Data-driven maintenance optimization

Victor Cohen[1],Axel Parmentier[1]

Ecole des Ponts Paristech, CERMICS (ENPC), F-77455 Marne-la-Vallée, France

{victor.cohen,axel.parmentier}@enpc.fr

Mots-clés : *Data-driven maintenance, Markov Decision Processes, Linear Programming*

1 Introduction

Context. The quantity of data available on industrial systems has dramatically increased in the last few years. The goal of predictive maintenance is to exploit this data to predict failures, maintain the equipments before they fail, and reduce the overall maintenance costs. Our objective is to provide a data-driven optimization framework for airplane maintenance. Airlines performances largely depend on their ability to operate their airplanes as much as possible and reliably. For each equipment of an airplane, the airline has access to a collection of 20 to 200 time series corresponding to different kinds of signals recorded during flights, such as the temperature of a pump. When an airplane arrives in maintenance, the airline uses this data to decide which equipments it should maintain. Maintenance decisions must strike a balance between costs due to over-maintenance and due to equipments failures. Besides, since maintenance slots are rare, airlines must prioritize between different equipments.

Our goal is to provide a decision support methodology that, given the information available at the beginning of a maintenance slot, indicates which equipments should be maintained. To achieve this goal, we do not have any model or simulator of the equipments behavior, but we have several years of historical data containing sensors values, several failure dates, and maintenance dates. The problem we consider is therefore a *data-driven multistage stochastic optimization problem*.

Current approach and its limits. In their current practice, the airline maintenance engineers use *fault trees* to support their decisions. It takes in input a high-dimensional vector of features extracting relevant information from the time series. Using a succession of binary rules splitting the features space in two, it partitions this high-dimensional feature space into a small number of clusters. Each of the clusters has a meaning from an engineering point of view: some correspond to normal behavior, some to high failure risk. The engineers maintain the equipment if it is in a high failure risk cluster. An example of fault tree is illustrated on Figure 1. Using machine learning terminology, a fault tree is a decision tree, generally hand-designed and of small size. This approach reduces the failures and maintenance costs when a small number of failure-prone equipments are considered. Indeed, when only very few equipments are considered, prioritizing between equipments is not an issue, and this heuristic makes perfectly sense. Furthermore, the fault tree used can make these diagnosis very accurately because it leverages a physical understanding of the equipments.

However, such an approach cannot be extended to a large number of equipments. When there are many equipments and scarce maintenance resources, anticipating the future on several decision steps becomes crucial, and a fault tree-based heuristic cannot work anymore. We therefore need a richer multistage stochastic optimization approach, which itself requires a richer prediction model.

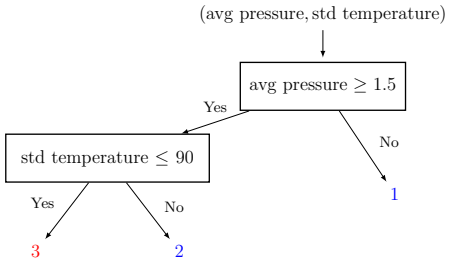

FIG. 1 – An example of fault tree that takes in input the average (avg) of the pressure and the standard deviation (std) of the temperature. It returns a discrete label in $\{1, 2, 3\}$. Each label corresponds to a cluster. Clusters 1 and 2 correspond to normal behavior (blue), and cluster 3 corresponds to high-failure risk (red).

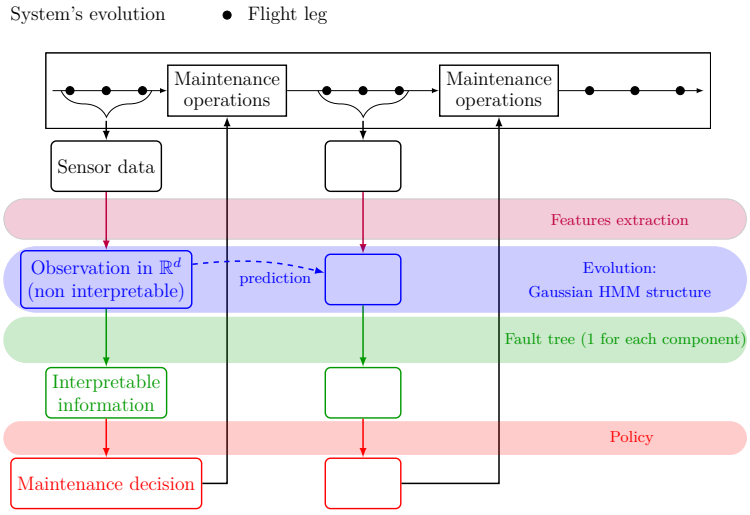

FIG. 2 – The three elements of our approach: the prediction model(in blue), the fault tree(in green) and the policy (in red).

How to build a prediction model trusted by maintenance engineers. Maintenance operations are expansive and time consuming. And when a failure happens, the airline has to cancel the next flights operated by the airplane, which is even more expensive. Hence, our statistical model and the maintenance decisions suggested by our approach must be trusted by the maintenance engineers. There are two ways of building trust in models: experimental testing, or validation by experts that understand the model. A specific difficulty comes from the fact that the data are *censored*. Indeed, airlines try to avoid failures as much as possible, which leads to a small number of failures observed on the whole history of each equipment. Given the small number of failures in our dataset, and that we do not have access to a simulator, experimental validation is not possible. *We must therefore propose a model that maintenance experts can validate.*

Our approach represented on Figure 2, solves that conundrum as follows: *our model takes decision using only information that maintenance experts can easily understand*. Since fault trees are the standard method used by the engineers of our airline to take maintenance decisions we interpret "information that maintenance expert can easily understand" as "the result of a small size fault tree". We use a *statistical model* to predict the evolution of the system, which we observe only through sensor data. We observe a vector of features in $\mathbb{R}^d$ for each component, where d is the number of features, extracted from the sensor data. An example of feature can be the average pressure in a pump over a flight. We have vector of features for each component. To predict the evolution of these observations, we use a Hidden Markov Model (HMM) with finite state space and Gaussian observations (in blue on Figure 2). We then use a *fault tree* (in green on Figure 2), which turns features into informations that can be interpreted by experts. Finally, the *policy* (in red on Figure 2) turns this interpretable information into a maintenance decision. We insist on the fact that only the fault tree is validated by experts. Since we do not have a simulator for the airplane equipments, we evaluate the performance of our approach using a simple simulator of a system with multiple components.

Literature review. The tools we used to build our statistical model are well-known in predictive maintenance literature: HMMs and fault trees. The originality of our work lies in the way we combine these tools to obtain a HMM with discrete states and discrete observations, where observations can easily be interpreted by maintenance engineers. The resulting optimization problem we obtain is a *Partially Observed Markov Decision Process* (POMDP).

Our HMMs model a component's degradation. Their latent variables represent the degradation states, and their observed variables correspond to the available vector of features in $\mathbb{R}^d$ [7].

While in airline industry the fault trees are hand-designed, state-of-the-art machine learning algorithms allow to automatically learn decision trees that can be interpreted as fault trees [11].

POMDPs have already been used to model maintenance problems [6, Sec 12.5] with a single component. The component's degradation is modeled using a discrete-time Markov chain in a finite space and the observations are emitted randomly and independently given current state. Each state transition leads to an immediate reward and the goal is to maximize the expected average reward over a finite horizon. However, the POMDPs we consider model a system with multiple components. Due to the *curse of dimensionality* [9], usual POMDPs algorithms Krishnamurthy [6] cannot solve optimally our problem. Indeed, the state space and observation space grow exponentially with the number of components. Recent work [1] gives new approximate policies for a similar problem. However, the statistical model considered has at most two or three degradation states for each component, which limits the prediction's quality of component's deterioration. Our optimization approach admits a more complex statistical model. We choose to restrict our POMDP to *memoryless policies*, i.e., policies that only depend on the current observation instead of the history of decisions and observations [8]. To the best of our knowledge, there is no compact formulation that solves a POMDP with memoryless policies.

Contributions. Our contribution is threefold.

1. We introduce a general data-driven approach to the maintenance problem. It takes sensor data in input, and the components to maintain in output at each maintenance slot. Numerical experiments (Section 5) on simulated systems with several components show that our approach leads to about 45% cost saving over the current practice in the airline industry.

Making this approach required several contributions. First, we propose a new statistical model.

2. Even if HMMs and decision trees are well-known in predictive maintenance literature, it is not common to combine them to learn a HMM with discrete and interpretable observations from continuous sensor data. The technical challenge was to learn a decision tree that turns continuous non-interpretable features into a finite set of interpretable observation. (Section 3)

Second, on the optimization part, we make several contributions to the literature on POMDPs.

3. A mixed integer linear program (MILP) that solves optimally a POMDP in the set of memoryless policies. To the best of our knowledge, it is one of the fastest method to solve POMDP in the set of memoryless policies. Furthermore, the linear relaxation of our MILP provides an upper bound on the value of the POMDP. (Section 4)
4. An extended formulation with new valid inequalities that improve the resolution of our MILP. Such inequalities come from a probabilistic interpretation of the dependence between random variables. (Section 4)
5. A MILP-based heuristic to solve POMDPs with multiple components (Section 4). It leverages a *weakly coupled POMDPs* modeling where each component evolves independently and individually as a POMDP. It extends the definition of weakly coupled *Markov Decision Processes* (MDPs) of Adelman and Mersereau [2] to POMDPs. (Section 2).

We emphasize that our approach is easy to implement since it only requires the use of a machine learning toolbox and a MILP solver.

2 Model

The system is composed of M components. At each maintenance slot $t \in \{1, 2, \dots\}$, the decision maker receives a observation O_t^m from component m, that belongs to a finite space $\mathcal{X}_O^m$. The degradation of component m is modeled using a state S_t^m, that belongs to a finite state

space $\mathcal{X}_S^m$ and is not observed by the decision maker. We assume that there is a failure state $s^{m,F}$ in $\mathcal{X}_S^m$ for each component m in $[M]$. Let $\mathcal{X}_S = \mathcal{X}_S^1 \times \cdots \times \mathcal{X}_S^M$ and $\mathcal{X}_O = \mathcal{X}_O^1 \times \cdots \times \mathcal{X}_O^M$ be the state space and observation space of the complete system. In practice the sensor data observed are continuous. Hence, $\mathcal{X}_O^m$ should be a continuous space. However, the statistical methodology of Section 3 shows how we can learn from continuous data a model with $\mathcal{X}_O^m$ finite.

Component m starts in state s with probability $p^m(s)$. At each time t, component m is in state $S_t^m = s$, it emits an observation $O_t^m = o$ with probability $p^m(o|s)$. Then, based on the observations of the full system $\mathbf{O}_t = (O_t^1, \ldots, O_t^M)$, the decision maker takes an action $\mathbf{A}_t = (A_t^1, \ldots, A_t^M)$ in a finite space $\mathcal{X}_A$, where A_t^m is the action taken on component m. Since the decision maker can maintain at most one component at each maintenance slot, we assume that the set of action is $\mathcal{X}_A = \left\{\mathbf{a} = (a^1, \ldots, a^M) \in \{0,1\}^M \text{ s.t } \sum_{m=1}^M a^m \leq 1\right\}$ and $A_t^m = 1$ indicates that component m is maintained. We assume that when a component is maintained, it behaves like a new one. Each component m then evolves independently from state $S_t^m = s$ to state $S_{t+1}^m = s'$ with probability $p^m(s'|s,a)$, and the decision maker receives reward $r^m(s,a,s')$. Then, $p^m(s'|s,a)$ is equal to $p^m(s)$ if $a = 1$, and $p^m(s'|s)$ otherwise. The probabilities factorize as

$$\mathbb{P}(\mathbf{O}_t = \mathbf{o}|\mathbf{S}_t = \mathbf{s}) = \prod_{m=1}^M p^m(o^m|s^m), \text{ and } \quad \mathbb{P}(\mathbf{S}_{t+1} = \mathbf{s}'|\mathbf{S}_t = \mathbf{s}, \mathbf{A}_t = \mathbf{a}) = \prod_{m=1}^M p^m(s'^m|s^m, a^m),$$

where $\mathbf{S}_t = (S_t^1, \ldots, S_t^M)$ for all $t \geq 1$. We associate a maintenance cost C_R^m and a failure cost C_F^m to each component m. The individual immediate reward function can be written $r^m(s^m, a^m, s'^m) = -\mathbb{1}_{s'^m = s_F^m} C_F^m - \mathbb{1}_{a^m=1} C_R^m$. We assume that the reward decomposes additively: $r(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}') = \sum_{m=1}^M r_m(s^m, a^m, s'^m)$. We suppose that we look over a finite horizon T. We explain this assumption at the beginning of Section 4. The choices made by the decision maker are modeled using a *policy* $\boldsymbol{\delta} = (\delta^1, \ldots, \delta^T)$, where δ^t is the conditional probability distribution of action $\boldsymbol{A}_t$ given observation $\mathbf{O_t}$. We introduce the set of policies Δ defined as follows $\Delta = \left\{\delta \in \mathbb{R}^{T \times |\mathcal{X}_A| \times |\mathcal{X}_O|}, \sum_{\boldsymbol{a} \in \mathcal{X}_A} \delta^t_{\boldsymbol{a}|\boldsymbol{o}} = 1 \text{ and } \delta^t_{\boldsymbol{a}|\boldsymbol{o}} \geq 0, \forall \boldsymbol{o} \in \mathcal{X}_O, \boldsymbol{a} \in \mathcal{X}_A\right\}$. The goal is to find a policy $\boldsymbol{\delta}$ maximizing the average expected reward over the horizon T

$$\max_{\boldsymbol{\delta} \in \Delta} \mathbb{E}_{\boldsymbol{\delta}}\left[\sum_{t=1}^T r(\boldsymbol{S}_t, \boldsymbol{A}_t, \boldsymbol{S}_{t+1})\right], \tag{1}$$

where the expectation in (1) is taken with respect to the distribution $\mathbb{P}_{\boldsymbol{\delta}}$ induced by $\boldsymbol{\delta}$. Problem (1) is a POMDP restricted to the set of memoryless policies Δ, which is a NP-hard problem [8]. In a POMDP, the policy usually depends on the history of observations and actions at each time step, which leads to its PSPACE-completeness [9]. We choose a memoryless policy for two reasons. First, maintenance technicians can easily apply them in practice. Second, weakly coupled POMDPs are practically intractable even for a small number of components. For example, if $M = 3$ and $|\mathcal{X}_S^m| = |\mathcal{X}_O^m| = 10$, then POMDP solvers cannot solve it.

3 Learning the model

The purpose of this section is to explain how to learn the parameters of the weakly coupled POMDP $(p^m(s), p^m(s'|s), p^m(o|s))_{s,s',o,m}$. This task is done in two steps. We first learn the parameters of a Gaussian HMM, and then, we learn a fault tree to make the observations discrete and interpretable. Since our methodology can be applied to any component, we omit index m in this section.

Gaussian HMM. At each time t, let $X_t \in \mathbb{R}^d$ be the continuous observation where $d \in \mathbb{N}$ is the dimension. We make the widely used assumption that the continuous observations follow a

Gaussian distribution given the current state, i.e., $X_t|S_t = s \sim \mathcal{N}(\alpha_s, \Sigma_s)$ for all $s \in \mathcal{X}_S$, where $\alpha_s \in \mathbb{R}^d$ and $\Sigma_s \in \mathbb{R}^{d\times d}$. This normality assumption is sound for a deteriorating system[5, p.585]. For generic HMM, there is no restriction on the choice of $p(s'|s)$. Since we model a deteriorating system, we enforce this deterioration in the parameters of HMM using a *left-right* HMM, i.e., there exists an total ordering $<$ on $\mathcal{X}_S$ and $p(s'|s) = 0$ when $s' < s$. The assumption means that a component cannot repair itself. Given the dataset of observations, we estimate parameters $(p(s), p(s'|s), \alpha_s, \Sigma_s)_{s,s'\in\mathcal{X}_S}$ using Baum-Welch algorithm [10]. The number of states $|\mathcal{X}_S|$ is also optimized using the *Bayesian Information Criterion* [7].

Fault tree. We explain how to learn a fault tree. We precise the input/output of the fault tree. Let $x_1, x_2, \ldots, x_{\tilde{T}} \in \mathbb{R}^d$ be a sequence of $\tilde{T}$ continuous observations, with $\tilde{T} \in \mathbb{N}$. We compute the most probable sequence of states $\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_{\tilde{T}}$ using the Viterbi algorithm [10]. Then, we learn a decision tree that takes as input x_t and predicts $\hat{s}_t$, for all t in $[T]$. The decision tree obtained $\hat{f}$ is a classification tree that maps any continuous observations to discrete states, i.e., $\hat{f} : \mathbb{R}^d \to \mathcal{X}_O$ where $\mathcal{X}_O = \mathcal{X}_S$ is a finite space. We choose the most frequently used CART (Classification And Regression Trees) algorithm to learn $\hat{f}$. Then, the emission probabilities can be written $p(o|s) = \mathbb{E}\left[\mathbb{1}_{\{\hat{f}(X_t)=o\}}|S_t = s\right]$ for all $s \in \mathcal{X}_S$ and $o \in \mathcal{X}_O$. Hence, we can compute it easily with Monte-Carlo simulation, even in large dimension.

4 Decision making

We have now a weakly coupled POMDP and we explain how to solve it. We choose to solve the problem over a finite horizon for two reasons. First, since we only have estimated HMM parameters, the prognostic over long horizon may be inaccurate. Second, in practice the airline would like to modify the planning only over a short horizon. We introduce a Mixed-Integer Linear Program (MILP) that solves Problem (1) with only one component ($M = 1$). We propose a tractable MILP-based heuristic to solve the weakly coupled POMDPs with $M \geq 1$.

Single component. We define the set of *deterministic* policies $\Delta^d = \Delta \cap \{0,1\}^{T\times|\mathcal{X}_A|\times|\mathcal{X}_O|}$. It is known that there always exists an optimal deterministic policy for Problem (1) [3, Prop. 1]. We introduce a collection of variables $(\boldsymbol{\mu} = \left((\mu^t_s)_s, (\mu^t_{sa})_{s,a}, (\mu^t_{soa})_{s,o,a}\right)_t, \boldsymbol{\delta} = (\delta_1, \ldots, \delta_T)$ and the following mixed-integer linear program (MILP).

$$\max_{\boldsymbol{\mu},\boldsymbol{\delta}} \sum_{t=1}^{T} \sum_{\substack{s,s'\in\mathcal{X}_S \\ a\in\mathcal{X}_A}} r(s,a,s')p(s'|s,a)\mu^t_{sa} \tag{2a}$$

$$\text{s.t. } \mu^1_s = p(s) \qquad \forall s \in \mathcal{X}_S \tag{2b}$$

$$\mu^{t+1}_{s'} = \sum_{s\in\mathcal{X}_S, a\in\mathcal{X}_A} p(s'|s,a)\mu^t_{sa} \qquad \forall s' \in \mathcal{X}_S, t \in [T] \tag{2c}$$

$$\mu^t_{sa} = \sum_{o\in\mathcal{X}_O} \mu^t_{soa} \qquad \forall s \in \mathcal{X}_S, a \in \mathcal{X}_A, t \in [T] \tag{2d}$$

$$\mu^t_{soa} \leq p(o|s)\mu^t_s \qquad \forall s \in \mathcal{X}_S, o \in \mathcal{X}_O, a \in \mathcal{X}_A, t \in [T] \tag{2e}$$

$$\mu^t_{soa} \leq \delta^t_{a|o} \qquad \forall s \in \mathcal{X}_S, o \in \mathcal{X}_O, a \in \mathcal{X}_A, t \in [T] \tag{2f}$$

$$\mu^t_{soa} \geq p(o|s)\mu^t_s + \delta^t_{a|o} - 1 \qquad \forall s \in \mathcal{X}_S, o \in \mathcal{X}_O, a \in \mathcal{X}_A, t \in [T] \tag{2g}$$

$$\boldsymbol{\delta} \in \Delta^d, \ \boldsymbol{\mu} \geq 0 \tag{2h}$$

Let v^* and z^* respectively denote the optimal values of Problem (1) and Problem (2).

Theorem 1. *Let $(\boldsymbol{\mu}, \boldsymbol{\delta})$ be a feasible solution of MILP (2). Then $\boldsymbol{\mu}$ is equal to the distribution $\mathbb{P}_{\boldsymbol{\delta}}$ induced by $\boldsymbol{\delta}$, and $(\boldsymbol{\mu}, \boldsymbol{\delta})$ is an optimal solution of MILP (2) if, and only if $\boldsymbol{\delta}$ is an optimal deterministic policy of Problem (1). Furthermore, $v^* = z^*$, and the linear relaxation of MILP (2) gives an upper bound of the POMDP.*

Theorem 1 ensures that μ_s^t, μ_{sa}^t and μ_{soa}^t respectively represent the probabilities $\mathbb{P}_{\boldsymbol{\delta}}(S_t = s)$, $\mathbb{P}_{\boldsymbol{\delta}}(S_t = s, A_t = a)$ and $\mathbb{P}_{\boldsymbol{\delta}}(S_t = s, O_t = o, A_t = a)$. It means that constraints of MILP (2) encode a solution $\boldsymbol{\mu}$ that satisfies all conditional independences between the random variables. In particular, constraints (2e), (2f) and (2g) correspond to the exact linearization of non linear constraints $\mu_{soa}^t = \delta_{a|o}^t p(o|s)\mu_s^t$, for all s in $\mathcal{X}_S$, o in $\mathcal{X}_O$, a in $\mathcal{X}_A$ and t in $[T]$, ensuring that A_t is conditionally independent of S_t given O_t. Such independences are no longer enforced in the linear relaxation of Problem (2), since the linearization is no longer exact.
We now introduce an extended formulation and valid inequalities that enable to restore slightly weaker independences in the linear relaxation of MILP (2): A_t is independent from S_t given O_t, A_{t-1} and S_{t-1}. We introduce new variables $\mu_{s'a'soa}^t$ that can be interpreted as the probabilities $\mathbb{P}(S_{t-1} = s', A_{t-1} = a', S_t = s, O_t = o, A_t = a)$. Consider the following equalities

$$\sum_{s' \in \mathcal{X}_S, a' \in \mathcal{X}_A} \mu_{s'a'soa}^t = \mu_{soa}^t, \qquad \forall s \in \mathcal{X}_S, o \in \mathcal{X}_O, a \in \mathcal{X}_A, \tag{3a}$$

$$\sum_{a \in \mathcal{X}_A} \mu_{s'a'soa}^t = p(o|s)p(s|s', a')\mu_{s'a'}^{t-1}, \qquad \forall s', s \in \mathcal{X}_S, o \in \mathcal{X}_O, a' \in \mathcal{X}_A, \tag{3b}$$

$$\mu_{s'a'soa}^t = p(s|s', a', o) \sum_{\overline{s} \in \mathcal{X}_S} \mu_{s'a'\overline{s}oa}^t, \qquad \forall s', s \in \mathcal{X}_S, o \in \mathcal{X}_O, a', a \in \mathcal{X}_A, \tag{3c}$$

where $p(s|s', a', o) = \mathbb{P}(S_t = s | S_{t-1} = s', A_{t-1} = a', O_t = o)$. Equalities (3) are valid inequalities for Problem (2) and provide a tighter root relaxation. It helps the resolution in practice [4].

Multiple components. Since a weakly coupled POMDP is a POMDP, we can still solve Problem (1) using MILP (2). However, the number of constraints and variables grows exponentially with the number of components M, and even the linear relaxation of MILP (2) becomes quickly intractable. We propose a heuristic that repeatedly solves a tractable MILP. We introduce new variables $\boldsymbol{\tau}^m = \left((\tau_s^{t,m})_s, (\tau_{sa}^{t,m})_{s,a}, (\tau_{soa}^{t,m})_{s,o,a}, (\tau_a^{t,m})_a\right)_{t \in [T]}$, $\boldsymbol{\delta}^m = (\delta^{t,m})_{t \in [T]}$ for all m in $[M]$, and the following mixed-integer linear program.

$$\max_{\boldsymbol{\tau}, \boldsymbol{\delta}} \sum_{t=1}^{T} \sum_{m=1}^{M} \sum_{\substack{s, s' \in \mathcal{X}_S^m \\ a \in \{0,1\}}} r_m(s, a, s') p^m(s'|s, a) \tau_{sa}^{t,m} \tag{4a}$$

$$\text{s.t. } (\boldsymbol{\tau}^m, \boldsymbol{\delta}^m) \text{ satisfy constraints of (2)} \qquad \forall m \in [M] \tag{4b}$$

$$\sum_{s \in \mathcal{X}_S^m} \tau_{sa}^{t,m} = \tau_a^{t,m} \qquad \forall a \in \{0,1\}, m \in [M], t \in [T] \tag{4c}$$

$$\sum_{m \in [M]} \tau_1^{t,m} \leq 1 \qquad \forall t \in [T] \tag{4d}$$

Variables $(\tau_s^{t,m})_s$, $(\tau_{sa}^{t,m})_{s,a}$, $(\tau_{soa}^{t,m})_{soa}$, and $(\tau_a^{t,m})_a$ can still be interpreted as probability distributions on $\mathcal{X}_S^m$, $\mathcal{X}_S^m \times \{0,1\}$, $\mathcal{X}_S^m \times \mathcal{X}_O^m \times \{0,1\}$, and $\{0,1\}$. However, there is no guarantee that there exists a joint distribution on $(\mathcal{X}_S \times \mathcal{X}_O \times \mathcal{X}_A)^T$ from which they can be derived as marginal distributions. Constraints (4d) ensure that the individual POMDPs are linked by the maintenance capacity constraints. In practice, this new formulation gives good performances on a deteriorating system. The size of MILP (4) is tractable since the size of action space of each component is 2. Therefore, the policy can characterized by $\delta_{1|o}^{t,m}$ for $o \in \mathcal{X}_O^m$, $m \in [M]$ and $t \in [T]$. Hence the number of binary variables is $T \sum_{m \in [M]} |\mathcal{X}_O^m|$. Furthermore, constraints (3) can be added in MILP (4) to strengthen the linear relaxation.

Heuristic. We propose a heuristic policy in Algorithm (1) for weakly coupled POMDPs (1) that solves MILP (4) at each decision time. Note that when we solve Problem (4), the initial state probability distribution corresponds to the current belief state, denoted $b^m(s)$ for each component m in $[M]$. Given a new observation and the last action, we update the belief state over time according to the belief update [6, Eq. (7.11)].

Algorithm 1 Heuristic policy for Problem (1)

1: **Input** T, p^m, b^m, r^m for all $m \in [M]$, observation $\tilde{\mathbf{o}} \in \mathcal{X}_O$, action $\tilde{\mathbf{a}} \in \mathcal{X}_A$
2: Compute current belief state $\tilde{b}^m$ using $\tilde{o}^m$ and $\tilde{a}^m$ in the belief update for all m in $[M]$
3: Solve Problem (4) with initial observation $\tilde{\mathbf{o}}$ and initial probability distribution $(\tilde{b}^m)_{m\in[M]}$ to obtain an optimal solution $\boldsymbol{\tau}$
4: Compute an $\tau_1^{1,M+1} = 1 - \sum_{m=1}^{M} \tau_1^{1,m}$
5: Let $a^* \in \operatorname{argmax}_{m\in[M+1]} \tau_1^{1,m}$
6: If $a^* = M+1$, do nothing, otherwise replace a^*

Note that each iteration of Algorithm 1 solves a MILP with a polynomial number of constraints and variables. Numerical experiments in Section 5 show the efficiency of this heuristic.

Remark 1. *If K components can be replaced in maintenance with $K \geq 1$, then we replace the right-hand side of constraint* (4d) *by K. Hence, we modify steps* (4) *to* (6) *in order to replace the K components which have the greatest value of $\tau_1^{1,m}$ and above $\tau_1^{1,M+1}$.* △

5 Numerical experiments

In airline industry, maintenance decisions are taken given leverages the binary output of the fault tree $\{0, 1\}$. An equipment is maintained when the fault tree returns 1, i.e., diagnoses the presence of failure. We would like to benchmark our approach against this current practice in industry. Since we cannot evaluate it on a real dataset, we construct a simulator of a deteriorating system based on the predictive maintenance literature and we reproduce the current practice on such a system. Then, we present the benefits of using our policy over the current practice on such system. We used the library scikitlearn for all machine learning algorithms. All linear programs have been implemented in Julia with JuMP interface and solved using Gurobi 7.5.2. Experiments have been run on a server with 192Gb of RAM and 32 cores at 3.30GHz.

System's description. We want to simulate a system composed of M deteriorating components over a simulation time H. Each time t corresponds to a maintenance slot. The maintenance slots are periodically scheduled with fixed interval time h, i.e., $t \times h$ is the time of maintenance slot t. Several cracks are present in each component of the system. The deterioration of the component corresponds to the propagation of the cracks. Between two maintenance slots t and $t+1$, we have access to noisy measurements $X_i^m \in \mathbb{R}^3$ of the crack depth evolution at each time $i \in [th, (t+1)h]$ for each component $m \in [M]$. We assume that we do not need to perform feature extraction on this system. We assume that the crack depth in each component evolves independently according to the Fatigue Crack Growth (FCG) model, which is the mostly used model in maintenance predictive literature [7]. It ensures that the crack depth is driven by the Paris-Erdogan differential equation [7, Eq. (15)]. To simulate the evolution of a crack in a component, we use the discretization scheme described in Le et al. [7, Eq. (16)]. In addition to this model, we allow the crack propagation rate to increase randomly over time. A complete description of our crack depth propagation simulator is available here [1].

Current practice. We try to reproduce the current practice on our system. We learn a fault tree that detects when the crack depth is close to a critical threshold. In our case, we choose the 90th percentile, which represents the preventive aspect of the current practice. We choose to learn the decision tree using CART algorithm. The accuracy of such a learned tree is around 95%, which is much better than what is used in practice. If there are more than one component for which the corresponding decision tree returns 1, then we select the one with the highest value of $C_F^m - C_R^m$ or we select randomly if these values are equal.

1. https://github.com/Victor2175/maintenance_system

Results. We consider the policy (1) for different rolling horizon time $T = 1, 2, 5$, and the current practice in industry (Ind.). We evaluate each policy on different number of components $M \in \{3, 5\}$ and different interval time $h \in \{30, 50\}$. Our statistical approach gives HMMs between 5 and 10 states for each component. For each value of M, we randomly draw 30 sets of parameters, where a set of parameters characterizes a system of multiple components, and for each of them we simulate 50 times each policy. In total, for each value of M, a policy is evaluated 1500 times. All results below presents average values over the 1500 evaluations. A simulation consists in 200 maintenance slots regularly distributed on a periodic basis of h. We set the maintenance costs $C_R^m = 10$ and the failure costs $C_F^m = 100$ for each component $m \in [M]$. Table 1 and 2 respectively summarize the results obtained for $h = 30$ and $h = 50$. The first column indicates the number of components M. The second column indicates the policies used. Finally, the last three columns provide the average policy time, i.e., the time to solve MILP (4) at each maintenance slot, the percentage of average cost saving over policy Ind., and the average number of failures. The numerical results show that our approach outperforms the current practice in most examples and the computation time is reasonable.

M	Policy	Time (s)	Cost (%)	Failures
3	Ind.	-	-	12.8
	Alg. (1), $T = 1$	0.001	49.4	3.3
	Alg. (1), $T = 2$	0.007	54.0	1.8
	Alg. (1), $T = 5$	0.12	55.3	1.2
5	Ind.	-	-	28.4
	Alg. (1), $T = 1$	0.002	38.5	12.5
	Alg. (1), $T = 2$	0.01	45.5	8.8
	Alg. (1), $T = 5$	0.35	49.3	5.9

TAB. 1 – Policy performances for different number of components M and a time interval $h = 30$

M	Policy	Time (s)	Cost (%)	Failures
3	Ind.	-	-	26.9
	Alg. (1), $T = 1$	0.001	53.1	7.4
	Alg. (1), $T = 2$	0.008	50.6	6.8
	Alg. (1), $T = 5$	0.2	54.5	4.7
5	Ind.	-	-	43.5
	Alg. (1), $T = 1$	0.002	54.4	12.6
	Alg. (1), $T = 2$	0.02	58.3	9.9
	Alg. (1), $T = 5$	0.3	58.6	9.1

TAB. 2 – Policy performances for different number of components M and a time interval $h = 50$

Références

[1] Abderrahmane Abbou and Viliam Makis. Group maintenance: A restless bandits approach. *INFORMS Journal on Computing*, 31(4):719–731, 2019.

[2] Daniel Adelman and Adam J. Mersereau. Relaxations of weakly coupled stochastic dynamic programs. *Operations Research*, 56(3):712–727, 2008.

[3] J. A. Bagnell, Sham M Kakade, Jeff G. Schneider, and Andrew Y. Ng. Policy search by dynamic programming. In *Advances in Neural Information Processing Systems 16*, pages 831–838. MIT Press, 2004.

[4] Victor Cohen and Axel Parmentier. Linear programming for decision processes with partial information, 2018.

[5] Michael Jong Kim and Viliam Makis. Optimal control of a partially observable failing system with costly multivariate observations. *Stochastic Models*, 28(4):584–608, 2012.

[6] V. Krishnamurthy. *Partially observed markov decision processes: From filtering to controlled sensing*. 01 2016.

[7] Thanh Trung Le, Florent Chatelain, and Christophe Bérenguer. Hidden Markov Models for diagnostics and prognostics of systems under multiple deterioration modes. In *24th European Safety and Reliability Conference*, Sep 2014.

[8] Michael L. Littman. Memoryless policies: Theoretical limitations and practical results. In *In*, pages 238–245. The MIT Press, 1994.

[9] Christos Papadimitriou and John N. Tsitsiklis. The complexity of markov decision processes. *Math. Oper. Res.*, 12(3):441–450, aug 1987.

[10] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSp Magazine*, 1986.

[11] Alice X. Zheng, J. Lloyd, and E. Brewer. Failure diagnosis using decision trees. In *Proceedings of the First International Conference on Autonomic Computing*, 2004.