CIAMTIS

U.S. DOT Region 3 University Transportation Center

AI-enabled Fiscally Constrained Lifecycle Asset Management for Infrastructure Systems

November 30, 2021

Prepared by:

K.G. Papakonstantinou, I. Guler, V. Gayah, M. Saifullah, C.P. Andriotis, and M. Lu

The Pennsylvania State University

r3utc.psu.edu

PennState College of Engineering

LARSON TRANSPORTATION INSTITUTE

DISCLAIMER

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
CIAM-UTC-REG21		

4. Title and Subtitle	5. Report Date
AI-enabled Fiscally Constrained Lifecycle Asset Management for Infrastructure Systems	November 30, 2021
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
K.G. Papakonstantinou https://orcid.org/0000-0002-5254-1066, I. Guler https://orcid.org/0000-0001-6255-3135, V. Gayah https://orcid.org/0000-0002-0648-3360, M. Saifullah, C.P. Andriotis, and M. Lu	LTI 2022-05

9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
Department of Civil and Environmental Engineering The Pennsylvania State University 215 Sackett Building University Park, PA 16802	**11. Contract or Grant No.** 69A3551847103

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
U.S. Department of Transportation Research and Innovative Technology Administration 3rd Fl, East Bldg E33-461 1200 New Jersey Ave, SE Washington, DC 20590	Final Report 03/01/2019 – 08/31/2021
	14. Sponsoring Agency Code

15. Supplementary Notes

Work funded through The Pennsylvania State University through the University Transportation Center Grant Agreement, Grant No. 69A3551847103.

16. Abstract

Accurate evaluation and prediction of infrastructure components performance and determination of optimal maintenance and inspection actions for any infrastructure element throughout its lifetime are essential parts of an effective infrastructure asset management framework. The goals of this project were to develop artificial Intelligence (AI)-enabled solutions, providing infrastructure condition assessment and prediction models, as well as algorithms able to directly suggest optimal maintenance and inspection decisions for multi-component infrastructure systems over long planning horizons. This report develops a prediction and decision-making framework for inspecting and maintaining deteriorating systems with incomplete information and constraints. In doing so, a Partially Observable Markov Decision Processes (POMDPs) approach is used, with an original deep reinforcement learning formulation. Thus, a Deep Decentralized Multiagent Actor-Critic (DDMAC) architecture is devised and manages to successfully tackle numerous challenges imposed by this stochastic control problem. Various constraints are also effectively incorporated in this framework. Further, a deterioration model for bridge decks using Random Survival Forest is developed. The results suggest that AI methods can achieve high accuracy in predicting the deterioration pattern of bridge decks, which is an important input into the stochastic optimal control framework. However, while AI methods may be preferred for prediction, because it is difficult to interpret the impacts of different variables on deterioration, traditional stochastic methods can be more powerful for construction or design purposes.

17. Key Words	18. Distribution Statement
Infrastructure asset management, bridge deck deterioration, condition assessment, prediction model, artificial intelligence	No restrictions. This document is available from the National Technical Information Service, Springfield, VA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	60	

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Optimal management of structures and infrastructure is an ongoing and critical problem aimed at appropriate inspection and maintenance policies, dealing with different stochastic degradation impacts and recommending optimum actions that serve multi-purpose lifecycle goals. The optimal allocation of economic and other resources in such systems is critical in establishing successful policies and is a compelling engineering goal. This problem becomes pressing based on the chronic lack of resources, which strains the nation's infrastructure, also given its condition, rated as fair to bad, according to the 2021 ASCE infrastructure report card. In particular, for pavements and bridges, 1 of every 5 miles of pavement in the United States is in poor condition, and 7.5% of the country's bridges have been structurally deficient over the last 20 years. Relevant agencies have thus increased interest in comprehensive decision-making strategies and solutions.

Accurate evaluation and prediction of infrastructure components performance and determination of optimal maintenance and inspection actions for any infrastructure element throughout its lifetime are essential parts of an effective infrastructure asset management framework. The goals of this project were to develop artificial lntelligence (AI)-enabled solutions, providing infrastructure condition assessment and prediction models, as well as algorithms able to directly suggest optimal maintenance and inspection decisions for multi-component infrastructure systems over long planning horizons.

BACKGROUND

Determination of inspection and maintenance policies to minimize long-term risks and costs in deteriorating engineering environments with various constraints constitutes a complex optimization problem. The major computational challenges include (i) the curse of dimensionality, due to exponential scaling of state/action set cardinalities with the number of components; (ii) the curse of history, related to exponentially growing decision trees with the number of decision steps; (iii) the presence of state uncertainties, induced by inherent environment stochasticity and variability of inspection/monitoring measurements; and (iv) the presence of constraints, pertaining to long-term stochastic limitations, due to resource scarcity and other infeasible/undesirable system responses. This class of hard optimization problems has been mostly tackled, as of now, by static age- or condition-based maintenance strategies, and risk-based or periodic inspection plans. However, such approaches can manifest various limitations related, among others, to optimality, scalability, and incorporation of incomplete information. In addition, many current optimization methodologies are based on unconstrained formulations, omitting the critical and often inevitable presence of constraints.

In this work, these challenges are addressed within a joint framework of *Partially Observable Markov Decision Processes* (POMDPs) (Madanat, 1993 ; Madanat & Ben-Akiva, 1994; Papakonstantinou & Shinozuka, 2014; Papakonstantinou, et al., 2018) and *multi-agent Deep Reinforcement Learning* (DRL) as in (Andriotis & Papakonstantinou, 2019; Andriotis & Papakonstantinou, 2021). POMDPs optimally tackle (ii)-(iii), combining stochastic dynamic programming with Bayesian inference principles. Multi-agent DRL addresses (i) through deep function parametrizations and decentralized control assumptions.

Challenge (iv) is herein handled through proper state augmentation and Lagrangian relaxation, with emphasis on lifecycle risk-based constraints and budget limitations. The underlying algorithmic steps are provided in this report and the proposed framework is found to outperform well-established policy baselines and facilitate adept prescription of inspection and intervention actions, in cases where decisions must be made in the most resource- and risk-aware manner.

In addition, accurate prediction of the performance of infrastructure components, including concrete bridge decks, is required when determining the maintenance and repair lifecycle actions that are to be performed. By using accurate estimates, agency costs due to maintenance, repair, and reconstruction, along with user costs, can all be minimized. Typically, the prediction of pavement conditions is utilizing statistical models; however, these methods restrain the deterioration modeling to follow a specific distribution, e.g., (Agarwal, et al., 2010; Sobanjo, et al., 2010; Manafpour, et al., 2018; Mishalani & Madanat, 2002). However, there is a vast amount of data available on infrastructure component performance, and AI techniques can hence be favorable in predicting future infrastructure conditions. To this end, in this report we explore and develop a *Random Survival Forest* approach (Lu & Guler, 2021) to effectively model the duration a bridge deck may be described by a certain condition rating (CR).

OBJECTIVES

In this work we are leveraging significant advances in AI methodologies to improve efficient life-long management of transportation infrastructure systems. The goal is to develop AI-enabled frameworks, able to provide infrastructure condition assessments and predictions, as well as direct optimal maintenance and inspection decisions/actions/policies for multi-component infrastructure systems. The key objectives are thus listed as follows:

(i) Improve predictive models for bridge deck conditions by using AI-enabled data analysis methods;

(ii) Further enhance our DRL framework by incorporating key financial, risk, and other related constraints, considered in any desired time period;

(iii) Suggest solutions and approaches toward a unified, integrated AI asset management framework;

(iv) Benchmark the developed algorithms against state-of-the-art and state-of-practice decision-making rules;

(v) Develop and validate approaches based on data, practices, standards, etc., by DOTs, with emphasis on PennDOT systems.

REFERENCES

Agarwal, A., Kawaguchi, A. & Chen, Z., 2010. Deterioration rates of typical bridge elements in New York. *Journal of Bridge Engineering,* 15, pp. 419-429.

Andriotis, C. P. & Papakonstantinou, K. G., 2019. Managing engineering systems with large state and action spaces through deep reinforcement learning. *Reliability Engineering & System Safety,* 191, p. 106483.

Andriotis, C. P. & Papakonstantinou, K. G., 2021. Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints. *Reliability Engineering & System Safety,* 212, p. 107551.

Lu, M. & Guler, S. I., 2021. Random survival forest for bridge deck deterioration. *Transportation Research Record (Under review).*

Madanat, S., 1993 . Optimal infrastructure management decisions under uncertainty. *Transportation Research Part C: Emerging Technologies,* 1(1), pp. 77-88.

Madanat, S. & Ben-Akiva, M., 1994. Optimal inspection and repair policies for infrastructure facilities. *Transportation Science,* 28(1), pp. 55-62.

Manafpour, A., Guler, I., Radlińska, A., Rajabipour, F., & Warn, G., 2018. Stochastic analysis and time-based modeling of concrete bridge deck deterioration. *Journal of Bridge Engineering,* 23(9), p. 04018066.

Mishalani, R. & Madanat, S., 2002. Computation of infrastructure transition probabilities using stochastic duration models. *Journal of Infrastructure Systems,* 8(4), pp. 139-148.

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

Papakonstantinou, K. G. & Shinozuka, M., 2014. Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part I: Theory. *Reliability Engineering & System Safety,* Volume 130, pp. 202-213.

Sobanjo, J., Mtenga, P. & Rambo-Roddenberry, M., 2010. Reliability-Based Modeling of Bridge Deterioration Hazards. *Journal of Bridge Engineering,* 15(6), pp. 671-683.

CHAPTER 2

Random Survival Forest Model for Bridge Deck Deterioration

The analysis of bridge deck deterioration is critical to infrastructure system management. Bridge deck deterioration models can predict the future conditions of assets and in turn guide rehabilitation programs and budget allocation to maximize the lifespan of bridges. Deterioration modeling can be done to model the lifecycle deterioration process or to model the deterioration probabilities from a specific condition rating to a lower CR for individual time steps. The present study focuses on the latter approach.

Interestingly, transportation infrastructure system management shares many similarities with medical research, since both focus on the survival probability of an asset or a patient, known as survival analysis. Classic models used for survival analysis include simple linear models, Kaplan-Meier estimator, Cox proportional regression, distribution-based stochastic models, etc. (Lunn, et al. 1995) (Goyal, Whelan and Cavalline 2019) (Manafpour, et al. 2018). With improvements in computational power, machine learning methods have started to demonstrate superiority over traditional models both in model accuracy and capability (Fathi, et al. 2019) (Assaad and El-adaway 2020). However, some advanced machine learning-based survival models, such as random survival forest (RSF), are only used in the medical field and their suitability in the infrastructure management area has not been examined. Thus, this project aimed at studying the suitability of RSF for bridge deck deterioration analysis and testing its performance compared to a traditional statistical method, i.e., Weibull distribution-based accelerated failure time model.

LITERATURE REVIEW

Survival models are different than most typical models, since the independent variable has two dimensions: (1) the duration that the object has been in a specific condition, known as the sojourn time; and (2) whether the entire duration of this sojourn time is observed or not, i.e., censoring. Censoring occurs when the start or endpoint of being in a condition is not observed, but the knowledge of the minimum duration an object was in a specific condition still provides valuable input into the model.

Generally, non-parametric and semi-parametric models can capture the deterioration process in a more realistic manner, since they are not required to follow a mathematical distribution (Goyal, Whelan and Cavalline 2019) (Mauch and Madanat 2001). However, these types of models assume that covariates increase or decrease hazard in a proportional manner, which is a significant weakness of these types of models such as the Cox regression. Parametric models are simple, efficient, and interpretive, but typically have low accuracy, since the real deterioration process is random, which is especially problematic when the data size is small. Commonly used distributions for parametric survival models include exponential, Weibull, Log-normal, gamma, and generalized gamma distributions (Manafpour, et al. 2018) (Pascoa, Ortega and Cordeiro 2011) (Edirisinghe, Setunge and Zhang 2013). The censored and uncensored data

This chapter is largely based on the journal paper: Lu, M., Guler, S.I. Random survival forest model for bridge deck deterioration. *Transportation Research Record*. Under Review.

are jointly considered in stochastic models by jointly using the hazard function (instantaneous probability of failure) for uncensored data and the reliability function (the cumulative probability of failure) for censored data in the estimation of the models (Goyal, Whelan and Cavalline 2019) (Pascoa, Ortega and Cordeiro 2011).

On the other hand, machine learning methods have been recently studied for modeling the deterioration of transportation infrastructure (Bashar and Torres-Machi 2021) (Contreras-Nieto, et al., 2018). One study compared five data mining techniques—logistic regression, decision trees, neural network, gradient boosting, and support vector machine—for steel bridge superstructure deterioration, and found that logistic regression achieved the highest prediction accuracy (Contreras-Nieto, et al., 2018). Another study showed that the back-propagation neural network can predict bridge deterioration with 75.4% accuracy (Ying-Hua 2010). Assaad and El-Adaway even observed a 91.44% testing accuracy for predicting the condition of a bridge deck using a well-tuned ANN model (Assaad and El-adaway 2020). Other commonly used machine learning approaches in infrastructure deterioration modeling include k-nearest neighbors, recurrent neural networks, and random forest, in which the ensemble learning algorithms (i.e., random forest) are believed to have superior performance (Piryonesi and El-Diraby 2019). Even though these machine learning methods have been shown to achieve good prediction accuracy, they only take traditional datasets as input. Therefore, these methods cannot incorporate censored data nor provide a complete deterioration probability curve for the entire analysis window.

Currently, two advanced machine learning methods that can model survival exist, namely random survival forest (RSF) and survival support vector machine (Survival-SVM). RSF is different from the traditional random forest (RF) in where the splitting role for partitioning the dataset and the predicted approach for the terminal leaves are adjusted to incorporate censored data and provide a complete deterioration probability curve. Survival-SVM is an extension of Rank SVM and only treats a pair of ranks as valid when the lower observed time is uncensored, since the exact duration of censored data is unknown (Belle, et al. 2008). Estimating a Survival-SVM can be very complex and time-consuming, especially when the kernel function is complex and the data size is large. Thus, RSF is more popular for survival analysis. However, to the authors' knowledge, these advanced machine learning-based survival models are only used in the medical area. Example applications include clinical risk prediction (Schmid, Wright and Ziegler 2016), survival prediction of breast cancer patients (Wright, Dankowski and Ziegler 2017), or comparison of survival from different illnesses (Nasejje, et al. 2017). Different applications have focused on tailoring the methods to the specific problem considered, such as determining how to best implement the splitting of the tree (Schmid, Wright and Ziegler 2016) (Wright, Dankowski and Ziegler 2017). Further, one study compared the RSF to a Cox regression and demonstrated that the two methods achieved compatible results in modeling breast cancer survival, while RSF showed a slightly better performance than other approaches (Omurlu, Ture and Tokatli 2009).

The use of survival machine learning methods in the area of infrastructure deterioration is unknown. Since the infrastructure deterioration and medical survival process are significantly different and possess different deterioration patterns, the appropriate implementation of survival machine learning methods for infrastructure deterioration needs to be studied.

Research objectives

Based on the literature review, the present study introduces the RSF model into the infrastructure management literature and adapts it for bridge deck deterioration analysis. The performance of RSF is compared to a traditional stochastic model to analyze the advantages of the different types of modeling approaches. Further, the RSF method's independent variable selection process is tailored to the bridge deck deterioration analysis.

METHODOLOGY

The primary model used in the present study is RSF. This section outlines the basic theory of RSF and the major difference from a traditional random forest methodology.

Random Survival Forest

RSF is a type of ensemble learning approach that combines a series of basic learners to improve predictive accuracy and robustness. The basic theory is similar to a traditional random forest. The general construction of a decision tree and a random forest is illustrated in Figure 1. A typical random forest can be described with the following hyperparameters: (a) number of estimators, (b) maximum depth, (c) minimum number of samples in a leaf, and d) the maximum number of features considered for splitting. The number of estimators determines the number of trees estimated to be combined for the random forest. As this number increases, the random forest model becomes more robust but also more complex and difficult to interpret. The maximum depth of the tree determines the number of layers considered. As the tree depth increases, each terminal leaf will be representative of a smaller subset of the data. The minimum number of samples in a leaf is also related to the tree depth, i.e., the tree will not be split further once the minimum number of samples in a leaf is met, even if the tree depth allows for further splitting. Finally, the maximum number of features considered for splitting represents the number of variables considered in each tree. While all variables are considered for a decision tree, in a random forest method, each tree only considers a subset of variables, which improves the robustness of the model and avoids overfitting. Detailed mathematical expressions can be found in the literature (Biau and Scornet 2016).

The traditional random forest is a powerful tool, but censored data cannot be incorporated into this model. Hence, RSF modifies traditional random forests to improve the splitting role, prediction method, and evaluation metric to be able to account for censored deterioration data. The other approaches used to improve an RSF, like bagging, boosting, and pruning, are similar to the traditional random forest, and thus the details are not repeated here (Ishwaran, et al. 2008).

Splitting role

The splitting role is used to partition the dataset into subsets that maximize the difference between and minimize the difference within each subset. As a result, observations that share similar characteristics are grouped into the same terminal node; thus, a prediction based on the observations within a terminal node can closely represent the data pattern of its members.

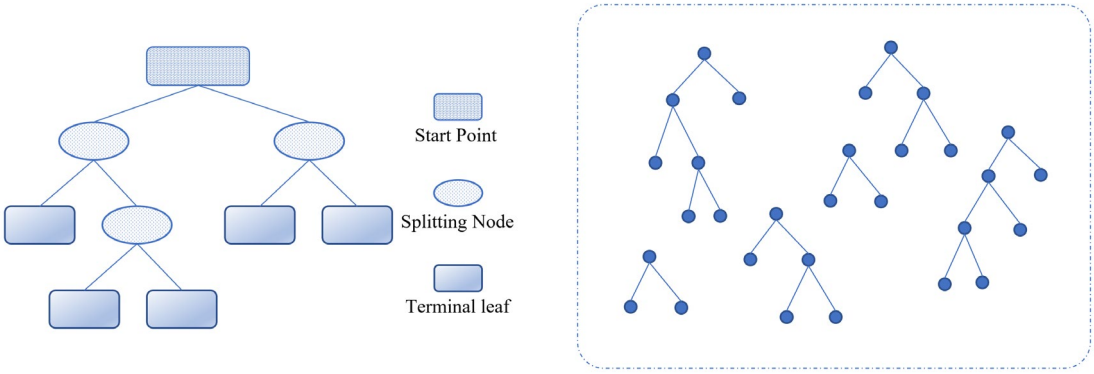


Figure 1. Architecture of a decision tree (left) and a forest (right).

In the traditional random forest, the common splitting roles, such as Gini index and entropy, aim at maximizing the similarity of the output within each subset. Normally, the output is a one-dimension variable and has clear equations to measure its similarity. However, for the survival data, the output is two-dimensional, and the final prediction is a complete deterioration probability curve. Thus, the splitting role in RSF should aim at generating subsets that have the most different deterioration patterns. The log-rank test is commonly used to quantify the difference in the deterioration patterns predicted from each subset (Wellek 1993). In the log-rank test, the null and alternative hypotheses are:

$\boldsymbol{H_0}$**:** The deterioration pattern in the two datasets is identical.

$\boldsymbol{H_1}$**:** The deterioration pattern in the two datasets is significantly different.

The test statistic function for the Z-value to accept the null hypothesis is shown in Equation (1):

$$Z = \frac{\sum_{i=1}^{k}(O_{1,i} - E_{1,i})}{\sqrt{\sum_{i=1}^{k} V_i}} \sim N(0,1) \tag{1}$$

where,

$O_{1,i}$ is the observed number of deaths at the time, t_i, in subset 1;

$E_{1,i}$ is the expected number of deaths at the time, t_i, in subset 1, which can be calculated as $E_{1,i} = \frac{d_i Y_{1,i}}{Y_i}$;

$Y_{1,i}$ is the number of individuals at risk (neither dead nor censored) at the time t_i in subset 1;

d_i is the number of individuals dead at the time t_i;

V_i is the variance of the observed number of deaths, which can be calculated as:

$$V_i = \frac{Y_{2,i} Y_{1,i} d_i (Y_i - d_i)}{Y_i^2 (Y_i - 1)} \tag{2}$$

Prediction method

The prediction method for the RSF relies on the use of a cumulative hazard function (CHF) that is measured using the Nelson-Aalen estimator. The cumulative hazard represents the aggregated hazard, or instantaneous risk of failure, over time. It can be interpreted as the number of times a failure (i.e., the condition rating lowering) would be expected over the analysis window. The Nelson-Aalen estimator is much focused on the hazard of the asset during the lifecycle. The CHF of the Nelson-Aalen estimator is:

$$\widehat{H}(t) = \sum_{t_i < t} \frac{d_i}{Y_i} \tag{3}$$

where,

t_i are the elements of all distinct event times;

d_i is the number of deaths at the time t_i;

Y_i is the total number of individuals at risk (neither dead nor censored) at the time t_i.

To predict the Nelson-Aalen estimator for a given terminal leaf, the hazard of all data that fall in that leaf are combined. This can be used to determine the risk of failure at a given time for a new bridge deck.

Evaluation metric

In order to understand how well a given RSF performs, the accuracy in prediction of that RSF needs to be determined. However, the prediction of the deterioration pattern for a single observation is not necessarily

meaningful, since the results are probabilistic. Therefore, the performance of the RSF method is evaluated by comparing the ranking of the predicted risk score to the actual survival data in the whole testing dataset. The risk score, r, is the total number of failures expected over the lifetime of the study for a given bridge deck with a set of attributes, $\boldsymbol{x}$, within the analysis window (Pölsterl 2020). This risk score can be estimated as the sum of the estimated CHF, $\hat{H}_h$, for a terminal node h as shown in Equation (4).

$$r = \sum_{j=1}^{n_h} \hat{H}_h(T_{hj}|\boldsymbol{x}) \tag{4}$$

where,
n_h is the number of distinct uncensored times of samples in terminal node h;
T_{hj} is the $j\ th$ item of these distinct uncensored times in terminal node h.

The model evaluation first predicts the risk scores for a set of bridges in the testing dataset. Next, these bridges are ranked by risk score from lowest to highest. Finally, the ranking of the real (observed) survival times are determined. The ranking from the model prediction is compared to the ranking from the real data to determine the concordance index (C-index), which reflects the ability of a survival model to predict the survival time rank based on the predicted risk scores. The C-index can be computed as Equation (5).

$$C = \frac{\sum_{i,j} I_{T_j<T_i} * I_{r_j<r_i} * \delta_j}{\sum_{i,j} I_{T_j<T_i} * \delta_j} \tag{5}$$

where,
r_i is the predicted risk score of a unit i;
$I_{T_j<T_i} = 1$ if $T_j < T_i$ else 0;
$I_{r_j<r_i} = 1$ if $r_j < r_i$ else 0;
δ_j denote the censorship of the data. $\delta_j = 0$ means uncensored data, $\delta_j = 1$ means censored data. The range of the C-index is from 0.5 to 1, where $C = 1$ corresponds to the best model prediction, and $C = 0.5$ represents a random prediction.

Finally, the C-index can also be rank the importance of different input variables in determining the survival curve. To do so, a permutation-based feature importance is calculated by measuring the C-index of the original model and comparing it to a shadow model. The shadow model is created by randomly shuffling the values of a given attribute in the training data and determining the C-index. The difference in the C-index of the original model and shadow model is assumed to be indicative of the importance of that variable in determining the final survival curve. Hence, the rank of features is determined as the ordered gains in C-index.

EXPERIMENTS AND RESULTS

To demonstrate the performance of RSF in bridge deck deterioration modeling, the dataset of 22,000 bridge decks' inspection records from Pennsylvania was adopted; a summary of the dataset is described in this section. First, the choice of independent variable is explored. Two candidate variables, namely sojourn time and cumulative truck traffic (CTT), are tested. Next, the hyperparameters of the RSF are calibrated, the best structure is demonstrated, and the model is compared to a commonly used stochastic model, namely the Weibull distribution-based accelerated failure time model.

Data description

The Pennsylvania Department of Transportation (PennDOT) conducts regular inspection of approximately 22,000 state-owned bridges at most every two years. In the inspection report, a general condition rating is assigned to a bridge deck to reflect the general condition of the bridge deck between 1 and 9, where condition rating 9 represents the best condition. In the present study, the dataset was separated into 9 subsets based on the condition rating. The sojourn times for each CR were extracted, including whether the sojourn time was censored or not. A sojourn time with an unobserved start point or endpoint or suffering from an incident or rehabilitation that significantly changed the CR was treated as a censored data point. After cleaning the raw data, valid information for 18,354 bridges was obtained, and a total of 44,086 sojourn times were extracted and classified given the CR. Summary statistics for the distribution of the sojourn times were determined as shown in Table 1.

It can be seen that only a few bridge decks have uncensored datapoints with condition rating lower than 4, since typically CR 1 through 3 are considered poor condition. Thus, models of CR 4 and higher are more reliable. In this study, the RSF was illustrated with the sub-dataset of CR 6 and from hereon in, deterioration probability refers to the probability of a bridge deck deteriorating from CR 6 to CR 5. However, note that similar results were obtained for other CR values.

The attributes of each bridge deck structural component are collected as covariates to correlate with the reliability of the bridge. The major attributes used in this study are summarized in Table 2.

Table 1. Statistic of sojourn times of bridge decks.

Condition	Censored	Censored	Censored	Uncensored	Uncensored	Uncensored
Rating	Count	Mean (days)	Std (days)	Count	Mean (days)	Std (days)
CR 1	19	2,809	2,509	2	1,040	348
CR 2	104	1,690	1,263	13	1,818	1,124
CR 3	1,007	2,022	1,709	170	2,034	1,421
CR 4	3,132	3,197	2,410	783	2,581	1,717
CR 5	6,016	4,010	2,759	2,317	2,935	1,794
CR 6	7,264	4,024	2,622	3,865	2,977	1,719
CR 7	8,636	3,957	2,603	3,817	3,054	1,760
CR 8	3,234	2,610	1,927	2,612	2,501	1,443
CR 9	654	1,420	1,106	381	1,747	1,049
Total	30,066	25,739	18,908	13,960	20,687	12,375

Independent variable selection

An appropriate choice of the independent variable can significantly improve the performance of the model. Sojourn time is commonly used for infrastructure management, which represents the duration for which a bridge has been in a specific condition. However, time-based deterioration models may not be able to fully capture the reliability of bridge decks, since the amount of truck traffic can significantly impact bridge deck deterioration. The average daily truck traffic (ADTT) varies across different bridges and influences the design of a bridge deck. Hence, in addition to the sojourn times, the cumulative truck traffic, defined as the

product of the sojourn time and average daily truck traffic, is proposed as an alternative independent variable.

Table 2. Attributes' description and values, including the count of each value in the dataset.

Attribute	Description	Values (Counts)
DISTRICT	District number	District 1 (2,707); District 2 (2,102); District 3 (3,171); District 4 (2,200); District 5 (2,215); District 6 (2,912); District 8 (5,369); District 9 (3,497); District 10 (2,443); District 11 (2,672); District 12 (2,817).
STRUC_TYP	Deck structure type	Concrete - Reinforced (26,324).
MAIN_MATERIAL_TYPE	Main materials type	Steel (8,531); Concrete (Cast in Place) (6,205); Concrete (Precast) (537); Prestressed Precast Concrete (P/S) (15,774); Concrete Encased Steel (982).
MAIN_PHYSICAL_TYPE	Physical makeup of the main span of the structure	Reinforced (6,744); Pretensioned (15,600); Rolled Sections (4,787); Rolled Sections with Cover Plates (1,174); Combination, Rolled Sections/Cover-Plates (334); Other (3,313).
MAIN_SPANS	Main bridge spans (number of spans in the main unit)	Single span (20,209); Multi-span (11,122).
MAIN_STRUC_CONFIG	Structural configuration for the main span of the structure	Slab (Solid) (2,378); T-Beams (3,985); I Beams (11,653); Box Beam - Single (5,681); Box Beam - Adj (6,614); I-Welded Beams (410); Girder Weld/Deck (722).
DECK_REBAR_TYPE	Deck rebar type	Bare Rebar Type (12,960); Galvanized Rebar Type (561); Epoxy Rebar Type (11,738); Unknown (6,794).
MEMBTYPE	Waterproofing membrane on the bridge main span	None (26,722); Preformed Fabric (3,816); Other (368).
SURF_TYPE	Wearing surface types on the bridge main span	Concrete Overlay (17,543); Epoxy Overlay (974); Bituminous (13,340).
LENGTH	Bridge length	The total overall length of the bridge.
DECK_WIDTH	Bridge deck width	Bridge deck width.
ADT_Total	Average daily traffic in total	Average daily traffic in total, including all types of vehicles.
ADTT	Average daily truck traffic	Average daily truck traffic.

Note: The Values (Counts) only show the values which count is larger than 1% of the whole dataset.

The reliability analysis for both independent variables, sojourn time and CTT, is demonstrated for three typical attributes, i.e., rebar type, span number, and surface type to compare the performance of the two independent variables.

Rebar type

Three different types of rebar used for bridge decks were compared: bare, epoxy-coated, and galvanized. Figure 2a shows the sojourn time of bridge decks with different CRs and rebar types and suggests that the

average sojourn time (~7 years) is mostly independent of rebar type. However, the number of bridges (Figure 2b) and the ADTT that a bridge experienced (Figure 2c) with different rebar types are significantly different. There are more bridges in higher condition ratings with galvanized or epoxy rebar, and these bridges often experience a higher ADTT. Thus, even though the bridge decks have similar sojourn times, the advantage of the galvanized or epoxy rebar might be compromised by the heavier traffic load. Thus, the sojourn time alone is not enough to reflect the reliability difference of different rebar types. However, the difference in reliability based on rebar type can be seen when considering CTT (Figure 2d). The results suggest that bridge decks with epoxy or galvanized rebar have higher reliability compared to the bare rebar bridges when considering CTT, which aligns with engineering judgment (Yeomans 2001).

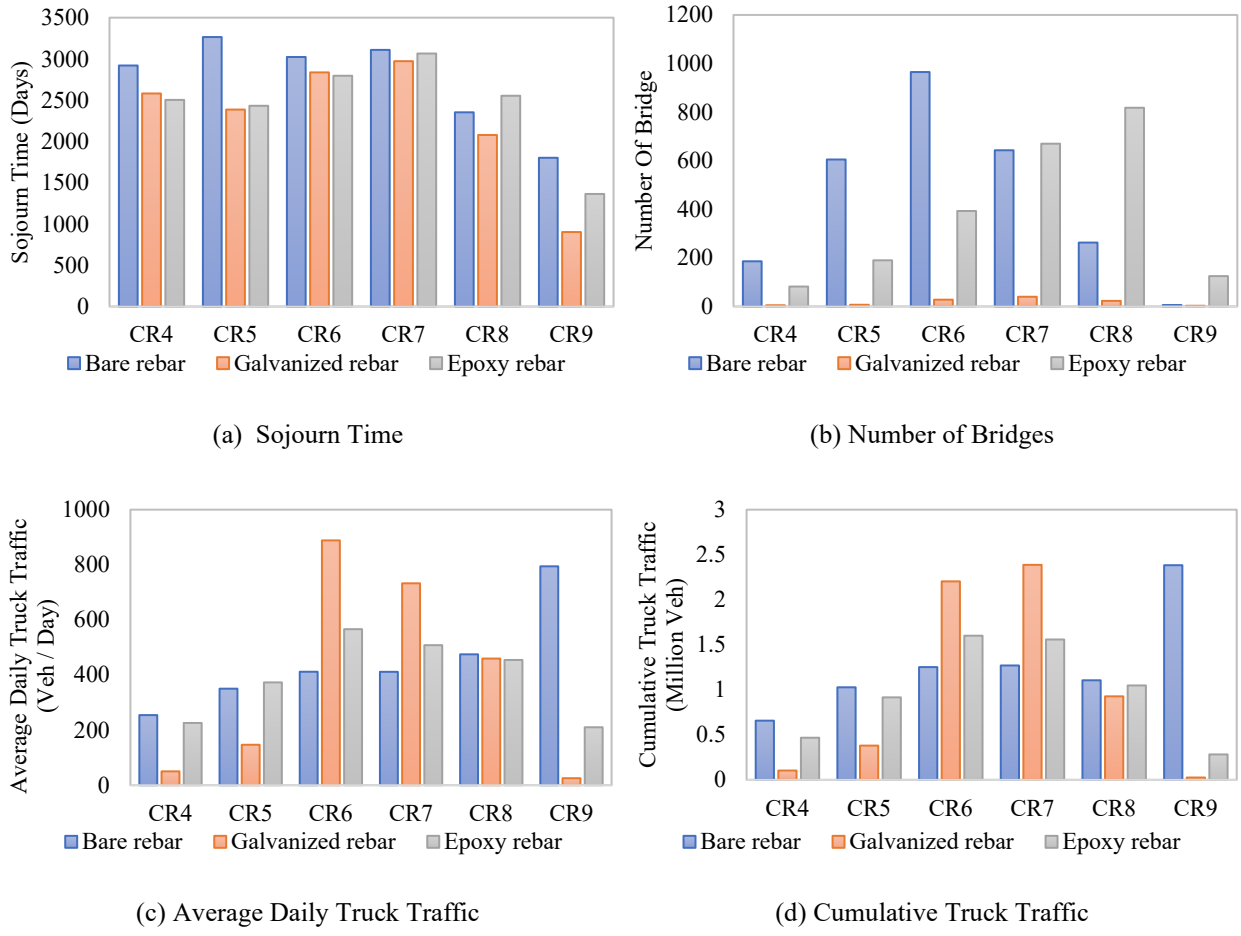
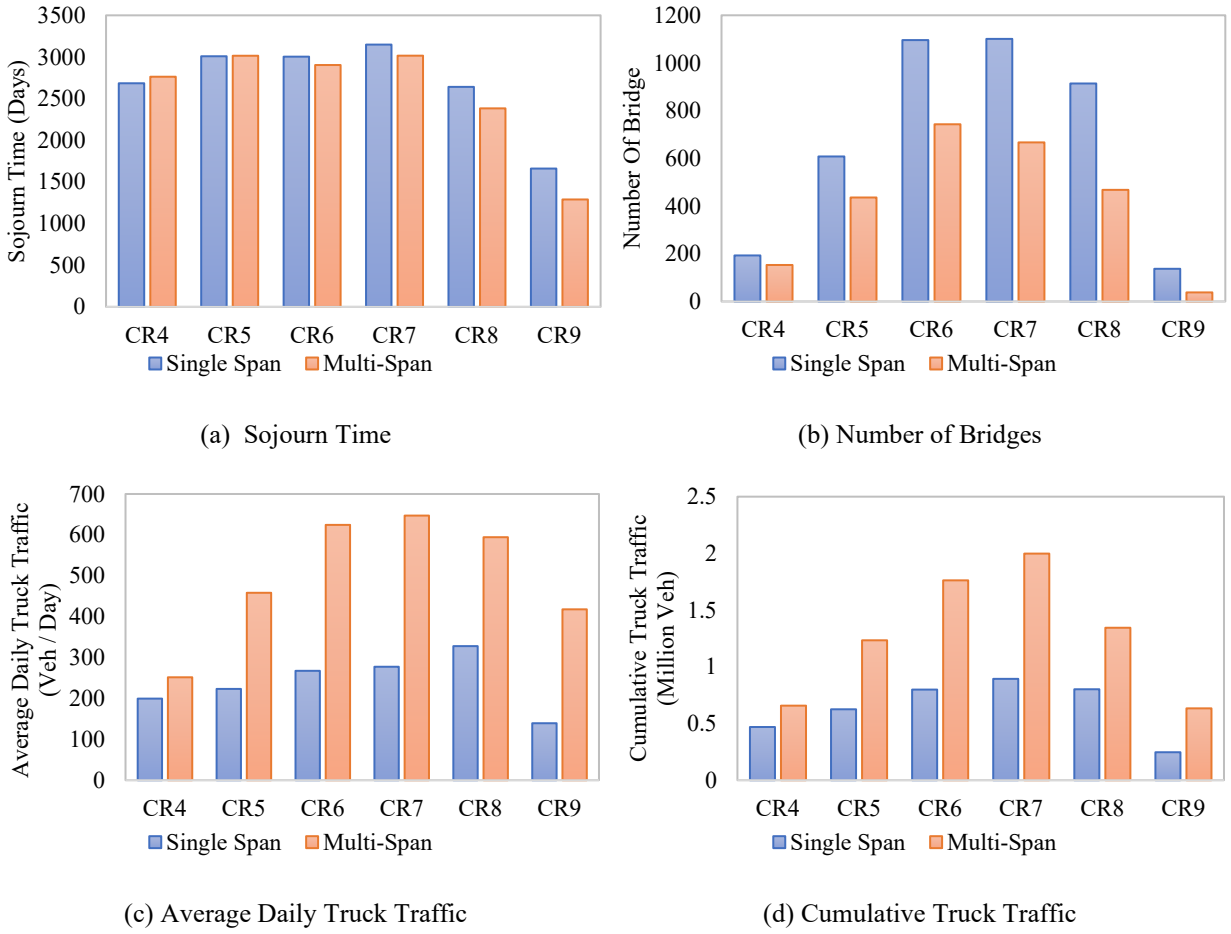


Figure 2. Distribution for rebar type by condition rating of (a) sojourn time, (b) number of bridges, (c) average daily truck traffic, and (d) cumulative truck traffic.



(a) Sojourn Time

(b) Number of Bridges

(c) Average Daily Truck Traffic

(d) Cumulative Truck Traffic

Figure 3. Distribution for span type by condition rating of (a) sojourn time, (b) number of bridges, (c) average daily truck traffic, and (d) cumulative truck traffic.

Span number

Single-span bridges were compared to multi-span bridges to see trends in deterioration. The average deck length of a single-span bridge is 42 ft, compared to 278 ft for a multi-span bridge. From the sojourn time distribution, single-span and multi-span bridges perform similarly. However, this does not necessarily indicate that a single-span bridge is as reliable as a multi-span bridge from an engineering perspective. Considering the number of bridges (see Figure 3b) and the ADTT (see Figure 3c), while there are more single-span bridges, the multi-span bridges carry more trucks. This implies that the multi-span bridges are mostly constructed in areas with heavy truck traffic but can still achieve similar sojourn times as single-span bridges. This indicates that the multi-span bridges have higher reliability than single-span bridges, which is confirmed by the CTT-based analysis (see Figure 3d).

Surface type

Overlays are used to remedy spalling and cracking for deteriorated bridge surfaces. Comparing the sojourn times for the three different overlay materials used (concrete, asphalt, and epoxy), it can be seen that bridge decks that have an asphalt overlay have on average an 8.8% greater sojourn time, see Figure 4a. However, only a few bridges have an epoxy overlay (see Figure 4b), and these bridges have larger daily truck traffic

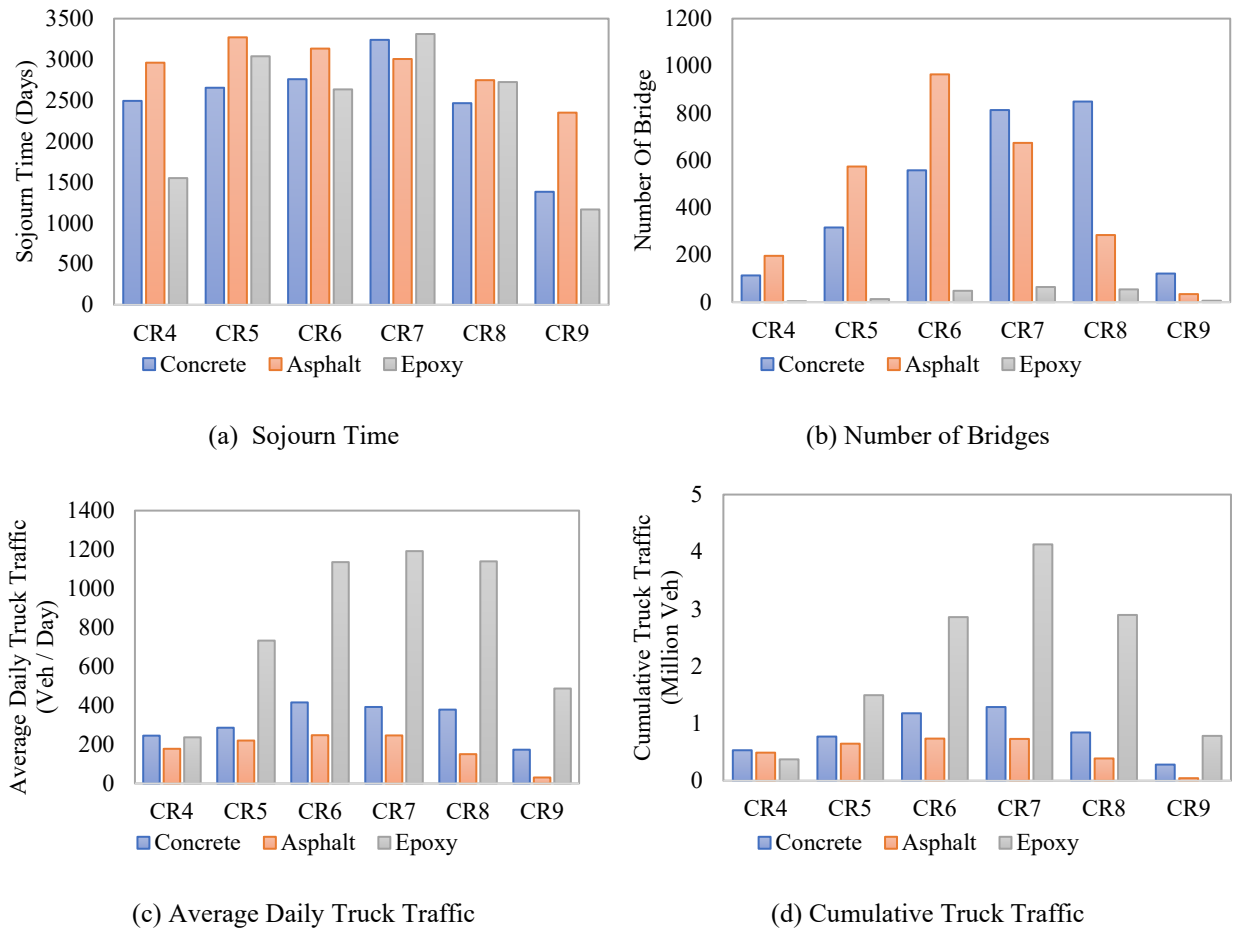
compared to bridges with concrete or asphalt overlay (see Figure 4c). Hence, when considering the CTT, bridges with epoxy overlay have the highest reliability, while bridges with asphalt overlay have the lowest reliability (see Figure 4d). The reliability of the different overlay material considering the CTT is more aligned with field experiments (Sprinkel 2001).

Overall, comparing the reliability of bridge decks with different rebar types, span numbers, and surface types reveals that CTT can better reflect the reliability of a bridge as compared to sojourn time and better match engineering judgment. Generally, it might be difficult to capture the reliability difference of the attributes considering only sojourn time, since the design choice is often influenced by expected traffic load.

General Machine Learning models

Initially, eight basic machine learning-based classifiers are used to model deterioration. These are:

1) K-nearest neighbors classifier (with a number of neighbors as 3);
2) Decision tree classifier (with maximum depth of the tree as 5);
3) Random forest classifier (with maximum depth of the tree as 5, the number of trees in the forest as 10, and the number of feathers to consider when looking for the best split as 1);



(a) Sojourn Time

(b) Number of Bridges

(c) Average Daily Truck Traffic

(d) Cumulative Truck Traffic

Figure 4. Distribution for surface type by condition rating of (a) sojourn time, (b) number of bridges, (c) average daily truck traffic, and (d) cumulative truck traffic.

4) Neural network (with the number of neurons in the hidden layer as 100, learning rate as 0.001);
5) Adaboost classifier;
6) Gaussian Naïve Bayes;
7) Quadratic discriminant analysis; and
8) Support Vector Machine with regularization parameter as 1 and kernel type is linear.

Each approach is tested on two datasets: the first dataset uses the entire dataset to train the model and then utilizes the same dataset for testing; the second dataset selects 2/3 of the whole dataset as the training dataset and the other 1/3 as the test dataset. Four tests are performed on each dataset: (1) all of the 58 input variables are used, (2) only selected bridges with reconstruction history are used to train the model (out of the 19,198 bridges in the inspection dataset, 3,393 of them have reconstruction records). These approaches are implemented on the dataset by the *Sk-learn* Python machine learning package. The correct prediction rates of each test are shown in Table 3.

Table 3. Results of basic machine learning methods.

	ML Approach	All Variables of All Bridges	All Variables of Bridges Reconstructed
Train - Train	Nearest Neighbors	66.26%	71.32%
	Decision Tree	43.29%	46.84%
	Random Forest	37.97%	41.99%
	Neural Net	40.82%	28.07%
	AdaBoost	38.27%	22.54%
	Naive Bayes	20.24%	26.73%
	QDA	2.79%	-
	SVM	-	-
Train - Test	Nearest Neighbors	36.01%	42.51%
	Decision Tree	42.58%	45.25%
	Random Forest	35.63%	39.98%
	Neural Net	38.97%	36.75%
	AdaBoost	28.80%	38.16%
	Naive Bayes	17.54%	27.34%
	QDA	5.07%	-

From the results, it can be seen that the Nearest Neighbors Classifier achieved the best results, especially for bridges that were reconstructed. The Nearest Neighbors Classifier achieves these results since its goal is to look for bridges with similar configurations to find patterns in the data that match closely.

However, as discussed earlier, even though these machine learning methods can achieve decent prediction accuracy, they cannot incorporate censored data nor provide a complete deterioration probability curve for the entire analysis window. Hence, next a Random Survival Forest model is implemented.

Random Survival Forest implementation

In this section, the RSF model is implemented considering both sojourn time and CTT as the independent variable to further compare their performance. When sojourn time is the independent variable, the ADTT is incorporated as one of the covariates, while when CTT is selected as the independent variable, ADTT is excluded from the model. 66.67% of the data is selected for training the model and the remainder is used for testing the performance of the model.

To achieve the best performance of the RSF, the hyperparameters need to be well-tuned. The main hyperparameters that need to be tuned include (a) the number of estimators, (b) maximum depth, (c) minimum samples in a leaf, and (d) the maximum features for splitting. A commonly used approach for tuning the hyperparameters of machine learning methods is grid-search, which is also used in the present study (Assaad and El-adaway 2020). The RSF model is implemented with the python package *scikit-survival* (Pölsterl 2020). The grid search space for these hyperparameters is shown in Table 4.
The hyperparameters of RSF that achieved the top five performances for the sojourn time-based model and CTT-based model are shown in Table 5.

Table 4. Grid search space for the hyperparameters of RSF.

Hyperparameters	Search space
Number of estimators	110, 120, 130, …, 300
Maximum depth	2, 3, 4, 5, …, 30
Minimum samples in a leaf	20, 40, 60, 80, …, 300
Maximum features for splitting	2, 3, 4, 5, …, 12

Table 5. Best model configuration from grid-search.

Independent Variable	Rank	Number of Estimators	Max Depth	Min Samples in a Leaf	Max # of Features for Splitting	C-index on Training Dataset	C-index on Testing Dataset
Sojourn time	1	200	12	40	5	0.6779	0.5898
Sojourn time	2	200	12	20	5	0.7299	0.5892
Sojourn time	3	200	12	60	5	0.6561	0.5812
Sojourn time	4	200	12	80	5	0.6428	0.5806
Sojourn time	5	190	12	200	5	0.6099	0.5801
CTT	1	200	12	200	8	0.8643	0.8336
CTT	2	200	12	40	5	0.8792	0.8322
CTT	3	200	12	60	5	0.8735	0.8319
CTT	4	200	12	200	6	0.8636	0.8316
CTT	5	200	12	80	5	0.8714	0.8310

The results suggest that the CTT-based models significantly outperformed the sojourn time-based model in terms of predictive score in both testing and training datasets. This further confirms that the

impacts of attributes on reliability can be more clearly differentiated based on its performance with respect to CTT. The best CTT-based model consisted of 200 estimators. As the number of estimators increases, the advantage of the ensemble approach becomes apparent and the accuracy of the model increases. However, increasing the number of estimators beyond 200 no longer improves the accuracy, but increases the complexity of the model. The optimal maximum depth is found to be 12, and the minimum number of samples in a leaf is found to be 200. These two hyperparameters together determine the size of each tree. When the tree is deep, the number of samples in a leaf becomes small, and the deterioration curve based on these samples becomes too specific and not representative of general categories. When the tree is too small, the dataset is not partitioned enough, and the information available from the covariates is not fully explored. The maximum number of features for splitting can help control the amount of randomness of the RSF. The optimal value of this is 8, which denotes that in each splitting node, the model randomly selects 8 out of the total of 12 features used in this study (as shown in Table 5) to search for the best splitting point. This helps build a diverse set of decision trees for the random forest and helps avoid overfitting.

With the well-tuned RSF model, the deterioration curve for a new observation can be predicted. The importance rank of each feature in the RSF determined as a result of the permutation-based feature importance is shown in Figure 5. As can be seen, the total amount of traffic has the highest importance in determining the CHF, followed by the deck width and the length of the bridge.

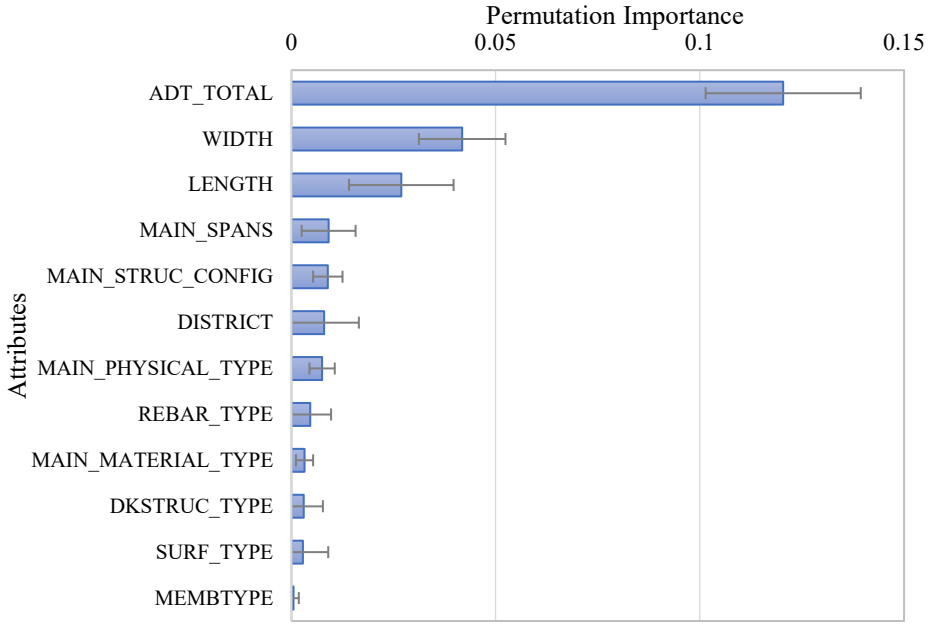


Figure 5. Permutation-based feature importance.

To further illustrate the deterioration curve of individual bridges, the first 5 observations in the testing dataset are used as an example, see Figure 6. This figure illustrates the CHF obtained by using the average of the deterioration curves from all the trees in the random forest. As can be seen, the RSF model can clearly differentiate between different shapes of CHF.

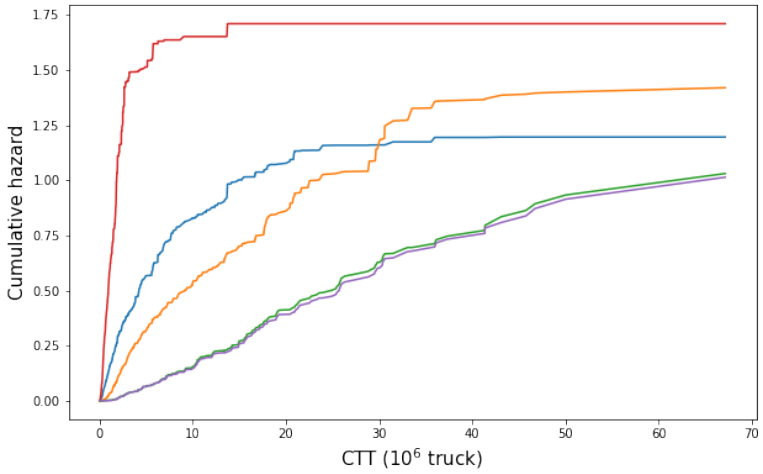


Figure 6. Cumulative hazard function prediction of new observation.

Comparison with AFT-Weibull model

To compare RSF to traditional deterioration models, a Weibull distribution-based accelerated failure time model is chosen as a benchmark, which is commonly used in infrastructure deterioration analysis (Manafpour, et al. 2018) (Andersson, Björklund and Haraldsson 2016). The AFT-Weibull model can take any bathtub shape distributions as the basic deterioration function and has more flexibility. The probability density function (PDF) of the Weibull distribution, $f(t)$, is shown in Equation (6):

$$f(t,\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{t}{\lambda}\right)^{k-1} e^{-\left(\frac{t}{\lambda}\right)^{k}} & t \geq 0 \\ 0 & t < 0 \end{cases} \tag{6}$$

where,
t is the independent variable, in this case, CTT;
λ and k are the parameters of the Weibull distribution, where λ can be replaced by the accelerated failure term, $e^{\boldsymbol{\beta X}}$, to incorporate the covariates. $\boldsymbol{\beta}$ is a vector of coefficients and $\boldsymbol{X}$ is the vector of covariates.

The probability of a bridge deck deteriorating to a lower CR can be modeled by the cumulative density function (CDF). The equation for the CDF of the Weibull distribution, $F(t)$, is shown as:

$$F(t) = 1 - e^{-\left(\frac{t}{\lambda}\right)^{k}} \tag{7}$$

The AFT-Weibull model is implemented on the same bridge deck deterioration data and is estimated with a python package *lifeline* considering CTT as the independent variable. Both censored data and uncensored data are incorporated by the probability density function and reliability function in the likelihood function (Ashraf-Ul-Alam and Khan 2021). The model is well tuned to achieve the highest possible reliability, and the parameters are estimated using the maximum likelihood estimation approach as shown in Table 6.

The results suggest that the district, rebar type, deck width, and total ADT are significant variables in predicting bridge deck deterioration. The surface type and member type, which are less important variables in the RSF model, were found to be insignificant in the Weibull model as well. Surprisingly, length, which was ranked third in the RSF model, was not found to be significant in the AFT-Weibull model.

The log-likelihood of the final model is -3312.80. A 10-fold cross-validation indicated that the C-index of the AFT-Weibull model in the training dataset is 72.34%, and the C-index for the prediction of the testing dataset is 70.05%. The accuracy of the AFT-Weibull model on the testing dataset is much lower than the accuracy of the RSF model, 83.36%, which could indicate that the RSF might be more powerful than stochastic models for predicting infrastructure deterioration.

Table 6. AFT-Weibull model coefficient estimations.

Attributes	Values	Coef.	SE (Coef.)	Z-value	p-value
DISTRICT	4	-0.21	0.11	-1.98	0.05
DISTRICT	5	-0.46	0.12	-3.77	<0.005
DISTRICT	6	-1.08	0.13	-8.65	<0.005
DISTRICT	9	-0.91	0.17	-5.38	<0.005
DISTRICT	11	-0.97	0.14	-7.11	<0.005
DISTRICT	12	-0.38	0.27	-1.39	0.16
STRUC_TYP	Concrete reinforced	0.43	0.22	1.97	0.05
MAIN_MATERIAL_TYPE	Prestressed precast concrete	-0.49	0.33	-1.5	0.13
MAIN_PHYSICAL_TYPE	Pretensioned	0.54	0.32	1.68	0.09
MAIN_PHYSICAL_TYPE	Rolled sections	-0.24	0.13	-1.84	0.07
MAIN_SPANS		0.04	0.02	1.78	0.07
MAIN_STRUC_CONFIG	Box beam - adj	-0.36	0.14	-2.59	0.01
REBAR_TYPE	Bare rebar	0.12	0.1	1.47	0.13
REBAR_TYPE	Galvanized rebar	0.18	0.11	1.66	0.1
WIDTH		0.01	0	2.03	0.04
ADT_TOTAL		5.2e-5	0	11.65	<0.005
Intercept		3.84	0.32	12.18	<0.005
k		0.19	0.03	5.69	<0.005

DISCUSSION

Sojourn time vs. CTT

The present study showed that CTT is more suitable to be the independent variable of bridge deck deterioration models compared to sojourn time, which is commonly used in survival analysis. From a practical perspective, this is due to the fact that the influence of sojourn time on reliability is compromised by the attributes values selection in the bridge design process. Engineers tend to select stronger materials or structures for bridges that are expected to experience heavier traffic, which leads to all bridges, regardless of the type of construction or materials, having a similar lifespan. Thus, it is difficult to distinguish reliability from a temporal perspective. However, the CTT reflects the actual load that a bridge experiences, which is the main contributor to deterioration (along with the environment), and thus can differentiate the reliability of a bridge more accurately. It is also feasible to use a CTT-based model for a real infrastructure

management process, since the traffic load of a bridge is usually closely observed, and the data is easily collected during a regular inspection.

RSF vs. AFT-Weibull

This study indicated that RSF has a better predictive power compared to the AFT-Weibull model. Another benefit of RSF compared to the AFT-Weibull model is that the importance of each feature ranking from RSF, shown in Figure 5, is helpful to determine which component is critical to a bridge deck design and maintenance. The magnitude of the permutation importance for the RSF is the C-index benefit of the corresponding feature. However, the interpretation of the impact of attribute values in RSF is not intuitive. On the other hand, the AFT-Weibull model is a typical parametric method, which is simpler and can be interpreted more easily. Since the coefficient for each attribute value is estimated, it is more suitable to analyze the impact on the reliability of different attributes' values. Take the coefficient estimations for the rebar type variable in Table 5 as an example. The coefficient estimation for bare rebar is 0.12, and the coefficient for galvanized rebar is 0.18. Both coefficients have low p-values, 0.13 and 0.10, respectively, which denote high confidence levels. Further, variables with larger coefficients represent a longer lifespan, hence galvanized rebar is found to be statistically more reliable than the bare rebar, which confirms the results found from the analysis of the raw data.

Overall, even though RSF outperformed the AFT-Weibull model in the respect of prediction accuracy, the model selection should be decided based on the research purpose, data quality, and application scenario.

CONCLUSIONS

This work introduced the random survival forest (typically used in the medical field) into infrastructure deterioration analysis and adapted it to bridge deck deterioration modeling. The use of sojourn time or cumulative truck traffic as the independent variable is considered. The experiment results suggest that CTT is more suitable to measure the reliability process of a bridge deck compared to sojourn time, since the CTT reflects that hazard exposure directly. The adapted RSF achieved a much higher predictive accuracy in the testing dataset when considering the CTT as the independent variable, 83.36% (C-index), as compared to considering the sojourn time as the independent variable, 58.98%. Further, the RSF model considering CTT as the independent variable also outperformed a representative stochastic model, the AFT-Weibull model, which also uses CTT as the independent variable, 70.05%. The results suggest that RSF has advantages in predicting the rank of risks of bridge decks, providing a complete deterioration probability curve, and determining the feature importance. The RSF model's use scenarios are different than stochastic models, since the RSF is a non-parametric method with higher predictive ability but lower interpretability of the impact of attributes' values.

Different enhancements of RSF should be considered for future studies, such as considering alternative splitting roles, boosting, bagging, and pruning techniques that are commonly used in the traditional random forest.

REFERENCES

Andersson, Mats, Gunilla Björklund, and Mattias Haraldsson. 2016. "Marginal railway track renewal costs: A survival data approach." *Transportation Research Part A-policy and Practice* 87: 68-77.

Ashraf-Ul-Alam, Md., and Athar Ali Khan. 2021. "Generalized Topp-Leone-Weibull AFT Modelling: A Bayesian Analysis with MCMC Tools using R and Stan." *Austrian Journal of Statistics* 50 (5): 52-76.

Assaad, Rayan, and Islam H. El-adaway. 2020. "Bridge Infrastructure Asset Management System: Comparative Computational Machine Learning Approach for Evaluating and Predicting Deck Deterioration Conditions." *Journal of Infrastructure Systems* 26 (3): 4020032.

Bashar, Mohammad Z., and Cristina Torres-Machi. 2021. "Performance of Machine Learning Algorithms in Predicting the Pavement International Roughness Index." *Transportation Research Record:* 0361198120986171.

Belle, Vanya Van, Kristiaan Pelckmans, Johan A. K. Suykens, and Sabine Van Huffel. 2008. "Survival SVM: a Practical Scalable Algorithm." *Proc. of the 16th European Symposium on Artificial Neural Networks (ESANN 2008).* 89-94.

Biau, Gérard, and Erwan Scornet. 2016. "A random forest guided tour." *Test* 25 (2): 197-227.

Contreras-Nieto, Cristian, Yongwei Shan, and Phil Lewis. 2018. "Characterization of Steel Bridge Superstructure Deterioration through Data Mining Techniques." *Journal of Performance of Constructed Facilities* 32 (5): 4018062.

Edirisinghe, Ruwini, Sujeeva Setunge, and Guomin Zhang. 2013. "Application of Gamma Process for Building Deterioration Prediction." *Journal of Performance of Constructed Facilities* 27 (6): 763-773.

Fathi, Aria, Mehran Mazari, Mahdi Saghafi, Arash Hosseini, and Saurav Kumar. 2019. "Parametric Study of Pavement Deterioration Using Machine Learning Algorithms." *Airfield and Highway Pavements 2019.* 31-41.

Goyal, Raka, Matthew Whelan, and Tara Cavalline. 2019. "Duration-based Forecasting of Bridge Condition with Non-Parametric Kaplan-Meier Survival Functions." *SMAR 2019 - 5th Conference on Smart Monitorig, Assessment, and Rehabilitation of Civil Structures*.

Ishwaran, Hemant, Udaya B. Kogalur, Eugene H. Blackstone, and Michael S. Lauer. 2008. "Random survival forests." *The Annals of Applied Statistics* 2 (3): 841-860.

Lunn, Mary, McNeil, and Don. 1995. "Applying Cox Regression to Competing Risks.", *Biometrics*: 524-532

Manafpour, Amir, Ilgin Guler, Aleksandra Radlińska, Farshad Rajabipour, and Gordon Warn. 2018. "Stochastic Analysis and Time-Based Modeling of Concrete Bridge Deck Deterioration." *Journal of Bridge Engineering* 23 (9): 4018066.

Mauch, Michael, and Samer Madanat. 2001. "Semiparametric Hazard Rate Models of Reinforced Concrete Bridge Deck Deterioration." *Journal of Infrastructure Systems* 7 (2): 49-57.

Nasejje, Justine B., Henry Mwambi, Keertan Dheda, and Maia Lesosky. 2017. "A comparison of the conditional inference survival forest model to random survival forests based on a simulation study as well as on two applications with time-to-event data." *BMC Medical Research Methodology* 17 (1): 115-115.

Omurlu, Imran Kurt, Mevlut Ture, and Füsun Tokatli. 2009. "The comparisons of random survival forests and Cox regression analysis with simulation and an application related to breast cancer." *Expert Systems With Applications* 36 (4): 8582-8588.

Pascoa, Marcelino A.R. de, Edwin M.M. Ortega, and Gauss M. Cordeiro. 2011. "The Kumaraswamy generalized gamma distribution with application in survival analysis." *Statistical Methodology* 8 (5): 411-433.

Piryonesi, S Madeh, and T. El-Diraby. 2019. "A Machine-Learning Solution for Quantifying the Impact of Climate Change on Roads." *CSCE Annual Conference*. Canadian Society for Civil Engineering, Laval.

Pölsterl, Sebastian. 2020. "scikit-survival: A Library for Time-to-Event Analysis Built on Top of scikit-learn." *Journal of Machine Learning Research* 21 (212): 1-6.

Schmid, Matthias, Marvin N. Wright, and Andreas Ziegler. 2016. "On the use of Harrell's C for clinical risk prediction via random survival forests." *Expert Systems With Applications* 63: 450-459.

Sprinkel, Michael. 2001. "Maintenance of Concrete Bridges." *Transportation Research Record* 1749 (1749): 60-63.

Wellek, Stefan. 1993. "A log-rank test for equivalence of two survivor functions." *Biometrics* 49 (3): 877-881.

Wright, Marvin N., Theresa Dankowski, and Andreas Ziegler. 2017. "Unbiased split variable selection for random survival forests using maximally selected rank statistics." *Statistics in Medicine* 36 (8): 1272-1284.

Yeomans, Stephen R. 2001. "Applications of Galvanized Rebar in Reinforced Concrete Structures." In *Corrosion 2001,* Houston, Texas.

Ying-Hua, Huang. 2010. "Artificial Neural Network Model of Bridge Deterioration." *Journal of Performance of Constructed Facilities* 24(6): 597–602.

CHAPTER 3

Deep Reinforcement Learning-driven Inspection and Maintenance Planning under Incomplete Information and Constraints

INTRODUCTION AND OVERVIEW

Optimal inspection and maintenance planning delineates a class of important engineering decision-making problems, aimed at supporting the sustainable and resilient operation of systems and networks over their lifecycle. Optimality refers to minimizing various societal, environmental, and economic risks, along with other operational costs, as these emerge due to the combined consequences of the selected actions of the decision-maker and their effects based on the future exogenous deterioration of the environment. Within this context, the goal of the decision-maker is to determine an appropriate policy, i.e. an optimal rule of sequential decisions over a presumed time frame, which is able to aptly map states and times to intervention and observation actions (Frangopol, et al., 2004; Sanchez-Silva, et al., 2016).

Literature indicates several approaches to solving this problem, from threshold-based nonlinear and mixed-integer programming formulations (e.g., in (Bocchini & Frangopol, 2011; Saydam & Frangopol, 2014; Yang & Frangopol, 2019; Marseguerra, et al., 2002)), to analysis of decision trees (e.g., in (Frangopol, et al., 1997; Faber & Stewart, 2003; Straub & Faber, 2005; Luque & Straub, 2019)), and from renewal theory (e.g., in (Grall, et al., 2002; Grall, et al., 2002; Castanier, et al., 2003; Rackwitz, et al., 2005)), to stochastic optimal control (e.g., in (Madanat, 1993 ; Ellis, et al., 1995; Papakonstantinou & Shinozuka, 2014; Papakonstantinou, et al., 2018)). These approaches are also applicable to infrastructure problems beyond inspection and maintenance planning, such as post-disaster recovery, e.g., in (Bocchini & Frangopol, 2012; González, et al., 2016; Nozhati, et al., 2020). Respectively, admissible solution strategies to the above approaches span from exhaustive policy enumeration and genetic algorithms to gradient-based schemes and dynamic programming. Besides formulations that leverage dynamic programming and stochastic optimal control concepts, a common characteristic underlying traditional inspection and maintenance planning methods is that the decision-making problem, despite its inherent sequential and dynamic nature, is articulated by means of static optimization formulations. As a result, many otherwise practical approaches tend to be more susceptible to optimality limitations, especially in problems with high-dimensional spaces and long decision horizons, challenges also known as the *curse of dimensionality* and *curse of history*, respectively (Bellman, 1957; Pineau, et al., 2003). Moreover, many solution techniques often lack cohesive and generalizable mathematical capabilities regarding the consistent integration of stochastic environments and/or uncertain observation outcomes in the optimization

This chapter is largely based on the journal paper: Andriotis, C.P., Papakonstantinou, K.G., 2021, "Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints," Reliability Engineering and System Safety, 212, 107551- 1:16.

process, as well as the incorporation of stochastic or deterministic constraints that need to be satisfied over multiple time steps or even the entire operating life of the system.

To address the above issues, this work follows a stochastic optimal control approach, casting the optimization problem within the joint context of constrained Partially Observable Markov Decision Processes and multi-agent Deep Reinforcement Learning. POMDPs are able to alleviate the curse of history as a result of their dynamic programming principles and to facilitate optimal reasoning in the presence of real-time noisy observations (Kaelbling, et al., 1998). Their efficiency in inspection and maintenance planning has been thoroughly studied and exemplified in (Papakonstantinou & Shinozuka, 2014; Papakonstantinou & Shinozuka, 2014; Papakonstantinou, et al., 2016; Memarzadeh & Pozzi, 2015; Schöbi & Chatzi, 2016), among others. Within the same class of applications, in the confluence of DRL and point-based POMDPs, the Deep Centralized Multi-agent Actor Critic (DCMAC) approach has been recently developed in (Andriotis & Papakonstantinou, 2019; Andriotis & Papakonstantinou, 2019b), an off-policy algorithm with experience replay, belonging in the general family of actor-critic approaches (Wang, et al., 2016; Degris, et al., 2012). DCMAC leverages the concept of belief-state MDPs, a fundamental idea for the development of point-based POMDP algorithms, thus directly operating on the posterior probabilities of system states given past actions and observations (Shani, et al., 2013). In DCMAC, individual control units are centralized in terms of global state information and sharing of policy network parameters; nonetheless, they are *decentralized* in terms of policy outputs. Hence, based on classic Markov decision processes formalism, DCMAC provides Decentralized POMDP (Dec-POMDP) solutions (Oliehoek & Amato, 2016; Bernstein, et al., 2002) for a setting where the agents representing the various control units have access to the entire state distribution of the system, however, having the autonomy to make their own choices without being aware of each other's actions. DRL is extremely efficient in tackling the curse of dimensionality stemming from high-dimensional and/or combinatoric state spaces, whereas the computational hurdle of exponential scaling of the number of actions with the number of components is seamlessly handled by the decentralized multi-agent formulation of the problem, given that decentralization enables linear scaling.

Building upon the above-described DRL concepts in this work, a modified architecture in relation to the original DCMAC approach is implemented for the actor. We consider a sparser parametrization of the actor, without parameter sharing, i.e., each agent has its own individual policy network. We call this architecture Deep Decentralized Multi-agent Actor Critic (DDMAC). Similar approaches exist for various cooperative/competitive multi-agent robotic and gaming control tasks (Gupta, et al., 2017; Baker, et al., 2019). Thorough reviews on state-of-the-art methods and applications can be also found in (Oroojlooyjadid & Hajinezhad, 2019; Hernandez-Leal, et al., 2019). Despite the architectural differences with DCMAC, DDMAC solves the same Dec-POMDP problem, eliminating, however, inter-agent interactions in the hidden layers for the sake of computational efficacy. Based on this numerical approach, this report is particularly focused on investigating the effects of incorporating resource constraints and other limitations, especially in the forms of budget and lifecycle risk constraints. Depending on the nature of the modeled limitations, the constraints can be addressed through either state augmentation or primal-dual optimization approaches based on the Lagrangian function of the problem.

Constrained static optimization formulations for operation and maintenance policies exist in the literature, e.g., in (Bocchini & Frangopol, 2011; Rackwitz, et al., 2005; Goulet, et al., 2015; Sørensen, 2009), mainly reflecting short-term risk, reliability-based, and budget-related considerations. In the case of POMDPs, the optimization problem now falls in the category of constrained POMDPs. Constrained Markov decision processes have been given model-based solutions with the aid of linear programming formulations in (Altman, 1999; Poupart, et al., 2015). Exact POMDP alpha-vector value iteration can be extended to constrained problems as well, inheriting, however, the PSPACE complexity of the unconstrained solution (Isom, et al., 2008). Unconstrained point-based POMDP algorithms, which are well-suited for inspection and maintenance planning of systems with up to thousands of states and hundreds of actions and observations (Papakonstantinou, et al., 2016; Papakonstantinou, et al., 2018), have also been extended to

constrained problems (Kim, et al., 2011). In multi-component systems, under the assumption of component-wise independent cost functions, states, and actions, (Walraven & Spaan, 2018) derives constrained POMDP solutions through a series of unconstrained solutions controlled by a linear master program. Overall, and notwithstanding their principled mathematical descriptions, the above value iteration and linear or nonlinear programming formulations are fundamentally hard to extend to high-dimensional systems that are of interest in this work.

In DRL, constraints typically refer to either the parameters of the approximated functions, or the cumulative returns related to auxiliary functions of interest (Schulman, et al., 2015; Achiam, et al., 2017; Zhang, et al., 2020). The former methods restrain the iterate increment of the policy parameter updates to be within a trust region of the Kullback-Leibler divergence between the new and the old policy, thus preventing abrupt policy changes and, consequently, training instabilities. In such cases, optimization is typically based on surrogates of the objective and constraint functions (Schulman, et al., 2015). The latter methods typically aim to protect the agent from unsafe or otherwise undesirable states and choices during training or policy deployment. To this end, the objective is optimized with the aid of primal-dual formulations, either through trust region concepts, or Lagrangian relaxation, or domain-based manual penalization (Achiam, et al., 2017; Tessler, et al., 2018; Peng, et al., 2018). Safe RL formulations similarly integrate risk and policy variance in the constraint functions of the problem, or directly intervene in exploration to guide training (Garcıa & Fernández, 2015; Chow, et al., 2017). Such "safety" constraints can, for example, pertain to the probability of failure over multiple steps and, as such, they reflect soft constraints, meaning that they only need to be satisfied in a probabilistic or expected sense. The satisfaction of hard constraints, such as budget constraints, is easier to account for in the optimization process through state augmentation. Such constraints tend to be relevant for other resource limitations as well (e.g., in cases of limited availability of operating crews, inspectors, etc.). In this work, we consider and study both types of constraints.

In summary, in this chapter we consider and optimize DRL-driven, non-periodic inspection and maintenance policies in the presence of resource limitations and risk-related constraints. First, the preliminaries of the POMDP formulation in inspection and maintenance planning are elaborated, with insights in the problem-specific modeling requirements. State updating equations and inspection, maintenance, shutdown, and risk cost definitions are presented. It is studied and discussed how the selected actions affect the above costs, and which inherent mechanisms drive observational strategies in POMDPs are. Theoretical analysis pertaining to risk definitions and related accruable and instantaneous costs is presented, along with their relation to classical definitions. The optimization problem is cast within the context of decentralized multi-agent DRL control, where agents operate directly on the belief space (i.e., the space of posterior system statistics based on past actions and observations). The developed and employed DRL approach, DDMAC, is an off-policy actor-critic method with experience replay, modifying the original architecture presented in (Andriotis & Papakonstantinou, 2019). The relevant algorithmic steps for implementing the above-described decentralized DRL framework are provided, based on state augmentation for hard constraints and Lagrangian relaxation for soft constraints. Quantitative investigation is conducted based on a stochastically deteriorating multi-component system. Numerical experiments include evaluation of different baseline policies, and different budget and risk constraint scenarios. The resulting evolution of various system metrics, pertaining to risk, reliability, inspection, and intervention choices over the system operating life, is parametrically studied and discussed based on the learned policies.

The optimization problem

The goal of the decision-maker (agent) in a lifecycle inspection and maintenance optimization problem is to determine an optimal policy $\pi = \pi^*$ that minimizes the total cumulative future operational costs and risks in expectation:

$$\begin{aligned}\pi^* &= \underset{\pi\in\Pi_C\subseteq\Pi}{\arg\min}\, \mathbb{E}_{s_{0:T},o_{0:T},a_{0:T}}\left[\sum_{t=0}^{T}\gamma^t c_t \,\middle|\, a_t \sim \pi\left(o_{0:t}, a_{0:t-1}\right), s_0 \sim \mathbf{b}_0\right] \\ &= \underset{\pi\in\Pi_c\subseteq\Pi}{\arg\min}\, V^{\pi}(\mathbf{b}_0)\end{aligned} \tag{1}$$

where $c_t = c(s_t,a_t,s_{t+1})$ is the cost incurred at time t by taking action $a_t \in A$, and transitioning from state $s_t \in S$ to state $s_{t+1} \in S$; $o_t \in \Omega$ is an observation outcome; $\gamma \in [0,1]$ is the discount factor translating future costs to current value; $\mathbf{b}_0$ is an initial distribution over states (or initial belief); V^{π} is the value function, which expresses the total discounted cost given a state or a belief under policy π; and T is the length of the planning horizon. Planning horizon T can be either finite or infinite. A finite horizon problem can be solved as an infinite one, through proper formulation of the problem, i.e., through augmenting the state space with time, and considering an absorbing state at the final time step (Bertsekas, 2005).

Policy π is a rule according to which actions are taken by the decision-maker at different time steps, and it can be, at best, a map from histories of actions and observations to actions, π: $A^{t-1} \times \Omega^{t} \rightarrow A$. The policy function belongs to a space, $\pi \in \Pi_c$, which contains all possible policies that are admissible under the existing constraints of the problem. Π_c is a subset of Π, which is the policy space of the unconstrained problem. From the mapping a policy function conducts, it can be observed that the number of possible policies can easily become immense, even in problems with small planning horizons. Also known as the curse of history, this problem is optimally tackled by dynamic programming and POMDPs as explained in detail in the next section. Another approach to attack this complexity, however, often at the expense of solution efficiency, is to exploit problem-specific characteristics and employ simplified assumptions, including approaches that impose action periodicity, policy uniformity among components, component prioritization, ranking, or clustering (Grall, et al., 2002; Nicolai & Dekker, 2008; Memarzadeh, et al., 2016; Bismut & Straub, 2018; Rokneddin, et al., 2013; Zhang & Alipour, 2020). Particularly in inspection planning, periodic inspection visits or non-periodic inspections that exploit similarity and/or prioritization of components is typical for deteriorating structural systems (Luque & Straub, 2019; Bismut & Straub, 2018).

Policy π can also be stochastic, in which case it is a mapping to a probability distribution over actions, i.e. π: $A^{t-1} \times \Omega^{t} \rightarrow P(A)$. It can be shown under loose regularity conditions about the cost function that the optimal policy in a Markov decision process is deterministic (Putterman, 1994). However, in general and especially in the presence of constraints, the optimal policy is more broadly described by functions accounting for stochastic mappings (Altman, 1999).

Mapping posterior state distributions to actions

In a POMDP environment, transition from state $s_t = s$ to state $s_{t+1} = s'$ is Markovian. Detaching the effect of the maintenance action from the environment transition (natural deterioration), we can define an intermediate state, $s_t^a = s^a \in S$. This state succeeds s, with probability $\Pr(s^a|s,a)$, and reflects the system state immediately after maintenance and before the environment transition. This distinction is important to help us better define and quantify the risk in the next section, and additionally allows consideration of the probability of unsuccessful or partially successful actions. State s' succeeds s^a with probability $\Pr(s'|s^a,a)$,

after the environment transition, i.e., $s' = s^{a,e}$. Owing to the Markovian property, given a pair (s,a), the probability distribution of s' can be fully defined, regardless of the prior history of actions and states as:

$$\Pr(s'|s,a) = \sum_{s^a \in S} \Pr(s'|s^a,a)\Pr(s^a|s,a) \tag{2}$$

Similarly, the cost at a certain time step can be expressed as:

$$c(s,a,s') = \sum_{s^a \in S} \Pr(s^a|s,a)c(s,a,s^a,s') \tag{3}$$

where, for notational brevity, c on the right-hand side pertains to cost that additionally depends on s^a. State augmentation can be applied if higher-order temporal dependencies exist regarding the history of states and/or actions prior to t, or the environment is characterized by non-stationarity (Bertsekas, 2005; Papakonstantinou & Shinozuka, 2014). In POMDPs, at each time step, states are hidden to the agent, and are only perceivable through the noisy observation $o_t=o \in \Omega$. Observation o depends on the state of the system and the respective action at the current step, and is defined by probability $\Pr(o|\, s,\, a)$. The entire process described above is depicted in the network of Figure 7.

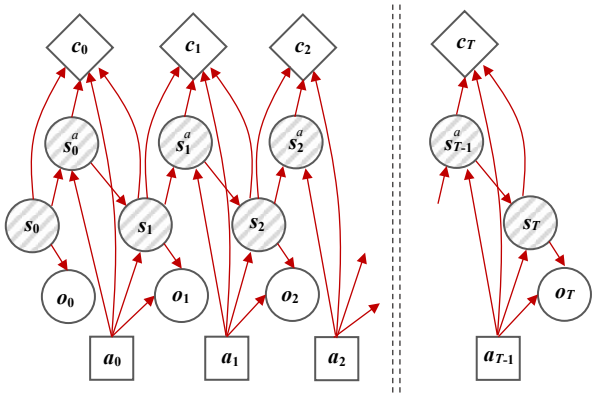


Figure 7. POMDP diagram in time, including intermediate states occurring after actions and before environment transitions.

As a result of the structure of POMDPs, optimal policy π^* can be defined, without any loss of information, as a function of belief $\mathbf{b}_t=\mathbf{b} \in B$: $S \rightarrow P(S)$, which is a sufficient statistic of the entire history of previous actions and observations, up to time t. Belief $\mathbf{b}$ is thus the posterior probability distribution over states, given the previous belief, and the current transition, action, and observation. Hence, the belief at time $t+1$, $\mathbf{b}_{t+1}=\mathbf{b}'=\mathbf{b}^{a,e,o}$, is computed by the Bayesian update:

$$
\begin{aligned}
b'(s') &= b^{a,e,o}(s') \\
&= \Pr(s' \mid o', a, \mathbf{b}) \\
&= \frac{\Pr(o' \mid s', a)}{\Pr(o' \mid \mathbf{b}, a)} b^{a,e}(s') \\
&= \frac{\Pr(o' \mid s', a)}{\Pr(o' \mid \mathbf{b}, a)} \sum_{s^a \in S} \Pr(s' \mid s^a, a) b^a(s^a)
\end{aligned}
\tag{4}
$$

where probabilities $b(s)$, for all $s \in S$, form the $|S|$-dimensional belief vector $\mathbf{b}$. The denominator of Eq. (4) $\Pr(o'|\mathbf{b},a)$, is the standard normalizing constant:

$$
\Pr(o' \mid \mathbf{b}, a) = \sum_{s' \in S} \Pr(o' \mid s', a) \sum_{s^a \in S} \Pr(s' \mid s^a, a) b^a(s^a) \tag{5}
$$

Similarly to s^a, belief b^a in Eqs. (4) and (5) is the intermediate belief, right after the maintenance action and before the environment transition and observation, defined as:

$$
b^a(s^a) = \sum_{s \in S} \Pr(s^a \mid s, a) b(s) \tag{6}
$$

In the special case that the environment is fully observable, i.e. $o = s$, observation specifies exactly which one of the belief vector entries is 1, assigning 0 otherwise. This defines an MDP environment and, accordingly, $\Pr(o'|\mathbf{b},a)$ reduces to $\Pr(s'|\mathbf{b},a)$, which is the transition probability of MDPs given the current state distribution. Following this remark, it is apparent that $\Pr(o'|\mathbf{b},a)$ holds transition probability semantics for the belief space, B, hence a POMDP can be seen as a belief-MDP, where now, however, states are the belief vectors. Accordingly, the transition between beliefs is given as:

$$
\Pr(\mathbf{b}' = \mathbf{x} \mid \mathbf{b}, a) = \sum_{o' \in \Omega} \delta_{\mathbf{b}'\mathbf{x}} \Pr(o' \mid \mathbf{b}, a) \tag{7}
$$

where δ_{ij} is the Kronecker delta, i.e. $\delta_{ij}=1$ if $i=j$, 0 otherwise.

This allows us to write the optimality equation, also known as the Bellman equation (Bellman, 1957), in the belief space as:

$$
\begin{aligned}
V(\mathbf{b}) &= HV(\mathbf{b}) \\
&= \min_{a \in A} \{Q(\mathbf{b}, a)\} \\
&= \min_{a \in A} \left\{ c_b + \gamma \sum_{o' \in \Omega} \Pr(o' \mid \mathbf{b}, a) V(\mathbf{b}) \right\}
\end{aligned}
\tag{8}
$$

where $V(\mathbf{b})=V^{\pi^*}(\mathbf{b})$ is the optimal *value function*, representing the total lifecycle cost under the optimal policy π^* given an initial belief $\mathbf{b}$, H is the Bellman backup operator, Q is the optimal *action-value function*, and c_b is the expected cost at belief $\mathbf{b}$, defined as:

$$
\begin{aligned}
c_b &= c_b(\mathbf{b}, a) \\
&= \mathbb{E}_{s,s^a,s'}\left[c(s,a,s^a,s')\right] \\
&= \sum_{s\in S} b(s) \sum_{s^a \in S} \Pr(s^a \mid s,a) \sum_{s'\in S} \Pr(s' \mid s^a, a)\, c(s,a,s^a,s')
\end{aligned}
\tag{9}
$$

Operator H is a contraction with unique fixed-point $V(\mathbf{b})$. It has been shown that the POMDP cost value function described by the Bellman equation in Eq. (8) is piece-wise linear and concave (convex for the maximization problem) at every time step, composed of linear hyperplanes, also called the *alpha-vectors* (Sondik, 1971). Each alpha-vector corresponds to an inspection and maintenance action (Papakonstantinou & Shinozuka, 2014; Papakonstantinou, et al., 2016).

Despite its analogies with MDPs, Eq. (8) is hard to solve exactly through standard MDP-based approaches, e.g., through value or policy iteration. However, there are numerous efficient approximate solution procedures along the lines of *point-based* algorithms (Shani, et al., 2013). Point-based algorithms sample a subset of the reachable belief space, starting from an initial root belief, thus making value iteration scale linearly with the cardinality of this subset. DRL is used for solving Eq. (8) in this work, using the point-based belief MDP concept combined with deep function approximations and actor-critic training (Andriotis & Papakonstantinou, 2019).

Risks and costs

Cost at different time steps for a selected action can be decomposed into inspection cost, c_I, maintenance cost, c_M, and damage state cost, c_D. In addition, it is often important for the decision-maker to account for the possibility of additional losses due to intentional system shutdowns, c_S, which may occur not as a consequence of damage, but rather as a result of the selected actions. Accounting for this as well, the total cost at each decision step can be generally expressed as:

$$
c(s,a,s^a,s') = \underbrace{c_M(s,a)}_{\text{maintenan. cost}} + \underbrace{c_S(s,a)}_{\text{shutdown cost}} + \gamma \underbrace{c_I(s',a)}_{\text{inspec.cost}} + \gamma \underbrace{c_D(s^a,s')}_{\text{damage state cost}}
\tag{10}
$$

Using Eq. (10) in Eq. (9), the expected inspection, maintenance, and shutdown costs can be written as:

$$
\begin{aligned}
c_{b,X} &= \mathbb{E}_s\left[c_X(s,a)\right] = \sum_{s\in S} b(s)\, c_X(s,a), \;\; X \in \{M,S\} \\
c_{b,I} &= \mathbb{E}_{s'}\left[c_I(a,s')\right] = \sum_{s'\in S} b^{a,e}(s')\, c_I(a,s')
\end{aligned}
\tag{11}
$$

Although Eq. (11) provides a broad description of the cost function, it is often appropriate to adopt the hypothesis that inspection and maintenance actions affect the respective costs independently, and are also independent of the system state (this hypothesis is stronger for inspections, since certain maintenance actions may depend on the extent of damage in the system):

$$
\begin{aligned}
c_I(s',a) &= c_{b,I}(a) = c_I(a_I) \\
c_M(s,a) &= c_{b,M}(a) = c_M(a_M)
\end{aligned}
\tag{12}
$$

where $a_I \in A_I$ is the selected inspection action and $a_M \in A_M$ is the selected maintenance action. Under this distinctive consideration of actions, the total action can be defined as $a \in A = A_I \times A_M$. We will refer here to no inspection and no maintenance actions as *trivial inspection* and *trivial maintenance actions*, respectively.

Trivial actions may also refer to routine maintenance and inspection actions, which are actions that are always taken at every time step, thus their costs do not affect the optimization process. Similarly to Eq. (12), it is also reasonable to assume in many problems of inspection and maintenance planning that scheduled shutdowns will be primarily triggered by maintenance actions only, namely:

$$c_S\left(s,a\right)=c_S\left(s,a_M\right) \tag{13}$$

Damage state cost c_D translates various losses associated with the damage states of the system to cost units. These can be grouped into two types of losses, which we will refer to as *instantaneous losses* and *accruable losses*. Instantaneous losses refer to costs incurred upon arrival at a damage state and do not continue to be collected for as long as the system sojourns this damage state. In the case of a failed civil engineering structure, for instance, such costs can be related to capital-related losses, which occur at the time step at which the structural system transitions to the *failure* state. This cost is collected once over the operating life, unless the system is restored and fails again. Accruable losses, on the other hand, refer to costs collected for as long as the system sojourns a certain damage state, regardless of which damage state it transitioned from. In the previously mentioned example of a failed civil engineering structure, such costs can be related to economic losses due to downtime, which are, of course, not instantaneous but accrue over time, until the system is restored to an operating status. Following this distinction, the damage cost component of Eq. (10) is written as:

$$c_D\left(s^a,s'\right)=c_D^{acc}\left(s'\right)+d_{s^a s'}c_D^{inst}\left(s'\right) \tag{14}$$

where $[d_{ij}]_{i,j\ \epsilon\ S}$ is the adjacency matrix pertaining to damage states, as this can be derived by state connectivity according to available actions. That is, if there is an action such that state j is an immediate successor of i, then d_{ij}=1. For i=j, d_{ij}=0. In deteriorating environments, it commonly happens that states are ordered; that is, transitions from s^a to s' form an upper-triangular transition matrix, meaning that the system can only transition to a worse state, or at best remain at the same one, due to environment effects. In this case, the adjacency matrix will be strictly upper-triangular.

As implied by Eq. (14), the cost of accruable losses is a function of the next state, s', whereas the part of instantaneous losses depends on the current state after the effect of the maintenance action, s^a, and the next state. The expected costs in Eq. (14), which is required to solve Eq. (8), with the aid of Eq. (9), give the step or interval risk as:

$$\begin{aligned} c_{b,D} &= c_{b,D}^{acc}+c_{b,D}^{inst} \\ &= \mathbb{E}_{s'}\left[c_D^{acc}\left(s'\right)\right]+\mathbb{E}_{s^a,s'}\left[d_{s^a s}c_D^{inst}\left(s'\right)\right] \\ &= \sum_{s^a\in S} b^a\left(s^a\right)\sum_{s'\in S}\Pr\left(s'\mid s^a,a\right)\left(c_D^{acc}\left(s'\right)+d_{s^a s'}c_D^{inst}\left(s'\right)\right) \end{aligned} \tag{15}$$

Using Eq. (15), risk is defined as the expected cumulative discounted damage state cost over the lifecycle:

$$\Re^{\pi}=\mathbb{E}_{o_{0:T}}\left[\sum_{t=0}^{T}\gamma^t\mathbb{E}_{s_t^a,s_{t+1}}\left[c_D^{acc}\left(s_{t+1}\right)+d_{s_t^a s_{t+1}}c_D^{inst}\left(s_{t+1}\right)\right]\right] \tag{16}$$

Quantification of risk is only relevant to the post-maintenance configuration of the system, thus from s^a. Note that if risk is quantified from s instead, it can take unrealistic negative values, since state s' can be of

lower damage. To better understand Eq. (16), one can consider a case where the system may suffer only instantaneous losses due to failure with cost c_F. In this case, Eq. (16) reduces to:

$$\Re_F^\pi = c_F \mathbb{E}_{o_{0:T}} \left[\sum_{t=0}^{T} \gamma^t \left(P_{F_{t+1}|a_{0:t},o_{0:t}} - P_{F_t|a_{0:t},o_{0:t}} \right) \right] \tag{17}$$

where P_{Ft} is the probability of failure up to time t. The specialized definition of risk provided by Eq. (17) follows standard risk and reliability assumptions and is well-studied in inspection and maintenance planning (Luque & Straub, 2019). The proof that Eq. (16) reduces to Eq. (17) under the above-stated assumptions is presented in Appendix A of (Andriotis & Papakonstantinou, 2021). This work employs the risk definition of Eq. (16) instead of that of Eq. (17), as it facilitates a broader consideration of losses related to multiple system states as in (Andriotis & Papakonstantinou, 2021).

Similarly, the other step costs of Eq. (10) assume the following expected cumulative discounted values over the lifecycle:

$$\begin{aligned} C_X^\pi &= \mathbb{E}_{o_{0:T}} \left[\sum_{t=0}^{T} \gamma^t \mathbb{E}_{s_t} \left[c_X \left(s_t, a_t \right) \right] \right], \; X \in \{M, S\} \\ C_I^\pi &= \mathbb{E}_{o_{0:T}} \left[\sum_{t=0}^{T} \gamma^t \mathbb{E}_{s_{t+1}} \left[c_I \left(a_t, s_{t+1} \right) \right] \right] \end{aligned} \tag{18}$$

Hence, the optimal POMDP value with its optimality equation described in Eq. (8) is:

$$\begin{aligned} V(\mathbf{b}) &= \min_{\pi \in \Pi_C \subseteq \Pi} \left\{ V^\pi(\mathbf{b}) \right\} \\ &= \min_{\pi \in \Pi_C \subseteq \Pi} \left\{ C_M^\pi + C_S^\pi + \gamma C_I^\pi + \gamma \Re^\pi \right\} \end{aligned} \tag{19}$$

Thus, overall, the problem of Eq. (1) consists in jointly minimizing the above lifecycle cumulative discounted costs.

OPERATING UNDER CONSTRAINTS

We consider the following form of the stochastic optimization problem of Eq. (1):

$$\begin{aligned} \pi^* &= \arg\min_{\pi \in \Pi} \mathbb{E}_{s_{0:T}, o_{0:T}, a_{0:T}} \left[\sum_{t=0}^{T} \gamma^t c_t \,\middle|\, a_t \sim \pi(o_{0:t}, a_{0:t-1}), s_0 \sim \mathbf{b}_0 \right] \\ \text{s.t. } G_{h,k} &= \sum_{t=0}^{T} \gamma^t g_{h,k}(s_t, a_t) - \alpha_{h,k} \le 0, \; k = 1,..,K \\ G_{s,m} &= \mathbb{E}_{s_{0:T}, o_{0:T}, a_{0:T}} \left[\sum_{t=0}^{T} \gamma^t g_{s,m}(s_t, a_t, s_{t+1}) \right] - \alpha_{s,m} \le 0, \; m = 1,..,M \end{aligned} \tag{20}$$

where $G_{h,k}$ and $G_{s,m}$ are the *hard* and *soft* constraints, respectively, $g_{h,k}$ and $g_{s,m}$ are their respective auxiliary costs (e.g. c_M, c_I, c_S, c_D, or else), and $\alpha_{h,k}$, $\alpha_{s,m}$ are real-valued scalars. The form of constraints in Eq. (20) is amenable to a broad class of constraint types that are relevant to infrastructure management. For example, hard constraints can model a great variety of fixed-resource allocation and control action availability problems, such as problems referring to strict budget limitations. In turn, soft constraints, of the Eq. (20)

form, can model a great variety of risk-based constraints. More details about these can be found in Section 3.2. The term *soft* constraints, although not standard in stochastic optimization and optimal control literature, is used here to distinguish from the term *hard* constraints, indicating that the underlying constraints do not need to be strictly satisfied, but are rather imposed in an expected or probabilistic fashion.

Hard constraints can be straightforwardly taken into account through state augmentation. On an interesting remark, in one of his notes on dynamic programming under constraints in 1956 (Bellman, 1956), R. Bellman mentions that this approach may not be favored since "due to the limited memory of present-day digital computers, this method founders on the reef of dimensionality." However, this restriction has been widely lifted today, whereas DRL has diminished the effects of the curse of state dimensionality even further. Thus, state augmentation is followed for the hard constraints here. Note that in the special case where functions $g_{s,m}$ are deterministic, soft constraints become hard. However, soft constraints are not practical to consider through state augmentation, since one should track the entire distribution of the cumulative discounted value of $g_{s,m}$. Therefore, probabilistic constraints are addressed here through Lagrangian relaxation (Bertsekas, 1999). Based on the above, the optimization problem is restated as:

$$\begin{aligned} V &= \max_{\lambda_1,\ldots,\lambda_M \geq 0} \min_{\pi\in\Pi} \mathbb{E}_{s_{0:T},\mathbf{y}_{0:T},o_{0:T},a_{0:T}} \left[\sum_{t=0}^{T} \gamma^t \left(\bar{c}\left(s_t, a_t, s_{t+1}, \mathbf{y}_t\right) + \sum_{m=1}^{M} \lambda_m g_{s,m} \right) - \sum_{m=1}^{M} \lambda_m \alpha_{s,m} \middle| a_t \sim \pi\left(o_{0:t}, a_{0:t-1}, \mathbf{y}_t\right), s_0 \sim \mathbf{b}_0, \mathbf{y}_0 \right] \\ &= \max_{\lambda_1,\ldots,\lambda_M \geq 0} \min_{\pi\in\Pi} V_\lambda^\pi(\mathbf{b}_0, \mathbf{y}_0) \\ \text{s.t. } & \mathbf{y}_t = \{y_{kt}\}_{k=1}^{K}, y_{kt} = \sum_{\tau=0}^{t-1} \gamma^\tau g_{h,k}\left(s_\tau, a_\tau\right), y_{k0} = 0, \\ & y_{kt} \in \left[0, \alpha_{h,k}\right], k = 1,\ldots,K \end{aligned} \tag{21}$$

where variables y_{kt} track the discounted cumulative value of the function related to hard constraints, $g_{h,k}$, up to time step t-1, and $\bar{c}$ is the cost function also considering y_{kt}. Variables y_{kt} are upper bounded by $a_{h,k}$. Lagrange multipliers, λ_m, constitute the dual variables of the max-min dual problem, they are positive scalars, and are associated with the soft constraints.

Budget constraints

Depending on the operational and resource allocation strategy of the management agency, available funding for inspection and maintenance must comply with certain short-term or long-term goals related to a specific budget cycle duration, T_B. Namely, in the extreme case of a short-term budget cycle duration, budget caps exist for every decision step (e.g., annual budget), whereas in the extreme case of a long-term budget cycle duration, there is a budget cap pertaining to the cumulative inspection and maintenance expenses over the entire lifecycle of the system, i.e., T_B=T. The cumulative cost of inspection and maintenance actions over period T_B is given for:

$$g_h\left(s_\tau, a_\tau\right) = \left(c_M + \gamma c_I\right)\mathbf{1}_{\tau\in\Lambda_t} \tag{22}$$

$$\Lambda_t = \left(\lfloor t / T_B \rfloor T_B, \left(\lfloor t / T_B \rfloor + 1\right) T_B\right] \tag{23}$$

where $\lfloor x \rfloor$ is the integer part of x. For a given budget cap α_h, the maintenance and inspection costs at each time step read:

$$\begin{aligned} \bar{c}_M &= \mathbf{1}_{y + g_h \leq \alpha_h} c_M \\ \bar{c}_I &= \mathbf{1}_{y + g_h \leq \alpha_h} c_I \end{aligned} \tag{24}$$

According to Eqs. (22)-(24), inspection and maintenance costs are accounted for only at the current budget cycle, and if the currently selected action does not violate the budget cap. The total cost at each time step of Eq. (10) is accordingly rewritten as:

$$\bar{c}_t = \bar{c}_M\left(s_t, a_t\right) + \gamma \bar{c}_I\left(a_t, s_{t+1}\right) + c_S\left(s_t, a_t\right) + \gamma c_D\left(s_t^a, s_{t+1}\right) \tag{25}$$

Transition and observation probabilities are also affected by the presence of the budget constraints as:

$$\begin{aligned}
&\Pr\left(s^a \mid s, y, a\right) = \mathbf{1}_{y+g_h \le \alpha_h} \Pr\left(s^a \mid s, a\right) + \left(1 - \mathbf{1}_{y+g_h \le \alpha_h}\right) \Pr\left(s^a \mid s, a_o\right) \\
&\Pr\left(s' \mid s^a, y, a\right) = \mathbf{1}_{y+g_h \le \alpha_h} \Pr\left(s' \mid s^a, a\right) + \left(1 - \mathbf{1}_{y+g_h \le \alpha_h}\right) \Pr\left(s' \mid s^a, a_o\right) \\
&\Pr\left(o' \mid s', y, a\right) = \mathbf{1}_{y+g_h \le \alpha_h} \Pr\left(o' \mid s', a\right) + \left(1 - \mathbf{1}_{y+g_h \le \alpha_h}\right) \Pr\left(o' \mid s', a_o\right)
\end{aligned} \tag{26}$$

where a_o is the trivial decision, where no inspection and no maintenance are performed. As indicated by Eqs. (21)-(26), incorporation of budget constraints can be accomplished by accounting for new state variables $y=y_t$. This way the agent is able to reason about control actions based on the available budget, $\alpha_h - y_t$, at each time step of the decision process. In the case of step-wise budget constraints, i.e., $T_B=1$, this state augmentation is not necessary, since the agent does not need to track any inspection and maintenance expenses made in the past, thus having the entire amount of α_h at its disposal for every single step.

As opposed to state variables s_t, new variables y_t are fully observable. In this regard, the problem can also be seen as a mixed observability Markov decision process, which admits favorable state decompositions and can be solved by value iteration algorithms in settings with moderate dimensions (Papakonstantinou, et al., 2018). In this case, constrained value iteration based POMDP solution procedures devised for constrained problems can be employed to drive the optimization process (Isom, et al., 2008; Kim, et al., 2011; Walraven & Spaan, 2018). However, as for the unconstrained case, such formulations can manifest limitations related to efficient scaling in systems with large state and action spaces, like the systems that are typically encountered in the class of sequential decision-making for infrastructure and networks.

Risk-based constraints

For notational efficiency of the present section, we introduce the following random variables:

$$\begin{aligned}
&J_i^\pi = \sum_{t=0}^{T} \gamma^t c_i\left(s_t, a_t, s_{t+1}\right),\ i \in \{M, I, S, D\} \\
&J^\pi = \sum_i J_i^\pi
\end{aligned} \tag{27}$$

where J_M^π, for example, accumulates total costs, related to maintenance actions over the lifecycle, and $\mathbb{E}_{s_{0:T}, o_{0:T}, a_{0:T}}[J_M^\pi] = C_M^\pi$ according to the definitions of Eq. (11).

We are now interested in incorporating constraints that bound risk over the system lifecycle. The risk-related random variable based on Eqs. (16) and (27) is J_D^π. Thus, the respective constraint function obtains the following form, for $g_s=c_D$ in Eq. (20):

$$
\begin{aligned}
G_s &= \mathbb{E}_{s_{0:T},o_{0:T},a_{0:T}}\left[J_D^{\pi}\right]-\alpha_s \\
&= \mathbb{E}_{s_{0:T},o_{0:T},a_{0:T}}\left[\sum_{t=0}^{T}\gamma^t\left(c_D^{acc}\left(s_{t+1}\right)+d_{s_t^a s_{t+1}}c_D^{inst}\left(s_{t+1}\right)\right)\right]-\alpha_s \\
&= \Re^{\pi}-\alpha_s
\end{aligned}
\tag{28}
$$

It should be noted that, although the budget constraints of focus in this work are not soft, budget constraints can also be expressed through G_s constraints, satisfied in expectation, depending on the modeling needs of the problem, as in Eq. (28). Any other costs as introduced in Eq. (10) can be considered in the same logic as well.

Constraints of the generic G_s form are also the *chance* or *probabilistic* constraints, which bound the probabilities of certain quantities or events (Garcıa & Fernández, 2015; Chow, et al., 2017). As such, if one wants to bound the probability of the optimal policy exceeding a certain lifecycle cost threshold J_{cr}, one may apply the following g_s function for any J_i^{π} similarly to Eq. (28):

$$
g_s = \mathbf{1}_{t=T}\cdot\mathbf{1}_{J_i^{\pi}>J_{cr}} \tag{29}
$$

where the second indicator signifies the cumulative cost constraint violation, and the first one ensures that this is taken into account once, at the end of the planning horizon. Taking the expectation of cumulative value of the constraint function of Eq. (29), we have:

$$
\begin{aligned}
G_s &= \mathbb{E}_{s_{0:T},o_{0:T},a_{0:T}}\left[\sum_{t=0}^{T}\mathbf{1}_{t=T}\cdot\mathbf{1}_{J_i^{\pi}>J_{cr}}\right]-\alpha_s \\
&= \Pr\left(J_i^{\pi}>J_{cr}\right)-\alpha_s
\end{aligned}
\tag{30}
$$

Considering Eq. (30), if α_s =1, we end up with a hard constraint requirement, i.e., $J_i^{\pi}>J_{cr}$. It is thus obvious that hard constraints can also be seen as a limiting case of soft constraints.

From a stricter reliability standpoint, many decision problems are interested in bounding the probability of failure (i.e., the probability reaching a failure state s_F from a non-failure state) over the system operating life. In this case, we just need to set $c_D^{per}=0$, γ=1, and $c_D^{inst}=\delta_{s_{t+1}s_F}$ in Eq. (28):

$$
\begin{aligned}
G_s &= \mathbb{E}_{s_{0:T},o_{0:T},a_{0:T}}\left[\sum_{t=0}^{T}d_{s_t^a s_{t+1}}\delta_{s_{t+1}s_F}\right]-\alpha_s \\
&= P_{F_T}-\alpha_s
\end{aligned}
\tag{31}
$$

P_{FT} is the probability of failure up to the end of the lifecycle t=T. Scalar α_s in Eq. (30) and (31) is a valid probability designating the (1 - α_s) percentile of risk and probability of failure, respectively, the decision-maker is willing to tolerate.

Other relevant constraint definitions in stochastic optimization and constrained Markov decision processes literature include constraints on the *value-at-risk* and *conditional-value-at-risk* (Uryasev & Rockafellar, 2002; Chow, et al., 2017) (with the former coinciding with probabilistic constraints), constraints on the policy variance (Di Castro, et al., 2012; Prashanth & Ghavamzadeh, 2016), as well as constraints whose satisfaction is implicitly encouraged through reward-based penalization (Smith, et al., 1995).

Constrained control with deep reinforcement learning

In recent work by the authors (Andriotis & Papakonstantinou, 2019; Andriotis & Papakonstantinou, 2019b), the Deep Centralized Multi-agent Actor Critic approach has been proposed for management of large engineering systems, shown to significantly outperform traditional maintenance and inspection decision rules. DRL approaches in general, either in the form of actor-critic, or policy gradients, or Q-learning, e.g. (Andriotis & Papakonstantinou, 2019; Rocchetta, et al., 2019; Liu, et al., 2020; Skordilis & Moghaddass, 2020), offer several computational advantages in high-dimensional state spaces, due to the fact that function parametrization over the state space alleviates the need for exhaustive state exploration. In addition, DCMAC concurrently accounts for partial state observability and high-dimensional action spaces. Its multi-agent formulation treats system control units as individual agents making decentralized decisions based on shared/centralized system information and actor-network hidden layer parameters. Control units are defined in reference to system parts for which separate actions apply at each decision step, and can be either individual system components or greater subsystem parts comprised of multiple components. As such, one control unit has at least one component, and one component may belong to more than one control units. The system policy function is written as:

$$\pi\left(\mathbf{a}\,|\,\hat{\mathbf{b}},\mathbf{y}\right)=\prod_{i=1}^{N_{CU}}\pi_i\left(a^{(i)}\,|\,\hat{\mathbf{b}},\mathbf{y}\right) \tag{32}$$

where $\mathbf{a}$ is a vector of actions and $\hat{\mathbf{b}}$ is a 2D matrix, such that:

$$\begin{aligned}\mathbf{a}&=\left[a^{(i)}\right]_{i=1}^{N_{CU}}\\ \hat{\mathbf{b}}&=\left[\mathbf{b}^{(j)}\right]_{j=1}^{N_C}\end{aligned} \tag{33}$$

where $a^{(i)}$ is the action of control unit i, $\mathbf{b}^{(j)}$ is the belief of system component j, N_{CU} is the number of control units, and N_C is the number of system components.

The policy functions of Eq. (32), as well as a centralized system Lagrangian value function are parametrized with the aid of deep neural networks as:

$$\begin{aligned}\pi_i\left(a^{(i)}\,|\,\hat{\mathbf{b}},\mathbf{y}\right)&\simeq\pi_i\left(a^{(i)}\,|\,\hat{\mathbf{b}},\mathbf{y},\boldsymbol{\theta}_\pi^{(i)}\right)\\ V_\lambda^\pi\left(\hat{\mathbf{b}},\mathbf{y}\right)&\simeq V_\lambda^\pi\left(\hat{\mathbf{b}},\mathbf{y}\,|\,\boldsymbol{\theta}_V\right)\end{aligned} \tag{34}$$

Parameters $\boldsymbol{\theta}_\pi^{(i)}$, $\boldsymbol{\theta}_V$ are real-valued vectors, and can either vary or be shared among control units. In either case, each control unit's policy is conditioned on the global belief and the budget-related states. Note that here we have a separate policy network for each agent, as denoted by superscript i in the policy parameters of Eq. (37), thus a completely decentralized actor parametrization is used. To distinguish this from the original DCMAC architecture, we call this Deep Decentralized Multi-agent Actor Critic (DDMAC). As discussed in Section 1, both provide decentralized POMDP policy solutions. The respective architectures are shown in Figure 8. In this figure, four components are depicted, and each component is a control unit, thus N_{CU}=N_C. DDMAC is trained based on off-policy experiences retrieved from the replay memory or replay

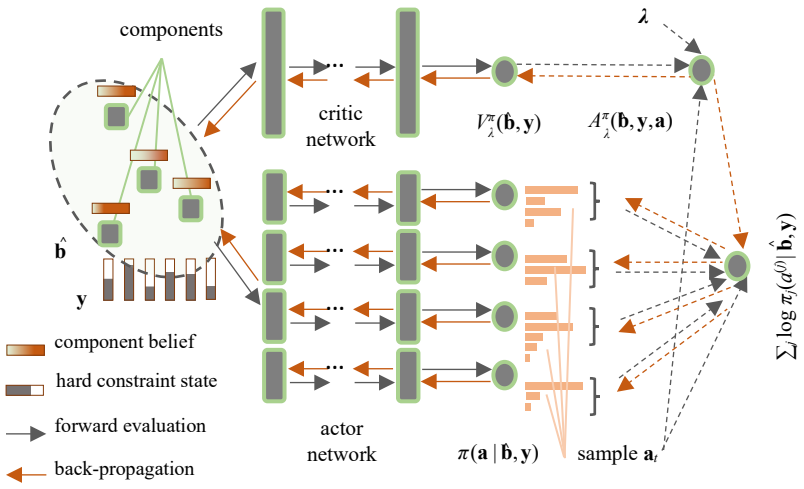


Figure 8. Constrained Deep Decentralized Multi-agent Actor Critic (DDMAC) architecture.

buffer, as agents interact with the environment. Thus, the replay memory is a stack of transition tuples.

The off-policy gradients of the policy function and the value function are computed by importance sampling as:

$$\nabla_{\boldsymbol{\theta}_\pi^{(i)}} V_\lambda^\pi = \mathbb{E}_{\mathcal{M}}\left[w \left(\sum_{i=1}^{N_{CU}} \nabla_{\boldsymbol{\theta}_\pi^{(i)}} \log \pi_i \left(a^{(i)} \mid \hat{\mathbf{b}}, \mathbf{y}, \boldsymbol{\theta}_\pi^{(i)} \right) \right) A_\lambda^\pi \left(\hat{\mathbf{b}}, \mathbf{y}, \mathbf{a} \right) \right] \tag{35}$$

$$\nabla_{\boldsymbol{\theta}_V} V_\lambda^\pi = \mathbb{E}_{\mathcal{M}}\left[w \nabla_{\boldsymbol{\theta}^V} V_\lambda^\pi \left(\hat{\mathbf{b}}, \mathbf{y} \mid \boldsymbol{\theta}^V \right) A_\lambda^\pi \left(\hat{\mathbf{b}}, \mathbf{y}, \mathbf{a} \right) \right] \tag{36}$$

where w is the importance sampling weight with sample distribution a policy $\boldsymbol{\mu}$ retrieved from the experience replay and target distribution the current policy. A_λ^π is the advantage function, which is herein approximated by the temporal difference:

$$A_\lambda^\pi \left(\hat{\mathbf{b}}, \mathbf{y}, \mathbf{a} \mid \boldsymbol{\theta}^V \right) \simeq \overline{c}_b + \sum_{m=1}^{M} \lambda_m g_{s,m} + \gamma V_\lambda^\pi \left(\hat{\mathbf{b}}', \mathbf{y}' \mid \boldsymbol{\theta}^V \right) - V_\lambda^\pi \left(\hat{\mathbf{b}}, \mathbf{y} \mid \boldsymbol{\theta}^V \right) \tag{37}$$

Algorithm 1 Constrained Deep Decentralized Multi-agent Actor Critic

Initialize replay buffer

Initialize actor, critic, and dual parameters $\left[\boldsymbol{\theta}_{\pi}^{(j)}\right]_{j=1}^{N_{CU}}, \boldsymbol{\theta}_{V}, \left[\lambda_m\right]_{m=1}^{M}$

for *number of episodes* **do**

  for $t=1,\ldots,T$ **do**

    Select action $\mathbf{a}_t$ at random according to exploration noise

    Otherwise select action $\mathbf{a}_t \sim \boldsymbol{\mu}_t = \left[\pi_j\left(\cdot \mid \hat{\mathbf{b}}_t, \mathbf{y}_t, \boldsymbol{\theta}_{\pi}^{(j)}\right)\right]_j^{N_{CU}}$

    Estimate costs $\overline{c}_{b,t} = \overline{c}_b$, $g_{s,mt} = g_{s,m}$ given $\hat{\mathbf{b}}_t$ and $\mathbf{a}_t$

    Observe $o_{t+1}^{(l)} \sim p\left(o_{t+1}^{(l)} \mid \mathbf{b}_t^{(l)}, \mathbf{y}_t, \mathbf{a}_t\right)$ for $l=1,2,\ldots,N_C$

    Compute beliefs $\mathbf{b}_{t+1}^{(l)}$ for $l=1,2,\ldots,N_C$

    Store tuple $\left(\hat{\mathbf{b}}_t, \mathbf{y}_t, \mathbf{a}_t, \boldsymbol{\mu}_t, \overline{c}_{b,t}, \left[g_{s,mt}\right]_{m=1}^{M}, \hat{\mathbf{b}}_{t+1}, \mathbf{y}_{t+1}\right)$ to replay buffer

  end for

  Sample batch $\left(\hat{\mathbf{b}}_i, \mathbf{y}_i, \mathbf{a}_i, \boldsymbol{\mu}_i, \overline{c}_{b,i}, \left[g_{s,mi}\right]_{m=1}^{M}, \hat{\mathbf{b}}_{i+1}, \mathbf{y}_{i+1}\right)$ from replay buffer

  If $\hat{\mathbf{b}}_i$ is terminal state $A_{\lambda,i}^{\pi} = \overline{c}_{b,i} + \sum_{m=1}^{M} \lambda_m g_{s,mi} - V_{\lambda}^{\pi}\left(\hat{\mathbf{b}}_i, \mathbf{y}_i \mid \boldsymbol{\theta}^{V}\right)$

  Otherwise $A_{\lambda,i}^{\pi} = \overline{c}_{b,i} + \sum_{m=1}^{M} \lambda_m g_{s,mi} + \gamma V_{\lambda}^{\pi}\left(\hat{\mathbf{b}}_{i+1}, \mathbf{y}_{i+1} \mid \boldsymbol{\theta}^{V}\right) - V_{\lambda}^{\pi}\left(\hat{\mathbf{b}}_i, \mathbf{y}_i \mid \boldsymbol{\theta}^{V}\right)$

  Update actor parameters $\boldsymbol{\theta}_{\pi}^{(j)}$ according to gradient:

$$\nabla_{\boldsymbol{\theta}_{\pi}^{(j)}} V_{\lambda}^{\pi} \simeq \sum_i w_i \left(\sum_{j=1}^{N_{CU}} \nabla_{\boldsymbol{\theta}_{\pi}^{(j)}} \log \pi_j\left(a_i^{(j)} \mid \hat{\mathbf{b}}_i, \mathbf{y}_i, \boldsymbol{\theta}_{\pi}^{(j)}\right) \right) A_{\lambda,i}^{\pi}$$

  Update critic parameters $\boldsymbol{\theta}^{V}$ according to gradient:

$$\nabla_{\boldsymbol{\theta}_V} V_{\lambda}^{\pi} \simeq \sum_i w_i \nabla_{\boldsymbol{\theta}^V} V_{\lambda}^{\pi}\left(\hat{\mathbf{b}}_i, \mathbf{y}_i \mid \boldsymbol{\theta}^{V}\right) A_{\lambda,i}^{\pi}$$

  Update dual variables λ_m, $m=1,\ldots,M$, based on current policy return, according to gradient:

$$\nabla_{\lambda_m} V_{\lambda}^{\pi} \simeq \sum_{t=0}^{T} \gamma^t g_{s,mt} - \alpha_{s,m}$$

end for

The gradient of dual variables λ_m is easily computed as (Tessler, et al., 2018):

$$\nabla_{\lambda_m} V_{\lambda}^{\pi} \simeq \sum_{t=0}^{T} \gamma^t g_{s,m} - \alpha_{s,m} \tag{38}$$

Dual variables are updated through on-policy samples, since off-policy weighted sampling of multiple time steps produces high-variance estimators that may trigger training instabilities. Algorithm 1 describes the aforementioned implementation steps.

RESULTS

Environment details

A stochastic, non-stationary, partially observable 10-component deteriorating system is considered, operating over a lifecycle period of 50 decision steps (years), with a discount factor of γ=0.975. For civil engineering systems, discount factors typically range from 0.93 to 0.98. Higher discount factors make the decision problem more challenging, in the sense that they increase the effective horizon of important

decisions. Links between components create the system shown in Figure 9. It is assumed that link operation depends solely on the operating status of the respective components. Overall system connectivity is determined by the connectivity of nodes A and B.

Each component has independent deterioration dynamics. These are expressed by 4x4x50 three-dimensional transition matrices, corresponding to 4 damage states (*intact*, *minor damage*, *major damage*, *severe damage*), combined with 50 deterioration rates, as many as the decision steps of the system lifecycle. Component transitions are given in Tables 7 and 8. Component transition parameters for the underlying hidden Markov models are assumed to be known or already learned, thus model uncertainty is not considered in this example. For learning of (hidden) Markov models and details on forming and maximizing the respective likelihood functions based on load-conditioned structural data, the interested reader can refer to (Andriotis & Papakonstantinou, 2018; Andriotis & Papakonstantinou, 2018), among various sources. In the case of latent states, as shown in the previous works, expectation-maximization or recurrent neural networks can be used. Parameter inference with hidden Markov models can be efficiently applied as in

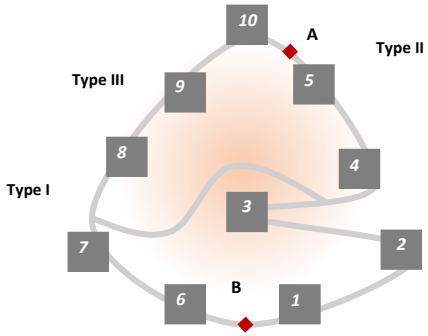


Figure 9. Multi-component deteriorating system. The system fails when connectivity between nodes A and B is lost. Major costs are incurred when the system fails. Minor costs are incurred for combinations of failed series subsystems. Types I-III refer to the severity of the deterioration model, from less to more severe, respectively.

(Papakonstantinou, et al., 2022; Amir, et al., 2021, In Print). Different failure probabilities are considered based on each one of the above damage states, as shown in Table 9. Thus, the system behavior as a whole, is described by the Bayesian network of Figure 10. The examined system has been kept application-agnostic, being however assigned general deteriorating characteristics that can, among others, resemble formations of transportation networks, where components 1-10 are deteriorating bridges controlling the functionality of the respective links (e.g., road segments), or parallel-series reliability block diagrams that can be applied to multi-member/unit structures such as a structural truss or a bridge-type diagram of an electrical circuit.

Table 7. Component initial damage state transition probabilities for deterioration model Types I, II, and III.

Deterioration Model	p_{12}	p_{13}	p_{14}	p_{23}	p_{24}	p_{34}
Type I	0.0129	0.0072	0.0008	0.0102	0.0038	0.0092
Type II	0.0311	0.0096	0.0014	0.0283	0.0057	0.0281
Type III	0.0428	0.0229	0.0033	0.0406	0.0095	0.0328

Table 8. Component final damage state transition probabilities for deterioration model Types I, II, and III.

Transition Probability	p_{12}	p_{13}	p_{14}	p_{23}	p_{24}	p_{34}
Type I	0.0618	0.0512	0.0036	0.0905	0.0091	0.0768
Type II	0.0862	0.0868	0.0051	0.1219	0.0121	0.1091
Type III	0.1347	0.0669	0.0098	0.1665	0.0244	0.1462

Table 9. Component failure probabilities for different deterioration types and damage states.

Damage State	*Intact*	*Minor*	*Major*	*Severe*
Type I	0. 0019	0. 0067	0. 0115	0. 0177
Type II	0. 0028	0. 0076	0. 0163	0. 0219
Type III	0. 0088	0. 0210	0. 0449	0. 0564

Further details on consistently coupling inference of dynamic Bayesian networks, both in the state and parameter space, with POMDPs for deteriorating structures, can be found in (Morato, et al., 2022; Morato, et al., 2019), whereas formulations without parametric updates also exist in (Morato, et al., 2022). The final state vector for each component is $s^{(i)}=(x^{(i)},\tau^{(i)},f^{(i)},t)$, where $x^{(i)}$ is the damage state, $\tau^{(i)}$ is the deterioration rate, $f^{(i)}$ is a binary failure indicator, and t is the decision time step (t is the same for all components). Vectors $s^{(i)}$ define the input space of the neural networks, thus naturally instilling non-stationarity in the learned policy. Failure is considered an absorbing state. Hence, we assume that when a component fails it remains failed at the next step, as long as no restorative action is taken. This allows us to augment the component state space, finally obtaining 5x5x50 transition matrices.

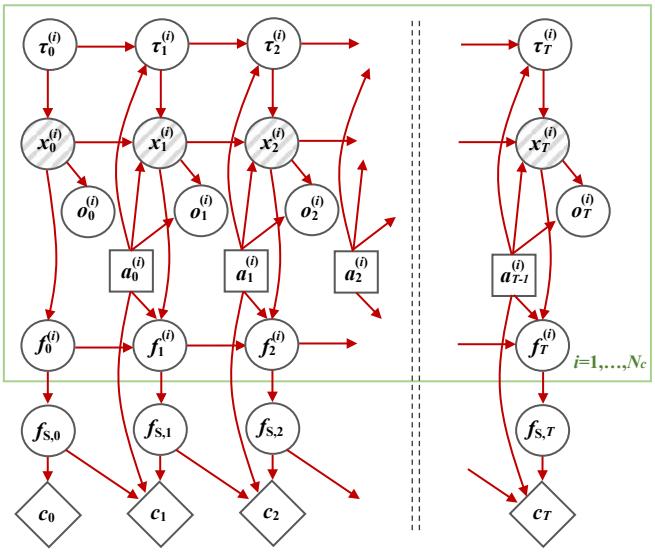


Figure 10. Dynamic Bayesian network of multi-component deteriorating system in time.

We consider three types of available maintenance actions; A_M ={*no-repair*, *partial-repair*, *restoration/replacement*}. There are also two types of available inspection actions; A_I ={*no-inspection*,

inspection}. Accordingly, to allow for utmost diversification between component policies, each component, which herein defines a separate control unit, is assigned five available inspection and maintenance actions, based on the combinations of the above-mentioned sets, i.e., $a^{(i)} \in A_M \times A_I \setminus$ (*restoration/replacement*, *inspection*). The (*restoration/replacement*, *inspection*) action is excluded from the set of available actions, as it is assumed that whenever a system component is replaced, thus returning to an as-good-as-new condition, a decision for inspection is strictly suboptimal. No-repair costs are null, whereas restoration/replacement costs are the same for all components. Partial-repair costs are 7.5%, 10%, 15% of the component replacement cost, for component Types I, III, and II, respectively. Inspection costs are the same for all components, at 1.5% of the component replacement cost. Partial-repairs send components one damage state back without changing the deterioration rate, restorations/replacements send components to the initial damage state and deterioration rate; whereas no-repairs have no effect on the damage state and deterioration rate. Partial-repairs have no effect on failed components and are considered to have been completed before the next environment transition. When restorations/replacements are chosen, these are completed at the end of the next time step, negating the deterioration transition during that step. Thus, in this case, the next state is the intact one with certainty.

If an inspection action is taken, observation probabilities are given by the following observation matrices:

$$\left[\Pr\left(o^{(i)} \mid s^{(i)}, a^{(i)} \in A_M \times \{inspection\}\right)\right]_{\substack{o^{(i)} \in \Omega \\ s^{(i)} \in S}} = \begin{bmatrix} 0.84 & 0.13 & 0.02 & 0.01 & \\ 0.11 & 0.77 & 0.09 & 0.03 & \\ 0.02 & 0.16 & 0.70 & 0.12 & \\ 0.01 & 0.02 & 0.13 & 0.84 & \\ & & & & 1 \end{bmatrix} \tag{39}$$

Observation matrices depend on state discretization and presumed measurement noise or estimated model errors (Madanat, 1993). Failure is considered to be a self-announcing event, hence, component (5,5) of the observation matrix of Eq. (39) is 1. Accordingly, if no inspection is taken, the observation matrix reads:

$$\left[\Pr\left(o^{(i)} \mid s^{(i)}, a^{(i)} \in A_M \times \{no-inspection\}\right)\right]_{\substack{o^{(i)} \in \Omega \\ s^{(i)} \in S}} = \begin{bmatrix} 1 & 1 & 1 & 1 & \\ & & & & 1 \end{bmatrix}^T \tag{40}$$

System failure, i.e., loss of connectivity between nodes A and B, is described by random variable f_s. Random variable f_s assumes four values associated with events E_0: all links available, E_1: 25% of links failed, E_2: 50% of links failed without system failure, and F_s: system failure. A link is failed if at least one component is failed. We can thus consider the series subsystems, controlling the link failures, l_1={1,2,3}, l_2={4,5}, l_3={6,7}, and l_4={8,9,10}. Their failure events are accordingly described by events $F_{l,1}$, $F_{l,2}$, $F_{l,3}$, and $F_{l,4}$. Based on the above, it can be derived that the system failure probability is:

$$\Pr(F_s) = \Pr(F_{l,1})\Pr(F_{l,3}) + \Pr(F_{l,2})\Pr(F_{l,4}) - \prod_{i=1}^{4} \Pr(F_{l,i}) \tag{41}$$

The corresponding non-failure events of interest, E_0, E_1, E_2, are defined as:

$$E_0 : \bigcap_{i=1}^{4} F_{l,i}^{-}; \qquad E_1 : \bigcup_{i=1}^{4}\left(F_{l,i} \bigcap_{j\neq i}^{4} F_{l,j}^{-} \right); \qquad E_2 : \left(\bigcup_{i,j=1,i>j}^{4} F_{l,i} \cap F_{l,j} \right) \cap F_s^{-} \tag{42}$$

Accordingly, the probabilities of events E_0, E_1, E_2 are computed as:

$$\begin{aligned} \Pr(E_0) &= \prod_{i=1}^{4} \left(1 - \Pr(F_{l,i})\right) \\ \Pr(E_1) &= \sum_{i=1}^{4} \prod_{j=1, j\neq i}^{4} \left(1 - \Pr(F_{l,j})\right) \Pr(F_{l,i}) \\ \Pr(E_2) &= 1 - \Pr(E_0) - \Pr(E_1) - \Pr(F_s) \end{aligned} \tag{43}$$

Accruable and instantaneous losses due to failure are equivalent to 2.5 and 50 times the system rebuild cost, respectively, i.e., $c_{F_s}^{acc} = 2.5 \cdot c_{reb}$ and $c_{F_s}^{inst} = 50 \cdot c_{reb}$. Similarly, we consider accruable and instantaneous losses incurred when 25% and 50% of system links are not available (i.e., at least one of their respective components is at the failure state). These losses are incurred if events E_1, E_2 occur, respectively, and are quantified in cost units as $c_{E_1}^{acc} = 0.05 \cdot c_{reb}$, $c_{E_2}^{acc} = 0.25 \cdot c_{reb}$, $c_{E_1}^{inst} = 1 \cdot c_{reb}$, $c_{E_2}^{inst} = 5 \cdot c_{reb}$. In the case of transportation networks, for example, such accruable losses may refer to time delays and/or additional user costs due to detours, whereas such instantaneous losses may pertain to capital loss due to asset failures related to those links.

Based on the above losses, the fact that system events are fully observable, and following the risk definition of Eq. (16), the interval risk reads:

$$c_{b,D} = \sum_{\substack{f_s \in \\ \{F_s, E_2, E_1\}}} \Pr(f_{s,t+1}) \left(c_{f_{s,t+1}}^{acc} + \left(1 - \Pr(f_{s,t})\right) c_{f_{s,t+1}}^{inst} \right) \tag{44}$$

Apart from the above losses, additional costs are included in the analysis, pertaining to scheduled system shutdowns. Those come as a result of different action combinations on different system components. That is, considering that non-trivial maintenance actions require some degree of component non-operability for completion during a time step, events Ea_0, Ea_1, Ea_2, and Fa_s can occur, in analogy to events E_0, E_1, E_2, and F_s. Those losses are only incurred if the system would be otherwise in an operating condition (i.e., not failed). Events and their probabilities are similarly defined as in Eqs. (42)-(44), whereas respective costs are the same as the accruable losses due to events E_0, E_1, E_2, and F_s.

Experimental setup

For the purposes of this numerical investigation, two sets of analyses are conducted. The first set considers a budget cycle period of T_B = 5. Each budget period shares the same budget cap, and 9 different levels of budget constraints are considered, which are given as functions of the system rebuild cost, {5, 7.5, 10, 12.5, 15, 17.5, 20, 25, 30}% c_{reb}. For the second set of analyses, 9 different levels of lifecycle risk constraints are considered, i.e., {1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3.25} c_{reb}. In addition to the above analyses, the unconstrained policy is also learned.

For training, the Keras deep learning python libraries are used with Tensorflow backend. For all analyses, the actor networks consist of two fully connected hidden layers with 50 Rectified Linear Unit

activation functions each, for all 10 components. No parameters are shared among component actors, and each control unit has a 5-dimensional softmax output corresponding to the cardinality of $A_M \times A_I$ \ (*restoration/replacement*, *inspection*). The critic network also consists of two fully connected hidden layers with 150 ReLU activations each. The critic has a one-dimensional linear output, which approximates the POMDP Lagrangian value function of the entire system.

The Adam optimizer (Kingma & Ba, 2014) is utilized for stochastic gradient descent on the networks parameter space, with learning rates being gradually adjusted from 1E-3 and 1E-4 to 1E-4 and 1E-5 for the critic and actor, respectively. The learning rate of Lagrange multipliers is set to 1E-5. The size of the experience replay is set equal to 300,000 and an exploration noise linearly annealed from 100% to 1% is added at the first 2,500 episodes of the training process.

The main factors influencing the computational cost are the sizes of the actor and critic networks, and the sample complexity of the learning scheme, which dictates the number of simulator calls. The depth and width of the hidden layers grow with the dimensions of state and action spaces (inputs and outputs, respectively), and the user is generally advised to decide about network size and hyperparameters on a problem-to-problem basis.

All analyses were run on an Intel Xeon Platinum 8260 CPU at 2.40GHz. DRL solutions required approximately 4 days to exceed the best risk-based baseline, presented in the next section. This time is comparable to the computational cost associated with obtaining the optimal parameters of this baseline through standard brute-force evaluation of possible policies.

DRL solutions and baseline policies

To verify the quality of DDMAC solutions, we construct and optimize various baseline policies, incorporating well-established condition- risk-, and time-based inspection and maintenance assumptions, which are also combined with periodic action considerations, as well as component prioritization approaches. These baselines are:

- Fail Replacement (FR) policy. No inspections are taken. If a component fails, it is immediately replaced. No variable is optimized.
- Age-Periodic Maintenance (APM) policy. No inspections are taken, whereas maintenance actions are taken based on the age of components. Two maintenance ages are optimized: periodic age for component partial-repair and periodic age for component restoration/replacement.
- Age-Periodic Inspections and Condition-Based Maintenance (API-CBM) policy. Age-based inspections are taken for all components, based on each component's age. At inspection times, maintenance actions are taken based on the observed damage state of each component. Five variables are optimized: age interval for component inspection, and type of maintenance for each of the four observed damage states.
- Time-Periodic Inspections and Condition-Based Maintenance (TPI-CBM) policy. Time-based inspections are taken for all components at fixed intervals of the planning horizon. At inspection times, maintenance actions are taken based on the observed damage state of each component. Five variables are optimized: time interval for block component inspection, and type of maintenance for each of the four observed damage states.
- Risk-Based Inspections and Condition-Based Maintenance (RBI-CBM) policy. Inspections are taken for all components each time the system exceeds a predefined failure probability threshold. At inspection times, maintenance actions are taken based on the observed damage state of each component. Five variables are optimized: failure probability threshold, and type of maintenance for each of the four observed damage states.

The last two baseline policies are also optimized with Component Prioritization (CP), which produces policies RBI-CBM-CP and TPI-CBM-CP. Components are prioritized based on their probability of failure.

In this case, one extra decision variable regarding the number of components (1 to 10) to inspect and maintain is added. This policy adapts a heuristic presented in (Luque & Straub, 2019). In all baselines, if a component fails, it is immediately replaced. Such decision rules can be optimized by evaluation of possible policies through simulations, based on an underlying Bayesian network, or an actual physics-based model, or a meta-model fitted on data (Straub & Faber, 2005; Luque & Straub, 2019; Colone, et al., 2019).

In Figure 11, a comparison of the learned DDMAC policy with the various baselines is presented, for the unconstrained environment (total costs and disaggregated costs in linear and log scales, respectively). The best optimal baseline is the policy combining risk-based inspections, condition-based maintenance and component prioritization. It can be observed that the lifecycle cost attained by the best baseline is about 42% worse than the DDMAC solution. The optimal age-periodic maintenance and fail-replacement policies do not include the possibility of inspections and achieve the worst life- cycle costs. It is overall observed that baselines including inspections achieve consistently better results. Adding to this remark, it is interesting to note that the DDMAC policy spends more for inspections, i.e., performs a higher number of inspections, compared to the two best optimal baselines. As discussed, these inspections are in principle non-periodic and, as shown in Section 2.4, are driven by the innate notion of VoI in POMDPs. This allows the agents to make more informed decisions regarding proper maintenance actions that, overall, minimize the total cumulative costs of Eq. (19) more efficiently. Risk is significantly reduced with the DDMAC policy, as also indicated in Figure 11, whereas scheduled system shutdown costs are more intelligently avoided compared to other baselines, due to the flexibility in intervention timings and action combinations.

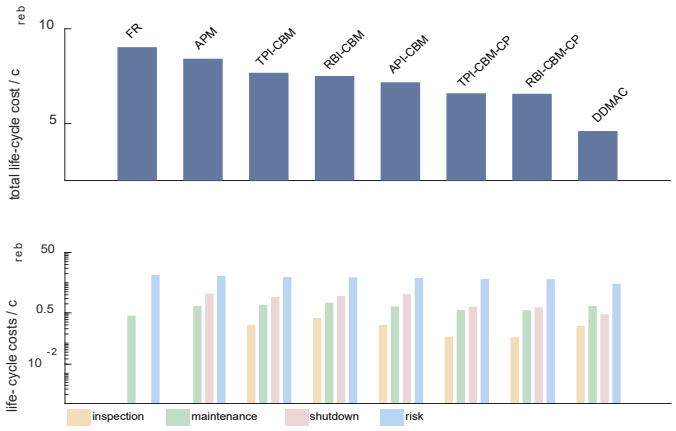


Figure 11. Comparison of DDMAC lifecycle policies with different baseline policies. Total lifecycle cost and lifecycle costs due to inspection, maintenance, shutdown, and risk (95% confidence intervals are lower than ±1%). The best optimized baseline is 42% worse than the DDMAC policy.

Constrained system solutions

Constrained DDMAC results for lifecycle inspection costs, maintenance costs, shutdown costs, and risk for different 5-year constraint levels are shown in Figure 12 (all costs in log scale). As expected, higher budget limits result in lower total lifecycle costs. Budget limits higher than 25% of the system rebuild cost, c_{reb}, practically converge to the unconstrained solution. A noticeable feature of the learned near-optimal policies is that as the budget becomes tighter, the agents tend to reduce their inspection expenses, to save resources in case of a need for major interventions (e.g., restoration/replacement actions). This means that they deliberately choose to forfeit better system information, in order to be more effective against disruption. It is characteristic that inspections are overall reduced in the budget cases below 15% c_{reb}, compared to the

cases above that budget threshold, since the component replacement cost is 10% c_{reb}. That is, below 10% c_{reb} budget constraints, restorations/replacements are infeasible. In Figure 13, the respective results for risk constraints are shown (all costs in log scale). It can be observed that as the decision-making task becomes more risk averse, the total lifecycle cost becomes higher, since more frequent inspection and maintenance actions need to be taken. Constrained solutions practically converge to the unconstrained one after the risk tolerance threshold of $2.75c_{reb}$. It is interesting to note here that for lower risk constraints (i.e., for scenarios where the agents need to keep total risk lower over the operating life), although the maintenance cost increases, the inspection cost is not following the same trend, hence, the inspection per maintenance cost ratio of the optimal policy consistently decreases. This is attributed to the fact that more frequent maintenance is unavoidable in a case where risks have to be kept low; something that, by itself, leads on average to longer sojourn in states of lower damage. As such, increased frequency of inspections, which would solely serve better state determination, is not favored by the agents, and thus lifecycle inspection costs do not present important changes for different risk-based constraints. Accordingly, due to the high demand for maintenance actions, scheduled shutdown costs also increase in low-risk cases.

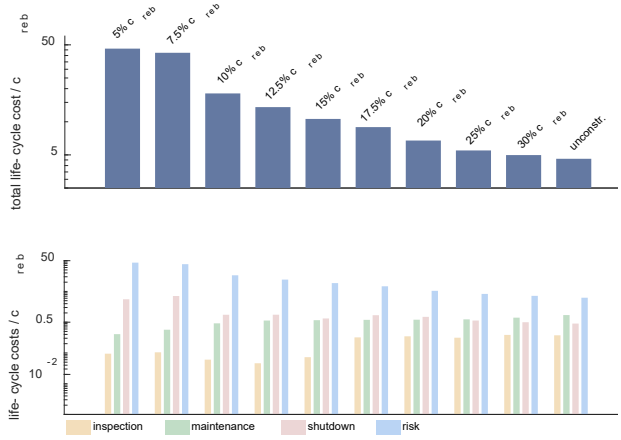


Figure 12. Comparison of DDMAC lifecycle policies for different 5-year constraints from 5% c_{reb} to infinity. Total lifecycle cost and lifecycle costs due to inspection, maintenance, shutdown, and risk (95% confidence intervals are lower than ±0.5%).

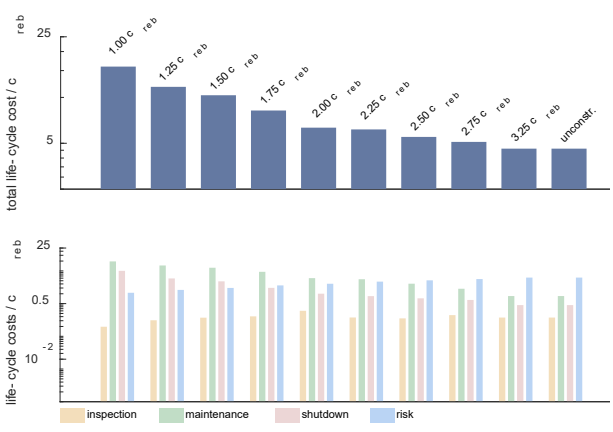


Figure 13. Comparison of DDMAC lifecycle policies for different life-cycle risk constraints from 1 c_{reb} to infinity. Total lifecycle cost and lifecycle costs due to inspection, maintenance, shutdown, and risk (95% confidence intervals are lower than ±0.5%).

In Figure 14, action frequencies and respective cost metrics of inspection and maintenance are depicted for two budget constraints corresponding to a low and a high budget scenario (i.e., to 15% and 20% c_{reb} 5-year budget constraints, respectively). Contour plots depict the frequency of maintenance and inspection actions per time unit. Adjacent graphs on the right show the mean step cost per component related to the respective action type, whereas the bottom graphs show the action cost per step, collectively for all system components. The same features are depicted for risk constraints of 2.75 and 3.25 c_{reb} in Figure 15. Examining Figures 12 and 14 together, we can observe that lowering the budget from 20% to 15% c_{reb} has significant consequences for risk, which increases disproportionally with the achieved reduction in the expected total lifecycle maintenance cost. What changes significantly for maintenance cost, as shown in Figure 14, is its distribution per time unit and component, rather than its total lifecycle value. This is indicative of the general observation that stricter budgets increase risk, without necessarily yielding clear

economic budget-related benefits, if any, in the long run. Another interesting feature is that, in the presence of stricter budgets, the imbalance in the allocation of maintenance resources among components increases. Inspections and their respective expenditures are considerably restricted, as mentioned previously. As also shown in Figure 14, for the 15% c_{reb} case, inspections are rather reserved mainly for component 4, as this is the most vulnerable component of path 6,7,4,5, which is the path securing system survival with the least number of components.

For the cases of risk-based constraints, examining Figures 13 and 15 together, we can observe that relevant costs are distributed more evenly in time over the planning horizon. Over the system lifecycle, we observe that lowering the risk tolerance considerably encumbers maintenance costs per step and in total. Similarly, to the budget-constrained cases, for the 2.75 c_{reb} versus 3.25 c_{reb} risk constraint case, inspections are prominently clustered to fewer components. Accordingly, it is observed that the agents reserve their inspection actions exclusively for components 3-5,7,8,10. This intrinsically prioritized selection of components to be frequently inspected allows the agents to track the state of at least half of the components from each link, and thereby to better synchronize maintenance actions in order to minimize system shutdowns and costs. It was observed that although mathematically feasible from an optimization perspective, policies below 2.0 c_{reb} start becoming practically unrealistic due to the very frequent restorations/replacements that need to be taken in order for the risk constraints to be satisfied.

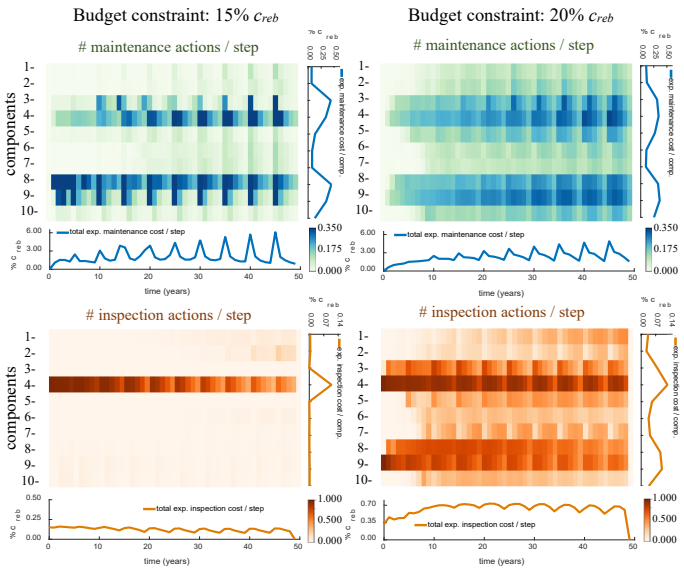


Figure 14. Components maintenance and inspection frequency per step and respective mean costs for 5-year budget constraints of 15% and 20% c_{reb} (95% confidence intervals are lower than ±0.5%).

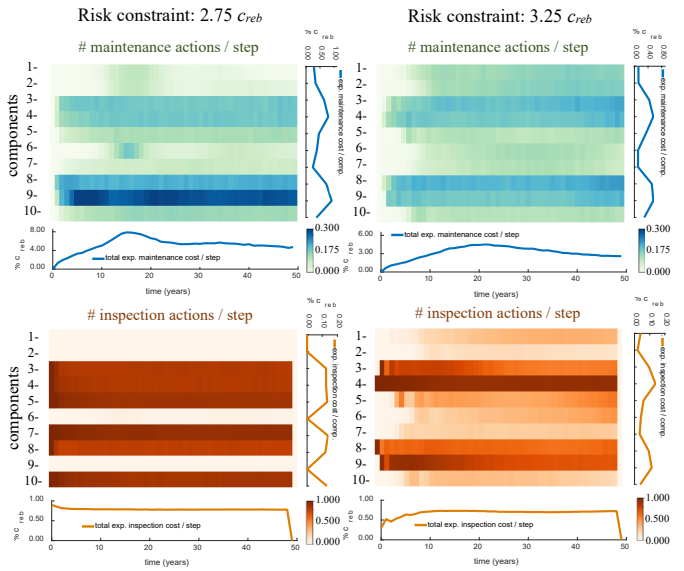


Figure 15. Components maintenance and inspection frequency per step and respective mean costs for risk constraints of 2.75 and 3.25 creb (95% confidence intervals are lower than ±0.5%).

To look closer into how policies change for different constraints, four detailed policy realizations are shown in Figures 16 and 17, for the constrained environments shown in Figures 14 and 15, respectively. In Figure 16(a), displaying the realization of component failure probabilities and respective inspection and maintenance actions, for two cases of 5-year budget constraints, it can be readily observed that, in the low-budget scenario, available budgetary resources are primarily allotted to the maintenance needs of components 3,4,8, and 9. This is explained by the fact that these are Type III components, thus being described by the most aggressive deterioration. In this realization example, only component 4 is inspected, since, as also explained earlier, with a budget limit close to the component replacement cost, the agents choose to inspect more rarely in order to save resources in case major interventions are needed. In the high-

budget scenario, inspections play a more prominent role, since the imposed budget restrictions have become looser, and the agents have the budgetary capacity to afford expenditure for acquiring information. Although Type III components continue to receive the majority of maintenance actions, intervention resources are now allotted more frequently to all components. Some of the most prominent intervention effects significantly changing the overall system failure probability are indicated in Figure 16(b). The various costs are also tracked in Figure 16(c). For the 20% c_{reb} case, a notable feature can be observed for components 6 and 7, controlling the operability of the third link. Component 7 fails at t=38 and available resources do not allow for immediate replacement, which is postponed to t=40, when the next budget cycle begins. In the meanwhile, the agent of component 6 takes advantage of the link shutdown and applies repeated opportunistic partial repairs, which do not yield additional shutdown costs. Overall, it can be interestingly observed in Figures 14 and 16, that the agents, despite their decentralized policies, form and increase collaboration as the budget becomes lower, directing their focus to components that are more vulnerable to deterioration, or more strategically placed in terms of system connectivity.

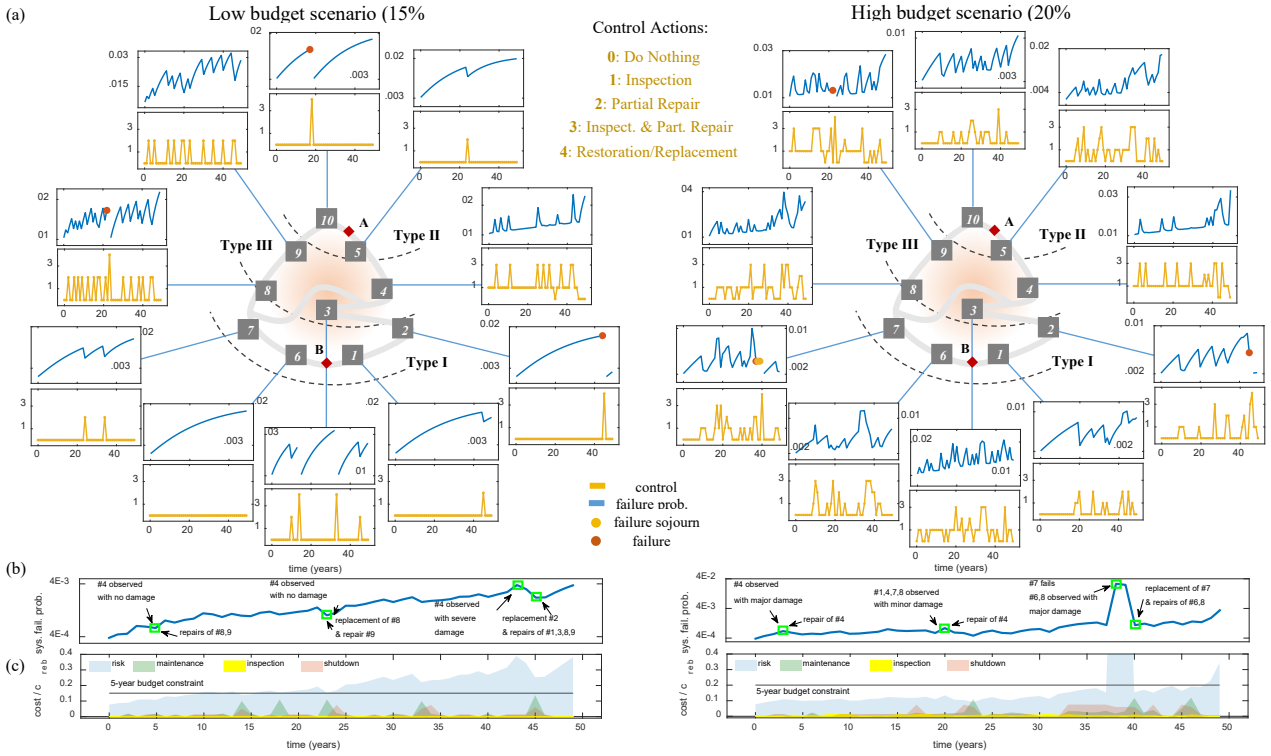


Figure 16. Lifecycle realization of the DDMAC policy for 15% creb and 20% creb 5-year budget constraints: (a) component failure probabilities and actions; (b) system failure with selected interventions; (c) costs of inspection and maintenance actions, scheduled shutdowns, and risks.

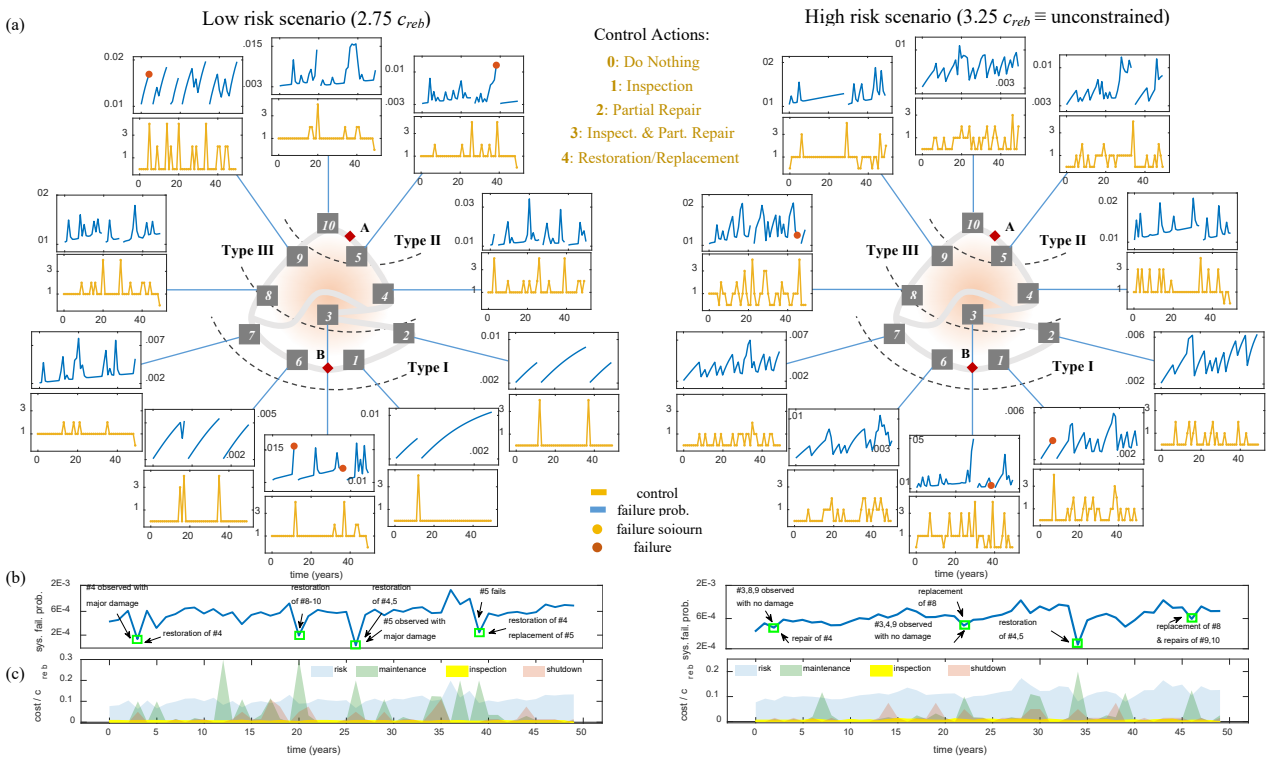


Figure 17. Lifecycle realization of the DDMAC policy for 15% creb and 20% creb 5-year budget constraints: (a) component failure probabilities and actions; (b) system failure with selected interventions; (c) costs of inspection and maintenance actions, scheduled shutdowns, and risks.

Similar features can be seen for the low- and high-risk constraints cases of Figure 17. In the 3.25 c_{reb} case, effectively coinciding with the unconstrained policy, a complex and diverse policy is overall illustrated. It is worth noting that, in the absence of any budget constraints, inspections are now taken frequently for all components, whereas restoration/replacement actions start to also have more prominent preventive characteristics (i.e., they are not only reserved for failure events). This is even more apparent in the low-risk scenario, in which case restorations need to be performed in a more recurrent fashion to ensure low probability of failure. In turn, this also causes more system closures and thus increases shutdown costs. To balance this side effect of frequent restorative actions, the agents are interestingly shown to deploy a block-restoration/replacement logic in their policies. That is, as shown in the 2.75 c_{reb} scenario of Figure 17(a), component agents of the same links synchronize their restoration actions (e.g. components 2,3 at t=37; components 8-10 at t=20; components 4,5 at t=26), whereas they also start to extensively leverage opportunistic interventions in links where failure events occur (e.g., components 1-3 at t=12; components 4,5 at t=39). The system failure probability and the various costs along with various actions that affect them are shown in Figures 17(b),(c), respectively.

The mean failure interval probability of the system over time is shown in Figure 18, for various 5-year budget and various lifecycle risk constraints. It is observed that, on average, system failure probability reaches its peak before the onset of new budget cycles. For the unconstrained case, mean failure probability is allowed to increase over time, without abrupt escalations, since no budget limitation is imposed. The 7.5% c_{reb} constrained case reflects an extreme lifecycle optimization setting where no replacement actions are feasible. Thus, in this case no major corrective steps are detected in the evolution of the mean failure probability. In the case of risk constraints, the more stringent the risk constraint is, the higher is the reliability of the system at each time step, as anticipated.

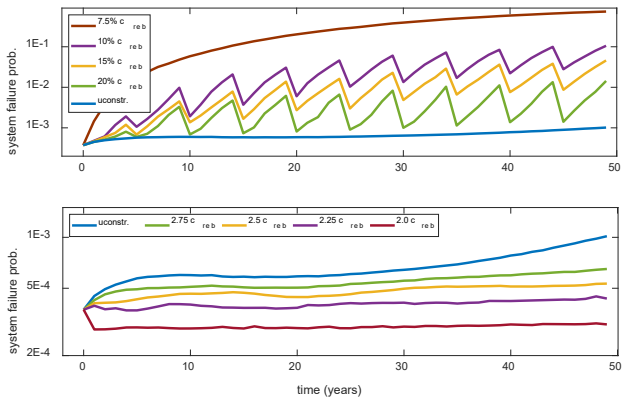


Figure 1. Lifecycle realization of the DDMAC policy for 15% creb and 20% creb 5-year budget constraints: (a) component failure probabilities and actions; (b) system failure with selected interventions; (c) costs of inspection and maintenance actions, scheduled shutdowns, and risks.

Overall, Figures 14-18 allow us to obtain insights in the ways the agents reason and adapt under a certain deteriorating environment, by forming and altering cooperative strategies, or dynamically re-prioritizing inspection and maintenance resources based on different risk and resource constraints. Such analyses are useful in order to interpret patterns in the learned policies and can be utilized to also enhance more traditional decision rules, bridging optimality gaps induced by their lifecycle planning assumptions and formulations.

CONCLUSION

In this work, a stochastic optimal control framework for inspection and maintenance planning of deteriorating systems operating under incomplete information and constraints is developed. Decision-making is cast in a multi-agent decentralized framework of DRL and POMDPs, where each system component, or control unit consisting of multiple components, acts as an independent agent given the dynamically updated global system state probabilistic information. While satisfying a shared overarching objective, each agent can make its own inspections and maintenance choices. Operational resource-based restrictions and policy risk considerations are taken into account by means of relevant stochastic soft and/or hard constraints. The latter are incorporated in the solution scheme through state augmentation, thus being rendered as environment properties, whereas the former are appended in the lifecycle objective function as dual variables, to form the Lagrangian function to be optimized. Modeling of various constraint choices is discussed, whereas a thorough numerical investigation is provided for budget and risk constraints, which are of particular significance in infrastructure management applications. Along these lines, a broad risk definition is also presented and utilized in the constrained optimization procedure, accommodating both the instantaneous and accruable nature of damage-related losses. This risk definition is further shown to be reducible to classic reliability-based definitions. Solutions to the optimization problem are driven by the introduced DDMAC algorithm. DDMAC uses both policy and value function parametrizations, experience replay, off-policy network parameter updating, and operates on the belief space of the underlying POMDP.

Operation under constraints is shown to considerably affect how the agents adapt their policies. The conducted parametric analysis shows that:

- The need for inspections fades in low-budget environments, where the agents tend to diminish expenses otherwise allotted to system information updating needs, in order to secure advanced intervention capabilities through availability of maintenance resources.

- Stricter budget constraints reduce inspection and maintenance costs for the respective budget cycle, however, without comparably reducing these costs in the long run, i.e., cumulatively, over the system lifecycle.
- In risk-averse environments, inspection costs do not follow the notable increase in maintenance costs, which are necessary in order to maintain low-risk levels over the system operating life.
- In such cases, agents are shown to increasingly leverage the structural properties of the system or incidental subsystem failure configurations, to develop opportunistic repair strategies, so that system operability is minimally disrupted.
- Budget limitations and risk intolerance disproportionally increase the risk and maintenance costs, respectively, compared to the reductions they achieve in the constrained quantities.
- For both types of constraints, multi-agent cooperation emerges more prevalent as restrictions become stricter, since resource scarcity and risk intolerance force the agents to more carefully reallocate resources and redefine management priorities, based on the specific deterioration dynamics and structural importance of different system parts. This rescheduling arises naturally and intrinsically through the training process, without any explicit user-based enforcement or penalty-driven motivation.

REFERENCES

Achiam, J., Held, D., Tamar, A. & Abbeel, P., 2017. *Constrained policy optimization.* Sydney, Australia, 34th International Conference on Machine Learning.

Altman, E., 1999. *Constrained Markov decision processes.* Sophia Antipolis, France: CRC Press.

Amir, M., Papakonstantinou, K. G. & Warn, G. P., 2021, In Print. Scaled Spherical Simplex Filter and state-space damage-plasticity finite element model for computationally efficient system identification. *Journal of Engineering Mechanics.*

Andriotis, C. P. & Papakonstantinou, K. G., 2018. Extended and generalized fragility functions. *Journal of Engineering Mechanics,* 144(9), p. 04018087.

Andriotis, C. P. & Papakonstantinou, K. G., 2018. *Probabilistic structural performance assessment in hidden damage spaces.* Paros, Greece, Proceedings of Computational Stochastic Mechanics Conference (CSM).

Andriotis, C. P. & Papakonstantinou, K. G., 2019. Managing engineering systems with large state and action spaces through deep reinforcement learning. *Reliability Engineering & System Safety,* Volume 191, p. 106483.

Andriotis, C. P. & Papakonstantinou, K. G., 2019b. *Life-cycle policies for large engineering systems under complete and partial observability.* Seoul, South Korea, 13th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP13).

Andriotis, C. P. & Papakonstantinou, K. G., 2021. Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints. *Reliability Engineering & System Safety,* Volume 212, p. 107551.

Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., & Mordatch, I., 2019. Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528.*

Bellman, R., 1956. Dynamic programming and Lagrange multipliers. *Proceedings of the National Academy of Sciences of the United States of America,* 42(10), p. 767.

Bellman, R. E., 1957. *Dynamic programming.* Mineola, NY: Princeton University Press.

Bernstein, D. S., Givan, R., Immerman, N. & Zilberstein, S., 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research,* 27(4), pp. 819-40.

Bertsekas, D., 1999. *Nonlinear Programming.* Belmont, MA: Athena Scientific.

Bertsekas, D., 2005. *Dynamic programming and optimal control.* Belmont, MA: Athena Scientific.

Bismut, E. & Straub, D., 2018. *Inspection and maintenance planning in large monitored structures.* Singapore, 6th International Symposium on Reliability and Engineering Risk Management (ISRERM).

Bocchini, P. & Frangopol, D. M., 2011. A probabilistic computational framework for bridge network optimal maintenance scheduling. *Reliability Engineering & System Safety,* 96(2), pp. 332-49.

Bocchini, P. & Frangopol, D. M., 2012. Optimal resilience-and cost-based postdisaster intervention prioritization for bridges along a highway segment. *Journal of Bridge Engineering,* 17(1), pp. 117-29.

Castanier, B., Bérenguer, C. & Grall, A., 2003. A sequential condition-based repair/replacement policy with non-periodic inspections for a system subject to continuous wear. *Applied Stochastic Models in Business and Industry,* 19(4), pp. 327-347.

Chow, Y., Ghavamzadeh, M., Janson, L. & Pavone, M., 2017. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research,* 18(1), pp. 6070-120.

Colone, L., Dimitrov, N. & Straub, D., 2019. Predictive repair scheduling of wind turbine drive-train components based on machine learning. *Wind Energy,* 22(9), pp. 1230-42.

Degris, T., White, M. & Sutton, R. S., 2012. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839.*

Di Castro, D., Tamar, A. & Mannor, S., 2012. Policy gradients with variance related risk criteria. *arXiv preprint arXiv:1206.6404.*

Ellis, H., Jiang, M. & Corotis, R. B., 1995. Inspection, maintenance, and repair with partial observability. *Journal of Infrastructure Systems,* 1(2), pp. 92-99.

Faber, M. H. & Stewart, M. G., 2003. Risk assessment for civil engineering facilities: critical overview and discussion. *Reliability Engineering & System Safety,* 80(2), pp. 173-84.

Frangopol, D. M., Kallen, M. J. & Noortwijk, J. M. V., 2004. Probabilistic models for life-cycle performance of deteriorating structures: review and future directions. *Progress in Structural Engineering and Materials,* 6(4), pp. 197-212.

Frangopol, D. M., Lin, K. Y. & Estes, A. C., 1997. Life-cycle cost design of deteriorating structures. *Journal of Structural Engineering,* 123(10), pp. 1390-401.

Garcıa, J. & Fernández, F., 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research,* 16(1), pp. 1437-80.

González, A. D., Dueñas-Osorio, L., Sánchez-Silva, M. & Medaglia, A. L., 2016. The interdependent network design problem for optimal infrastructure system restoration. *Computer-Aided Civil and Infrastructure Engineering,* 31(5), pp. 334-50.

Goulet, J. A., Der Kiureghian, A. & Li, B., 2015. Pre-posterior optimization of sequence of measurement and intervention actions under structural reliability constraint. *Structural Safety,* Volume 52, pp. 1-9.

Grall, A., Bérenguer, C. & Dieulle, L., 2002. A condition-based maintenance policy for stochastically deteriorating systems. *Reliability Engineering & System Safety,* 76(2), pp. 167-180.

Grall, A., Dieulle, L., Berenguer, C. & Roussignol, M., 2002. Continuous-time predictive-maintenance scheduling for a deteriorating system. *IEEE Transactions on Reliability,* 51(2), pp. 141-150.

Gupta, J. K., Egorov, M. & Kochenderfer, M., 2017. *Cooperative multi-agent control using deep reinforcement learning.* Sao Paulo, Brazil, International Conference on Autonomous Agents and Multiagent Systems, pp. 66-83.

Hernandez-Leal, P., Kartal, B. & Taylor, M. E., 2019. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems,* 33(6), pp. 750-97.

Isom, J. D., Meyn, S. P. & Braatz, R. D., 2008. *Piecewise linear dynamic programming for constrained POMDPs.* Chicago, IL, Association for the Advancement of Artificial Intelligence (AAAI) Conference, pp. 291-296.

Kaelbling, L. P., Littman, M. L. & Cassandra, A. R., 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence,* 101(1), pp. 99-134.

Kim, D., Lee, J., Kim, K. E. & Poupart, P., 2011. *Point-based value iteration for constrained POMDPs.* Barcelona, Spain, 22nd International Joint Conference on Artificial Intelligence (IJCAI).

Kingma, D. P. & Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Liu, Y., Chen, Y. & Jiang, T., 2020. Dynamic selective maintenance optimization for multi-state systems over a finite horizon: A deep reinforcement learning approach. *European Journal of Operational Research,* 283(1), pp. 166-81.

Luque, J. & Straub, D., 2019. Risk-based optimal inspection strategies for structural systems using dynamic Bayesian networks. *Structural Safety,* Volume 76, pp. 60-80.

Madanat, S., 1993. Optimal infrastructure management decisions under uncertainty. *Transportation Research Part C: Emerging Technologies,* 1(1), pp. 77-88.

Marseguerra, M., Zio, E. & Podofillini, L., 2002. Condition-based maintenance optimization by means of genetic algorithms and Monte Carlo simulation. *Reliability Engineering & System Safety,* 77(2), pp. 151-65.

Memarzadeh, M. & Pozzi, M., 2015. Integrated inspection scheduling and maintenance planning for infrastructure systems. *Computer-Aided Civil and Infrastructure Engineering,* 31(6), pp. 403-415.

Memarzadeh, M., Pozzi, M. & Kolter, J., 2016. Hierarchical modeling of systems with similar components: A framework for adaptive monitoring and control. *Reliability Engineering & System Safety,* Volume 153, pp. 159-69.

Morato, P. G., Nielsen, J. S., Mai, A. Q. & Rigo, P., 2019. *POMDP based maintenance optimization of offshore wind substructures including monitoring.* Seoul, South Korea, 13th International Conference on Applications of Staitstics and Probability in Civil Engineering (ICASP13).

Morato, P. G., Papakonstantinou, K. G., Andriotis, C. P., Nielsen, J. S., & Rigo, P. 2022. Optimal inspection and maintenance planning for deteriorating structural components through dynamic Bayesian networks and Markov decision processes. *Structural Safety*, 94: 102140.

Nicolai, R. P. & Dekker, R., 2008. Optimal maintenance of multi-component systems: A review. In: *Complex System Maintenance Handbook.* Springer, London, pp. 263-286.

Nozhati, S., Sarkale, Y., Chong, E. K. & Ellingwood, B. R., 2020. Optimal stochastic dynamic scheduling for managing community recovery from natural hazards. *Reliability Engineering & System Safety,* 193, p. 106627.

Oliehoek, F. A. & Amato, C., 2016. *A concise introduction to decentralized POMDPs.* Liverpool, UK: Springer.

Oroojlooyjadid, A. & Hajinezhad, D., 2019. A review of cooperative multi-agent deep reinforcement learning. *arXiv preprint arXiv:1908.03963.*

Papakonstantinou, K. G., Amir, M. & Warn, G. P., 2021. A Scaled Spherical Simplex Filter (S3F) with a decreased n+2 sigma points set size and equivalent 2n+1 Unscented Kalman Filter (UKF) accuracy. *Mechanical Systems and Signal Processing,* Volume 163, p. 107433.

Papakonstantinou, K. G., Andriotis, C. P. & Shinozuka, M., 2016. *Point-based POMDP solvers for life-cycle cost minimization of deteriorating structures.* In Life-Cycle of Engineering Systems, Delft, The Netherlands, CRC Press, p. 427.

Papakonstantinou, K. G., Andriotis, C. P. & Shinozuka, M., 2016. *POMDP solutions for monitored structures.* Pittsburgh, PA, IFIP WG-7.5 Conference on Reliability and Optimization of Structural Systems.

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

Papakonstantinou, K. G. & Shinozuka, M., 2014. Optimum inspection and maintenance policies for corroded structures using partially observable Markov decision processes and stochastic, physically based models. *Probabilistic Engineering Mechanics,* Volume 37, pp. 93-108.

Papakonstantinou, K. G. & Shinozuka, M., 2014. Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part I: Theory. *Reliability Engineering & System Safety,* Volume 130, pp. 202-213.

Papakonstantinou, K. G. & Shinozuka, M., 2014. Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part II: POMDP implementation. *Reliability Engineering & System Safety,* Volume 130, pp. 214-224.

Peng, X. B., Abbeel, P., Levine, S. & Van de Panne, M., 2018. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics,* 37(4), pp. 1-4.

Pineau, J., Gordon, G. & Thrun, S., 2003. *Point-based value iteration: An anytime algorithm for POMDPs.* Acapulco, Mexico, International Joint Conference on Artificial Intelligence, pp. 1025-1032.

Poupart, P. et al., 2015. *Approximate linear programming for constrained partially observable Markov decision processes.* Austin, TX, 29th Conference of the Association for the Advancement of Artificial Intelligence (AAAI), pp. 3342-3348.

Prashanth, L. A. & Ghavamzadeh, M., 2016. Variance-constrained actor-critic algorithms for discounted and average reward MDPs. *Machine Learning,* 105(3), pp. 367-417.

Putterman, M. L., 1994. *Markov Decision process: discrete stochastic dynamic programming.* Wiley.

Rackwitz, R., Lentz, A. & Faber, M. H., 2005. Socio-economically sustainable civil engineering infrastructures by optimization. *Structural Safety,* 27(3), pp. 187-229.

Rocchetta, R. et al., 2019. A reinforcement learning framework for optimal operation and maintenance of power grids. *Applied Energy 241,* Volume 241, pp. 291-301.

Rokneddin, K., Ghosh, J., Dueñas-Osorio, L. & Padgett, J. E., 2013. Bridge retrofit prioritisation for ageing transportation networks subject to seismic hazards. *Structure and Infrastructure Engineering,* 9(10), pp. 1050-66.

Sanchez-Silva, M., Frangopol, D. M., Padgett, J. & Soliman, M., 2016. Maintenance and operation of infrastructure systems. *Journal of Structural Engineering,* 142(9), p. F4016004.

Saydam, D. & Frangopol, D., 2014. Risk-based maintenance optimization of deteriorating bridges. *Journal of Structural Engineering,* 141(4), p. 04014120.

Schöbi, R. & Chatzi, E. N., 2016. Maintenance planning using continuous-state partially observable Markov decision processes and non-linear action models. *Structure and Infrastructure Engineering,* 12(8), pp. 977-994.

Schulman, J. et al., 2015. *Trust region policy optimization.* Lille, France, International Conference on Machine Learning, pp. 1889-1897.

Shani, G., Pineau, J. & Kaplow, R., 2013. A survey of point-based POMDP solvers. *Autonomous Agents and Multi-Agent Systems,* 27(1), pp. 1-51.

Skordilis, E. & Moghaddass, R., 2020. A deep reinforcement learning approach for real-time sensor-driven decision making and predictive analytics. *Computers & Industrial Engineering,* 147, p. 106600.

Smith, A. E., Coit, D. W., Baeck, T., Fogel, D., & Michalewicz, Z., 1997. Penalty functions. *Handbook of Evolutionary Computation,* Volume 1, p. 97.

Sondik, E., 1971. *The optimal control of partially observable Markov processes.* Stanford, CA: Stanford University, Stanford Electronics Labs.

Sørensen, J. D., 2009. Framework for risk-based planning of operation and maintenance for offshore wind turbines. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology,* 12(5), pp. 493-506.

Straub, D. & Faber, M. H., 2005. Risk based inspection planning for structural systems. *Structural Safety,* 27(4), pp. 335-355.

Tessler, C., Mankowitz, D. J. & Mannor, S., 2018. Reward constrained policy optimization.. *arXiv preprint arXiv:1805.11074.*

Uryasev, R. & Rockafellar, S., 2002. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance,* 26(7), pp. 1443-1471.

Walraven, E. & Spaan, M. T., 2018. Column generation algorithms for constrained POMDPs. *Journal of Artificial Intelligence Research,* Volume 62, pp. 489-533.

Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., & de Freitas, N., 2016. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224.*

Yang, D. Y. & Frangopol, D. M., 2019. Life-cycle management of deteriorating civil infrastructure considering resilience to lifetime hazards: A general approach based on renewal-reward processes. *Reliability Engineering & System Safety,* Volume 183, pp. 197-212.

Zhang, N. & Alipour, A., 2020. A two-level mixed-integer programming model for bridge replacement prioritization. *Computer-Aided Civil and Infrastructure Engineering,* 35(2), pp. 116-33.

Zhang, Y., Vuong, Q. & Ross, K. W., 2020. First order optimization in policy space for constrained deep reinforcement learning. *arXiv preprint arXiv:2002.06506.*

CHAPTER 4

Conclusions

This report develops a prediction and decision-making framework for inspecting and maintaining deteriorating systems with incomplete information and constraints. In doing so, a Partially Observable Markov Decision Processes (POMDPs) approach is used, with an original deep reinforcement learning formulation. Thus, a Deep Decentralized Multiagent Actor-Critic (DDMAC) architecture is devised and manages to successfully tackle numerous challenges imposed by this stochastic control problem. In the decentralized architecture, each system component, or control unit comprised of several components, functions as an autonomous agent. While sharing a common goal, each agent can choose their own inspection and maintenance actions.

Various constraints are also effectively incorporated in this framework. Hard constraints relate to deterministic quantities and available resources (e.g., yearly or 5-year fiscal budgets) and soft constraints are pertinent to stochastic measures, such as risk thresholds to be satisfied in expectation. Many interesting results and insights have been obtained based on various low- and high-budget scenarios for infrastructure systems, as described in Chapter 3. For example, inspection can play a less important role in low-budget scenarios, multi-agent cooperation emerges when resources are limited, etc. Overall, the DDMAC solutions significantly outperform traditional and state-of-the-art inspection and maintenance planning formulations, demonstrating exceptional flexibility and multi-agent cooperation in general, diverse contexts.

Further, a deterioration model for bridge decks using Random Survival Forest is developed. The results suggest that AI methods can achieve high accuracy in predicting the deterioration pattern of bridge decks, which is an important input into the stochastic optimal control framework. The accuracy can be improved using models that specifically consider censored data, i.e., random survival forest. However, the drawback of these data-based AI predictive models is that it is difficult to interpret the impacts of different variables on deterioration. Therefore, while AI methods may be preferred for prediction, for construction or design purposes traditional stochastic methods can be more powerful.