\theta_0 = 0, \theta_1 \to \infty, \theta_2 = \alpha/1.1 \simeq 45,$ and $\theta_3 = 1.1\beta C_s \simeq 1.0$. The optimal Q function should be larger than this greedy Q function. The third term in (9) may include the baseline cost $\theta_0 > 0$. Due to the convexity of $C_s$, the second term will gradually increase near the threshold $s_{t,i} = 45$, i.e., the smoothness should be introduced as $\theta < \infty$. Also, considering the future cost (the third term in (9)), the threshold might be smaller: $\theta_2 < 45$. The condition may not exceed the threshold $\theta_2$ so many times because the maintenance is performed preventively, and there are seldom instances in a region such as $s > 50$, where the slope parameter $\theta_3$ alone is dominant, so the optimal slope $\theta_3$ depends on the interaction with other parameters. Although we chose the candidate initial parameters taking these properties into account, it may be possible to choose them using a black-box optimization such as the Bayesian optimization [42], [43].

C. BASELINE METHOD: FIXED SECTION-BASED CBM

As the baseline method, we examined the fixed section-based CBM approach (Fig. 1(b)). With the parameter of window width $w$, the targets are split into intervals in advance, and the action is taken for all targets in the section if the most degraded target in it is greater than the threshold $\theta_t$.

$$\pi_{\text{CBM}}(a_{t,i} = 1 \mid s_t) = \begin{cases} 1 & \left( \max_{j \in A_i} (s_j) \geq \theta_t \right) \\ 0 & \text{(otherwise)} \end{cases}$$

where $A_i = \{ j \mid \lceil j/w \rceil = \lfloor i/w \rfloor \}$ is the set of components in the same section as the $i$-th component. The resulting condition history with parameters ($w = 10, \theta_t = 50$) is shown in Fig. 5, which is also used for generating training data. Performance under this policy was sensitive to the parameters as shown in Fig. 6. These parameters have to be appropriately...
TABLE 2. Performance comparison. We ran simulations for each policy and evaluated the test performance in $T_{\text{test}}$. For the proposed method, we learn the Q-function with data in $T_{\text{train}}$ and fixed the Q-function in the test phase. For the baseline (section-based CBM), we studied several parameters (as in Fig. 6) and show the best performance in the test period.

|                | Section-based CBM with best parameters | Proposed method |
|----------------|----------------------------------------|-----------------|
| Total cost     | $4.76 \times 10^5$                     | $4.31 \times 10^5$ |

FIGURE 7. Condition history under dynamic grouping policy with learned parameters (a). The better performance of our approach (in Table 2) possibly comes from the exploitation of the local economic dependency and the variety of degradation rates. Rapidly degrading targets (extracted in (b)) are maintained frequently with a small number of groups by selecting groups alternately (indicated by red lines).

optimized using the training data, which is another issue. To simplify the discussion, we used the optimal parameters selected by the test performance and demonstrate that our method with learned parameters still outperforms the baseline with the optimal parameters.

VI. RESULTS AND DISCUSSION
The proposed method outperformed the baseline approach even when the best parameters ($w, \theta_t$) in the test period were chosen for the baseline, as shown in Table 2.

A possible explanation of the performance of dynamic grouping is illustrated in Fig. 7. Rapidly degrading targets ($i \in [40, 45]$) are frequently maintained with negligible expense by selecting sections that cover such targets alternately (indicated by red lines in Fig. 7(b)). This alternate selection of sections cannot be achieved in the fixed section-based approach, and we consider this is a key benefit of the flexibility of dynamic grouping.

A. INTERPRETABILITY IN OPTIMIZATION
The advantage of the separability approximation of the state cost function, i.e., $C_i(s_t) = \sum_i q_i(s_t, i)$, is not only the computational efficiency but also the interpretability in optimization. Black-box optimization is difficult to accept for maintenance decision-makers in the field since they are responsible for the safety or have motivation for factors other than minimizing the explicitly defined cost function with observed data. As shown in Fig. 8, $q_i(s_t, i)$ can be interpreted as the maintenance priority of the $i$-th target. We can plot it in the same graph as observed physical quantities, which maintainers are familiar with.

VII. CONCLUSION
In this paper, we presented a condition-based infrastructure maintenance planning problem as a sequential and combinatorial optimization problem. This problem setting requires large-scale combinatorial optimization for the combination of current and future actions of each component considering the uncertainty of the future conditions. To achieve the dynamic grouping of small components of large infrastructures, we introduced the local economic dependency assumption for maintenance cost. We proposed a number of approximations, namely, the Q-learning approach for temporal scalability and uncertainty, and a parameterized Q-function and dynamic programming for spatially scalable optimization of the Q-function, which exploits the locality in economic dependency.

We investigated the performance in a simulated environment. The resulting condition history showed the advantage of dynamic grouping; that is, rapidly degrading targets could be maintained frequently by selecting alternate sections with a small extra expense only in working cost. The proposed method is not only has a superior performance but is also interpretable, which we feel is important for maintenance decision-makers to accept the recommended action. This is achieved by separating the objective function of action optimization (Q-function) into the action cost and sum of
maintenance priority for each component, which can be indicated in the same figure as the observed condition of each component. Comparing the cost and maintenance priority enables the maintenance planner to make a reasonable decision.

There are a number of remaining issues or limitations with our method, as well as possible extensions. In real applications, historical data is sometimes limited. Since the transition of each target at a single time step is summarized, deciding the condition observations are noise-free, but in the maintenance field, they often have severe noise or outliers. Therefore, estimating the true condition \( s_t \), or calculating \( q_t \) from many observations (e.g., a CNN-like model \( q_t(s_{t−1:t:i−k:i+k}) \)), is an important possible extension. In addition, we focused on 1-D infrastructures. Other possible applications of the dynamic grouping approach include whole network settings such as NTD and two-dimensionally distributed targets such as machine maintenance and inventory management of vending machines, ATMs, and so on.

**ACKNOWLEDGMENT**

The author would like to thank Y/Chiba for the discussion on conceptualization. They would also like to thank Shinji Ito for the discussion on optimization methods.

**REFERENCES**

[1] I. S. Papadakis and P. R. Kleinendorf, “Optimizing infrastructure network maintenance when benefits are interdependent,” *OR Spectr.*, vol. 27, no. 1, pp. 63–84, Jan. 2005.

[2] A. K. Jardine and A. H. Tsang, *Maintenance, Replacement, and Reliability: Theory and Applications*. Boca Raton, FL, USA: CRC Press, 2005.

[3] R. Dekker, R. E. Wildeman, and F. A. van der Duyn Schouten, “A review of multi-component maintenance models with economic dependence,” *Math. Methods Oper. Res.*, vol. 45, no. 3, pp. 411–435, Oct. 1997.

[4] R. P. Nicolai and R. Dekker, “Optimal maintenance of multi-component systems: A review,” in *Complex System Maintenance Handbook*. London, U.K.: Springer, 2008, pp. 263–286.

[5] S. Chambron and J.-M. Moliard, “Automatic road pavement assessment with image processing: Review and comparison,” *Int. J. Geophys.*, vol. 2011, pp. 1–20, Jun. 2011.

[6] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fennes, S. Glaser, and M. Turon, “Health monitoring of civil infrastructures using wireless sensor networks,” in *Proc. 6th Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, 2007, pp. 254–263.

[7] H.-N. Li, D.-S. Li, and G.-B. Song, “Recent applications of fiber optic sensors to health monitoring in civil engineering,” *Eng. Struct.*, vol. 26, no. 11, pp. 1647–1657, Sep. 2004.

[8] D. Inaudi and B. Glisic, “Long-range pipeline monitoring by distributed fiber optic sensing,” *J. Pressure Vessel Technol.*, vol. 132, no. 1, Feb. 2010, Art. no. 011701.

[9] A. Bousdekis, B. Magoutas, D. Apostolou, and G. Mentzias, “Review, analysis and synthesis of prognostic-based decision support methods for condition based maintenance,” *J. Intell. Manuf.*, vol. 29, no. 6, pp. 1303–1316, Aug. 2018.

[10] Z. Tian and H. Liao, “Condition based maintenance optimization for multi-component systems using proportional hazards model,” *Rel. Eng. Syst. Saf.*, vol. 96, no. 5, pp. 581–589, May 2011.

[11] K.-A. Nguyen, P. Do, and A. Grall, “Multi-level predictive maintenance for multi-component systems,” *Rel. Eng. Syst. Saf.*, vol. 144, pp. 83–94, Dec. 2015.

[12] A. Van Horenbeek and L. Pintelon, “A dynamic predictive maintenance policy for complex multi-component systems,” *Rel. Eng. Syst. Saf.*, vol. 120, pp. 39–50, Dec. 2013.

[13] J. C. H. Watkins and P. Dayan, “Q-learning,” *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.

[14] Y. Peng, M. Dong, and M. J. Zuo, “Current status of machine prognostics in condition-based maintenance: A review,” *Int. J. Adv. Manuf. Technol.*, vol. 50, nos. 1–4, pp. 297–313, Sep. 2010.

[15] C. P. Andriotis and K. G. Papakonstantinou, “Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints,” 2020, arXiv:2007.01380. [Online]. Available: http://arxiv.org/abs/2007.01380

[16] Z. Su, A. Jamshidi, A. Núñez, S. Baldi, and B. De Schutter, “Integrated condition-based track maintenance planning and crew scheduling of railway networks,” *Transp. Res. C, Emerg. Technol.*, vol. 105, pp. 359–384, Aug. 2019.

[17] Z. Su, A. Jamshidi, A. Núñez, S. Baldi, and B. De Schutter, “Multi-level condition-based maintenance planning for railway infrastructures—A scenario-based chance-constrained approach,” *Transp. Res. C, Emerg. Technol.*, vol. 84, pp. 92–123, Nov. 2017.

[18] K. Verbert, B. De Schutter, and R. Babuška, “Timely condition-based maintenance planning for multi-component systems,” *Rel. Eng. Syst. Saf.*, vol. 159, pp. 310–321, Mar. 2017.

[19] J. Liu, L. Sun, W. Chen, and H. Xiong, “Rebalancing bike sharing systems: A multi-source data smart optimization,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1005–1014.

[20] S. P. Singh and R. S. Sutton, “Reinforcement learning with replacing eligibility traces,” *Mach. Learn.*, vol. 22, nos. 1–3, pp. 123–158, 1996.

[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[22] N. Aissani, B. Behjadi, and D. Trentesaux, “Dynamic scheduling of maintenance tasks in the petroleum industry: A reinforcement approach,” *Eng. Appl. Artif. Intell.*, vol. 22, no. 7, pp. 1089–1103, Oct. 2009.

[23] S. R. A. Barde, S. Yacout, and H. Shin, “Optimal preventive maintenance policy based on reinforcement learning of a fleet of military trucks,” *J. Intell. Manuf.*, vol. 30, no. 1, pp. 147–161, Jan. 2019.

[24] M. Compare, L. Bellani, E. Cobelli, and E. Zio, “Reinforcement learning-based flow management of gas turbine parts under stochastic failures,” *Int. J. Adv. Manuf. Technol.*, vol. 99, nos. 9–12, pp. 2981–2992, Dec. 2018.

[25] A. S. Xanthopoulos, A. Kiatipis, D. E. Koulouriotis, and S. Stieger, “Reinforcement learning-based and parametric production-maintenance control policies for a deteriorating manufacturing system,” *IEEE Access*, vol. 6, pp. 576–588, 2018.

[26] A. Kuhnle, J. Jakubík, and G. Lanza, “Reinforcement learning for opportunistic maintenance optimization,” *Prod. Eng.*, vol. 13, no. 1, pp. 33–41, Feb. 2019.

[27] V. Muñh, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” 2013, arXiv:1312.5602. [Online]. Available: http://arxiv.org/abs/1312.5602

[28] L. Yao, Q. Dong, J. Jiang, and F. Ni, “Deep reinforcement learning for long-term pavement maintenance planning,” *Comput.-Aided Civil Infrastr. Eng.*, vol. 35, no. 11, pp. 1230–1245, Nov. 2020.

[29] S. Wei, Y. Bao, and H. Li, “Optimal policy for strategy maintenance: A deep reinforcement learning framework,” *Struct. Saf.*, vol. 83, Mar. 2020, Art. no. 101906.

[30] N. Zhang and W. Si, “Deep reinforcement learning for condition-based maintenance planning of multi-component systems under dependent competing risks,” *Rel. Eng. Syst. Saf.*, vol. 203, Nov. 2020, Art. no. 107094.

[31] Y. Liu, Y. Chen, and T. Jiang, “Dynamic selective maintenance optimization for multi-state systems over a finite horizon: A deep reinforcement learning approach,” *Eur. J. Oper. Res.*, vol. 283, no. 1, pp. 166–181, May 2020.

[32] M. Riedmiller, “Neural fitted Q iteration—first experiences with a data efficient neural reinforcement learning method,” in *Proc. Eur. Conf. Mach. Learn.*, Berlin, Germany: Springer, 2005, pp. 317–328.

[33] A. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, “Deep reinforcement learning: A brief survey,” *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.
[34] H. Zheng, Z. Yang, W. Liu, J. Liang, and Y. Li, “Improving deep neural networks using softplus units,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2015, pp. 1–4.

[35] L. Getoor and B. Taskar, Introduction to Statistical Relational Learning, Cambridge, MA, USA: MIT Press, 2007.

[36] N. K. Srivastava and S. Mondal, “Development of predictive maintenance model for N-component repairable system using NHPP models and system availability concept,” Global Bus. Rev., vol. 17, no. 1, pp. 105–115, Feb. 2016.

[37] R. R. Zhou, N. Serban, and N. Gebraeel, “Degradation modeling applied to residual lifetime prediction using functional data analysis,” Ann. Appl. Statist., vol. 5, no. 2B, pp. 1586–1610, Jun. 2011.

[38] S. M. Famurewa, T. Xin, M. Rantatalo, and U. Kumar, “Optimisation of maintenance track possession time: A tamping case study,” Proc. Inst. Mech. Eng. F, J. Rail Rapid Transit, vol. 229, no. 1, pp. 12–22, Jan. 2015.

[39] S. S. Adlinge and A. Gupta, “Pavement deterioration and its causes,” Int. J. Innov. Res. Develop., vol. 2, no. 4, pp. 437–450, 2013.

[40] H. P. Hong and S. S. Wang, “Stochastic modeling of pavement performance,” Int. J. Pavement Eng., vol. 4, no. 4, pp. 235–243, Dec. 2003.

[41] C.-Y. Peng and S.-T. Tseng, “Statistical lifetime inference with skew-wiener linear degradation models,” IEEE Trans. Rel., vol. 62, no. 2, pp. 338–350, Jun. 2013.

[42] S. Snoek, H. Larochelle, and R. Adams, “Practical Bayesian optimization of machine learning algorithms,” in Proc. Adv. Neural Inf. Process. Syst., vol. 25, 2012, pp. 2951–2959.

[43] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, “Taking the human out of the loop: A review of Bayesian optimization,” Proc. IEEE, vol. 104, no. 1, pp. 148–175, Jan. 2016.

[44] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, “Continuous deep Q-learning with model-based acceleration,” in Proc. Int. Conf. Mach. Learn., 2016, pp. 2829–2838.

AKIRA TANIMOTO received the B.S. and M.S. degrees in aerospace engineering from the University of Tokyo, Tokyo, Japan, in 2012 and 2014, respectively. He is currently pursuing the Ph.D. degree in informatics with Kyoto University, Kyoto, Japan.

Since 2014, he has been a Researcher with the NEC Data Science Research Laboratories. His research interest include machine learning for decision-making in business domains, such as reinforcement learning, sample-efficient machine learning, and causal inference.

***