



HAL
open science

A Decomposition Method by Interaction Prediction for the Optimization of Maintenance Scheduling

Thomas Bittar, Pierre Carpentier, Jean-Philippe Chancelier, Jérôme Lonchamp

► **To cite this version:**

Thomas Bittar, Pierre Carpentier, Jean-Philippe Chancelier, Jérôme Lonchamp. A Decomposition Method by Interaction Prediction for the Optimization of Maintenance Scheduling. *Annals of Operations Research*, 2022, 10.1007/s10479-021-04460-y . hal-02489304v2

HAL Id: hal-02489304

<https://hal.science/hal-02489304v2>

Submitted on 4 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Decomposition Method by Interaction Prediction for the Optimization of Maintenance Scheduling

T. Bittar^{*,‡} P. Carpentier[†] J-Ph. Chancelier^{*} J. Lonchamp[‡]

Abstract

Optimizing maintenance scheduling is a major issue to improve the performance of hydropower plants. We study a system of several physical components of the same family: either a set of turbines, a set of transformers or a set of generators. The components share a common stock of spare parts and experience random failures that occur according to known failure distributions. We seek a deterministic preventive maintenance strategy that minimizes an expected cost depending on maintenance and forced outages of the system. The Auxiliary Problem Principle is used to decompose the original large-scale optimization problem into a sequence of independent subproblems of smaller dimension while ensuring their coordination. Each subproblem consists in optimizing the maintenance on a single component. Decomposition-coordination techniques are based on variational techniques but the maintenance optimization problem is a mixed-integer problem. Therefore, we relax the dynamics and the cost functions of the system. The resulting algorithm iteratively solves the subproblems on the relaxed system with a blackbox method and coordinates the components. Relaxation parameters have an important influence on the optimization and must be appropriately chosen. An admissible maintenance strategy is then derived from the resolution of the relaxed problem. We apply the decomposition algorithm on a system with 80 components. It outperforms the reference blackbox method applied directly on the original problem.

1 Introduction

In industry, maintenance aims at improving the availability of physical assets and therefore impacts the overall performance of a system. There exists two main kinds of maintenance: corrective and preventive. Corrective maintenance (CM) is performed in reaction to a breakdown. Preventive maintenance (PM) is a planned operation that consists in repairing or replacing a component before a failure. Maintenance policies have an important economic impact and are therefore studied in various areas such as the electricity sector [19], the manufacturing industry [17] or civil engineering [36]. In the electricity sector, maintenance optimization plays a major role in ensuring a reliable and competitive electricity production.

In this work, we consider components of hydroelectric power plants that can be either turbines, transformers or generators. The system of interest gathers only components of the same family sharing a common stock of spare parts. This means that we consider either a system with turbines, a system with transformers or a system with generators. The time horizon is 40 years. Over time, components experience random failures that occur according to known failure distributions. Thus, the dynamics of the system is stochastic. A preventive strategy consists in choosing the dates of replacement for each component of the system. The goal is to find a preventive strategy that minimizes an expected cost depending on maintenance and on the occurrences of forced outages of

^{*}CERMICS, École des Ponts ParisTech, 6 et 8 avenue Blaise Pascal, 77455 Marne la Vallée Cedex 2, France

[†]Unité de Mathématiques Appliquées, ENSTA Paris, 828 Boulevard des Maréchaux, 91762 Palaiseau Cedex, France

[‡]EDF R&D PRISME, 6 quai Watier, 78400 Chatou, France

the system. The numerical experiments should involve systems constituted of up to 80 components in order to model the most demanding industrial case. This leads to optimization problems in high dimension that are numerically challenging.

The framework of this paper is close to [29] where a stochastic opportunistic replacement problem with a maintenance decision at each time step is studied. Here, the focus is to take into account the couplings that are induced by a common stock of spare parts and the cost due to the forced outages of the system. Some features make our problem singular:

1. We consider cutting-edge components for which there is a non-negligible delay between the order of a spare part and its arrival in the stock (several months, or even years).
2. Operational constraints impose to only look for deterministic maintenance strategies. This means that the dates of PM are chosen at the beginning of the time horizon with only a statistical knowledge of the future dates of failure: this is referred as an open-loop strategy. This differs from condition-based maintenance [28, 30] where maintenance decisions are taken given the degradation state of the components. Indeed, as the decisions depend on online observations, a condition-based maintenance strategy is stochastic.
3. Many studies consider periodic [32, 34] or age-based [11, 31] maintenance policies. Such strategies are only defined with one decision variable per component: either the periodicity of maintenance or the age at which a component is replaced. In this paper, more general strategies are considered as we can decide whether or not to perform a PM at each time step for each component. Suppose that there are T time steps and n components, then our maintenance strategy is defined by nT decision variables instead of n variables for time-based or age-based strategies. The effort is justified as we consider a system on a long-term horizon where the costs incurred by forced outages are of the order of millions of euros. Then, even a minor improvement in the maintenance strategy generates important savings.

Reviews on optimal maintenance scheduling [3, 12, 25] give a summary of the optimization techniques used for maintenance problems in the literature. They can be split in two main categories: mathematical programming and heuristic methods [19]. Heuristic methods can easily deal with non-linear objectives and constraints that arise when modeling complex industrial systems. They include genetic algorithms [2], particle swarm optimization [35] and simulated annealing [18]. However, these methods are known to be subject of the curse of dimensionality and cannot be used for our large-scale optimization problem. For high-dimensional problems, a frontal resolution is impracticable, and resorting to decomposition methods is relevant [19]. In [20], mixed integer programming is used to model a periodic maintenance optimization problem. A linear relaxation of the problem is then solved using a column generation technique. Column generation consists in iteratively solving a constrained master problem on a subset of variables and an unconstrained subproblem on the whole set of variables. The subproblem indicates which variables are to be considered in the master problem. In [23], a shutdown planning for the refueling of nuclear power plants is designed with a Benders decomposition coupled with various heuristics. Benders decomposition relies on a fixed splitting of the variables and consists in iteratively solving a master problem on the first subset of variables with initially few constraints and a subproblem on the remaining variables while the first variables are fixed to the value given by the master. The subproblems are easy to solve and generate additional constraints (Benders cuts) for the master problem that is solved again. Column generation and Benders decomposition are efficient when the problem exhibits some particular structure, but which does not appear in our study.

This is why we investigate decomposition-coordination methods [10]. Originated from the work of [5, 22, 24, 37], these decomposition schemes are based on variational techniques and consist in the iterative resolution of auxiliary problems whose solutions converge to the solution of the original

problem. The auxiliary problems are designed so that they are decomposable into independent sub-problems of smaller size. Different types of decomposition-coordination schemes can be implemented, by prices, by quantities or by prediction. They have been unified within the Auxiliary Problem Principle [13]. The types of decomposition differ in the way the auxiliary problems are designed and interpreted. In price decomposition, we aim at finding the saddle-point of the Lagrangian of the original problem. At each iteration, the auxiliary problem is the inner minimization of the Lagrangian, followed by an update of the multiplier using a projected gradient step. Price decomposition is the most commonly used decomposition-coordination method, and is applied in [27] for a power scheduling problem, in [16] for supply chain management or in [21] for maintenance scheduling. The decomposition by prediction introduced by Mesarovic [24] is often used when explicit constraints on the system dynamics are considered, which corresponds to the framework of this paper. The resolution uses a fixed-point algorithm that is more efficient than the gradient-based algorithm used in price decomposition. This scheme has been used for the optimal control of robot manipulators [33] or to solve an optimal power flow problem [26]. However, to our knowledge, it has never been applied to maintenance scheduling problems, which makes the originality of this work.

The industrial maintenance scheduling problem is modelled as a non-linear mixed integer program. We use a continuous relaxation of the system and decompose the large-scale optimization problem into several subproblems that consist in optimizing the maintenance on a single component. The decomposition algorithm iteratively solves the subproblems with the blackbox algorithm MADS [7] and coordinates the components in order to find an efficient maintenance strategy at the scale of the whole system. Then, we use the solution of the relaxed problem to design an admissible maintenance strategy for the original mixed integer problem. The aim of the paper is to show that the decomposition method can efficiently tackle maintenance problems with a large number of components, therefore it is applied on a system with 80 components. The relaxation parameters have an important influence on the optimization and must be appropriately chosen. On the test case, the decomposition method outperforms the blackbox algorithm applied directly on the original problem.

The paper is organized as follows: in Section 2, we describe the industrial system and formulate the maintenance optimization problem. The Auxiliary Problem Principle and the decomposition by prediction are introduced in Section 3. The application of the decomposition method to the maintenance optimization problem and the continuous relaxation of the system are presented in Section 4. Section 5 contains numerical results showing the efficiency of the method in high dimension. Finally, in Section 6, we conclude and give directions for future research.

2 System modeling and maintenance optimization problem

In this section, we describe the model of the studied industrial system and formulate the maintenance optimization problem. For any vector $v = (v_1, \dots, v_n)$, we denote the first k components of v by:

$$v_{1:k} = (v_1, \dots, v_k). \quad (2.1)$$

The notation $\langle \cdot, \cdot \rangle$ represents the inner product in a Hilbert space and $\|\cdot\|$ is the induced norm. Random variables are defined on a given probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and are denoted with capital bold letters. For a set $\mathcal{A} \subset \mathbb{R}^p$, we denote by $\mathbf{1}_{\mathcal{A}}$ the indicator function of the set \mathcal{A} , *i.e.* for $x \in \mathbb{R}^p$:

$$\mathbf{1}_{\mathcal{A}}(x) = \begin{cases} 1 & \text{if } x \in \mathcal{A}, \\ 0 & \text{if } x \notin \mathcal{A}. \end{cases} \quad (2.2)$$

2.1 Description of the system

This work is motivated by an industrial maintenance scheduling problem for components from hydroelectric power plants. In this paper, a *system* gathers several components of the same family:

we consider either a set of turbines, a set of transformers or a set of generators. The system also comprises a common stock of spare parts. All components belong to the same production unit and they must all be in their nominal state for the unit to produce electricity (series system). The components can either be healthy or broken, no degradation state is considered. Over time, failures occur according to known failure distributions. To replace a failed component, we perform a CM using a part from the common stock. As soon as a failure occurs, we order a new spare part. We consider complex cutting-edge components that are critical for the electricity production and for which the manufacturing and the qualification processes require several months or even years. Therefore the time of replenishment of the stock is important. If several components fail at close moments in time, it may happen that no spare part is available. In this case, some components cannot be replaced and no electricity is produced: the system is in *forced outage*. We can also perform PMs to replace healthy components in order to prevent a future failure. For a PM, we order a spare part so that it arrives just on time for the maintenance operation, therefore a PM does not use a part from the stock. A PM must be planned several years in advance as it requires the planning of the budget, the anticipation of the manufacturing of the part and involves highly specialized teams that operate all around the country and must be booked several years in advance. Hence, the dates of PMs are chosen at the initial time of the horizon based only on the statistical knowledge of the future dates of failures. We assume that PMs and CMs are performed in negligible time and that they are perfect operations meaning that a component is *as good as new* after a maintenance. As PMs are planned operations, they are cheaper than unpredictable CMs which require the unplanned requisition of highly specialized teams and induce a modification of their working schedules. PM and CM costs take into account all the costs generated by the maintenance operation (labor cost, cost of the new part, logistic cost, set up cost). Forced outage costs are much larger than that of a PM or a CM. No holding cost is considered for the stock of spare parts.

We denote by $n \in \mathbb{N}^*$ the number of physical components in the system. The horizon of study is denoted by $T \in \mathbb{N}^*$. In the sequel, $i \in \mathbb{I} = \{1, \dots, n\}$ denotes a component index, $t \in \mathbb{T} = \{0, \dots, T\}$ denotes a time step and we use the notation \mathbb{T}_{-1} for the set $\{0, \dots, T-1\}$. A sketch of the system with $n = 2$ components is represented in Fig. 1. The variables \mathbf{S}_t , $\mathbf{X}_{1,t}$ and $\mathbf{X}_{2,t}$ describe respectively the state of the stock and of the components. They are defined in §2.1.1. The variable $u_{i,t}$ represents the PM decision and is defined precisely in §2.1.2. Finally $j_{i,t}^P$, $j_{i,t}^C$ and $j_{i,t}^F$ represent the cost generated by a PM, a CM and a forced outage respectively, they are defined in §2.3. An example of the dynamics of a system with two components and a stock with initially one spare part is given in Fig. 2.

The model intends to represent the following real situations:

1. In a single production unit, we consider a system with a unique family of components from the plant (either turbines, transformers or generators). In this case, any failure of a component induces a failure of the system (series system). In a real situation, the number of components in this case can be up to 10.
2. We can also consider a system with components across several production units, still with one common stock. This time, a failure of a given component will only induce a failure of its own production unit and not of the whole system. Our methodology remains applicable to this system with the only difference that there is one forced outage cost for each unit. In this case the number of components can be up to 80.
3. Finally, the developed methodology is also easily applicable to a system with several families of components and one separate stock of spares for each family. The components can be either in a single production unit or distributed across several units.

The primary goal of our work is to show that we are able to tackle maintenance scheduling problems with a large number of components (up to 80). This is why, we design a model that is a proxy for

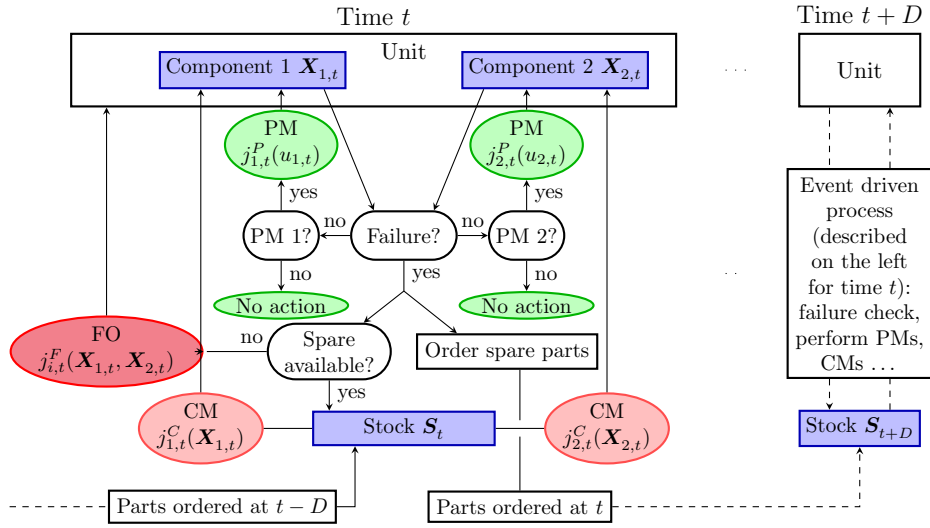


Figure 1: System of two components from a single unit sharing the same stock of spare parts. The right part of the figure is a copy of the system at time $t + D$. It emphasizes the coupling between time t and $t + D$ through the supply in spare parts.

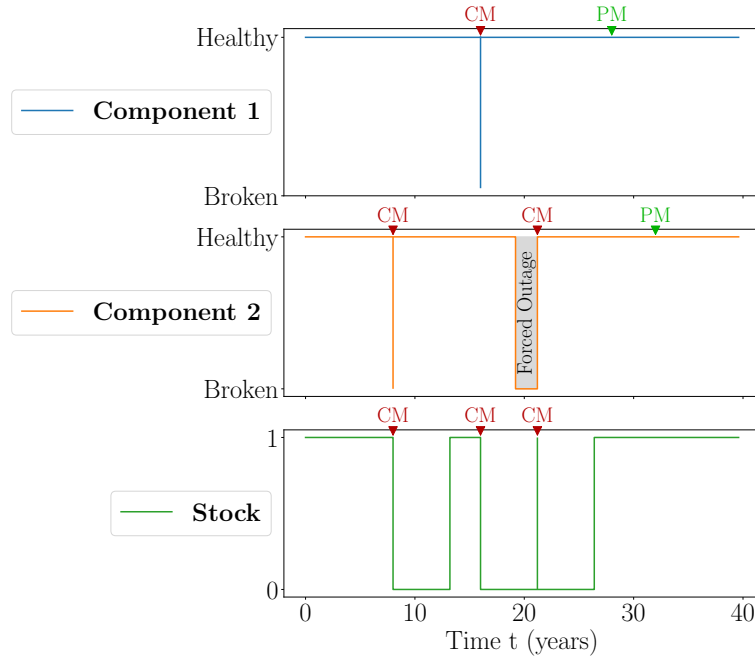


Figure 2: Illustration of the dynamics of a system with two components. When a failure occurs, if a spare part is available, a CM is performed immediately. The stock is then empty until the arrival of a new part (ordered at the time of the previous failure). If a failure occurs and the stock is empty, there is a forced outage until the arrival of a spare part. Some PMs can be performed to avoid failures, they do not use parts from the stock as they can be scheduled in advance.

the aforementioned cases. In Section 5, we explain how our model can be used to represent each of the three real situations described above.

2.1.1 Characterization of the stock and the components

The stock over time is characterized by the sequence of random variables

$$\mathbf{S} = (\mathbf{S}_0, \dots, \mathbf{S}_T) \in \mathcal{S}, \quad (2.3)$$

where $\mathbf{S}_t : \Omega \rightarrow \{0, \dots, s\}$ is the random variable representing the number of available spare parts at time t and \mathcal{S} is the set of all random variables taking values in $\{0, \dots, s\}^{T+1}$. The parameter $s \in \mathbb{N}$ is the maximum number of spare parts. The value of the initial stock is set to $\mathbf{S}_0 = s$. The replenishment delay for the parts, that is, the time from order to delivery of a part, is known and denoted by $D \in \mathbb{N}$.

At time t , component i is characterized by random variables representing:

- its regime

$$\mathbf{E}_{i,t} = \begin{cases} 0 & \text{if the component is broken,} \\ 1 & \text{if the component is healthy.} \end{cases} \quad (2.4)$$

A component has only two regimes: in the healthy regime, it runs in its nominal operating point. In the broken regime, it stops working completely. Initially all components are healthy *i.e.* $\mathbf{E}_{i,0} = 1$ for all $i \in \mathbb{I}$.

- its age (if healthy) or the time for which it has failed (if broken) denoted by the real-valued random variable $\mathbf{A}_{i,t}$. Initially the components are new *i.e.* $\mathbf{A}_{i,0} = 0$ for all $i \in \mathbb{I}$.
- the time elapsed since its last D failures

$$\mathbf{P}_{i,t} = (\mathbf{P}_{i,t}^1, \dots, \mathbf{P}_{i,t}^D), \quad (2.5)$$

where D is the number of time steps for the supply of spare parts. For $d \in \{1, \dots, D\}$, $\mathbf{P}_{i,t}^d$ is the number of time steps elapsed since the d -th undiscarded failure of component i . $\mathbf{P}_{i,t}^d$ takes a default value δ if the component has failed fewer than d times. Hence, $\mathbf{P}_{i,t}^d$ takes values in $\{\delta\} \cup \mathbb{R}_+$ and $\mathbf{P}_{i,0} = (\delta, \dots, \delta)$. The random vector $\mathbf{P}_{i,t}$ is useful to compute the dates of replenishment of the stock. It is enough to store at most the dates of the last D failures to describe the supply of the stock. More details are given in §2.2.1.

The characteristics of component i at time t are gathered in:

$$\mathbf{X}_{i,t} = (\mathbf{E}_{i,t}, \mathbf{A}_{i,t}, \mathbf{P}_{i,t}) \in \mathcal{X}_{i,t}, \quad (2.6)$$

where $\mathcal{X}_{i,t}$ is the set of all random variables defined on Ω taking values in $\{0, 1\} \times \mathbb{R}_+ \times (\{\delta\} \cup \mathbb{R}_+)^D$. The state of the system is then described at t by $(\mathbf{X}_{1,t}, \dots, \mathbf{X}_{n,t}, \mathbf{S}_t)$. Finally, to describe the components over the whole study period we introduce:

$$\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_n) = ((\mathbf{X}_{1,0}, \dots, \mathbf{X}_{1,T}), \dots, (\mathbf{X}_{n,0}, \dots, \mathbf{X}_{n,T})) \in \mathcal{X}, \quad (2.7)$$

where $\mathcal{X} = \prod_{i=1}^n \mathcal{X}_i$ and $\mathcal{X}_i = \prod_{t=0}^T \mathcal{X}_{i,t}$, for all $i \in \mathbb{I}$. In order to emphasize that \mathbf{X} depends on all the components of the system, we sometimes use the notation $\mathbf{X}_{1:n}$ instead of \mathbf{X} .

2.1.2 Preventive maintenance strategy

A PM consists in repairing a component although it is in the healthy regime. The dates of PM can be different for each component. They define the preventive maintenance strategy of the system. Operational constraints impose to look for deterministic strategies. This means that the dates of PM are chosen without any knowledge on the state of the system after the beginning of the time horizon and cannot be changed during the study. The maintenance strategy is defined by a vector

$$u = (u_1, \dots, u_n) = ((u_{1,0}, \dots, u_{1,T-1}), \dots, (u_{n,0}, \dots, u_{n,T-1})) \in \mathbb{U} = [0, 1]^{nT}, \quad (2.8)$$

where $u_{i,t}$ characterizes the PM for component i at time t . More precisely, we set a threshold $0 < \nu < 1$: a control $u_{i,t} \geq \nu$ corresponds to a rejuvenation of the component proportional to $u_{i,t}$ and a value $u_{i,t} < \nu$ corresponds to not performing a maintenance. We consider that the duration of the maintenance operation is negligible. Moreover, a PM does not use parts from the stock as it is planned in advance and the parts are ordered so that they arrive just on time for the maintenance. Note that the modeling of the PM strategy uses a continuous decision variable u , this choice is justified in §2.5.

2.1.3 Failures of the components

In our study, the distribution of the time to failure for component i is a known Weibull distribution with cumulative distribution function denoted by F_i . If component i is healthy at time t and has age $a \geq 0$, its probability of failure at time $t + \Delta t$ is given by:

$$p_i(a) = \frac{F_i(a + \Delta t) - F_i(a)}{1 - F_i(a)}. \quad (2.9)$$

We introduce the random sequence:

$$\mathbf{W} = (\mathbf{W}_1, \dots, \mathbf{W}_n) = ((\mathbf{W}_{1,1}, \dots, \mathbf{W}_{1,T}), \dots, (\mathbf{W}_{n,1}, \dots, \mathbf{W}_{n,T})) \in \mathcal{W}, \quad (2.10)$$

where \mathcal{W} is the set of all random variables defined on Ω taking values in $[0, 1]^{nT}$. The random process \mathbf{W} is an exogenous noise that affects the dynamics of the regime \mathbf{E} and the age \mathbf{A} . We assume that all $\mathbf{W}_{i,t}$ are independent random variables and follow a uniform distribution on $[0, 1]$. At time step t , component i has age $\mathbf{A}_{i,t}$. If $\mathbf{W}_{i,t+1} < p_i(\mathbf{A}_{i,t})$, component i fails at $t + 1$, otherwise no failure occurs.

2.2 Dynamics of the system

Now, we describe the dynamics of the system, that is, we explain how the variables characterizing the system evolve between two time steps.

2.2.1 Dynamics of a component

Let $i \in \mathbb{I}$ and $t \in \mathbb{T}_{-1}$. The dynamics of component i between t and $t + 1$ is described as follows.

1. If component i is healthy *i.e.* $\mathbf{E}_{i,t} = 1$:

(a) If $u_{i,t} \geq \nu$, then a PM is performed. After a PM, component i stays healthy and is rejuvenated so that:

$$(\mathbf{E}_{i,t+1}, \mathbf{A}_{i,t+1}) = (1, (1 - u_{i,t})\mathbf{A}_{i,t} + 1). \quad (2.11)$$

The PM is performed instantaneously at time t , so after the maintenance (still at time t) its age is $(1 - u_{i,t})\mathbf{A}_{i,t}$. Therefore, the age is $(1 - u_{i,t})\mathbf{A}_{i,t} + 1$ at time $t + 1$. Note that $u_{i,t} = 1$ makes the component as good as new: in this case we have $\mathbf{A}_{i,t+1} = 1$.

(b) If $u_{i,t} < \nu$, then no PM is performed. Component i fails with probability $p_i(\mathbf{A}_{i,t})$:

$$(\mathbf{E}_{i,t+1}, \mathbf{A}_{i,t+1}) = \begin{cases} (0, 0) & \text{if } \mathbf{W}_{i,t+1} < p_i(\mathbf{A}_{i,t}), \\ (1, \mathbf{A}_{i,t} + 1) & \text{otherwise.} \end{cases} \quad (2.12)$$

2. If component i is broken *i.e.* $\mathbf{E}_{i,t} = 0$:

(a) If a spare is available in the stock, a CM is performed to replace the component. We assume that the CM is an identical replacement, which implies that the component becomes as good as new. We get:

$$(\mathbf{E}_{i,t+1}, \mathbf{A}_{i,t+1}) = (1, 1). \quad (2.13)$$

A CM is performed instantaneously at time t , so the age of the component is 0 after the CM, therefore it has age 1 at time $t + 1$.

(b) If no spare part is available, the defective component stays in the broken regime:

$$(\mathbf{E}_{i,t+1}, \mathbf{A}_{i,t+1}) = (0, \mathbf{A}_{i,t} + 1). \quad (2.14)$$

As all components belong to the same power plant, when at least one component is broken, the unit is shut down until the arrival of a spare part and the execution of the CM. Such a situation is a *forced outage*. During the shut down no electricity is produced.

We have to express formally that a spare part is available for the replacement of component i . At time t , suppose that the stock has $\mathbf{S}_t = r$ parts and that m components are broken. If $r \geq m$, then all components can be replaced immediately. When $r < m$, we must choose which components to replace. Our modeling choice is to replace the broken components following the order of their index: if $i_1 \leq \dots \leq i_r \leq \dots \leq i_m$ are the indices of the broken components, we replace only the components with index i_1, \dots, i_r , the others stay in the broken regime and wait for new available parts. Using this choice, the availability of a spare part for component i corresponds to the condition:

$$\mathbf{S}_t \geq \sum_{j=1}^i \mathbf{1}_{\{0\}}(\mathbf{E}_{j,t}). \quad (2.15)$$

The right hand side of (2.15) simply counts the number of broken components with index smaller or equal than i .

To completely describe the dynamics of a component, we have to specify the dynamics of the vector $\mathbf{P}_{i,t}$. It has been introduced in §2.1.1 to store the dates of failures of the component and compute the dates for the replenishment of the stock.

– If $\mathbf{P}_{i,t} = (t_1, \dots, t_d, \delta, \dots, \delta)$ with $t_1, \dots, t_d \geq 0$, meaning that component i has undergone $d < D$ failures so far, then:

$$\mathbf{P}_{i,t+1} = \begin{cases} (t_1 + 1, \dots, t_d + 1, 0, \delta, \dots, \delta) & \text{if failure at } t + 1, \\ (t_1 + 1, \dots, t_d + 1, \delta, \delta, \dots, \delta) & \text{otherwise.} \end{cases} \quad (2.16a)$$

$$(2.16b)$$

– If $\mathbf{P}_{i,t} = (t_1, \dots, t_D)$ with $t_1, \dots, t_D \geq 0$, meaning that component i has undergone at least D failures so far, then:

$$\mathbf{P}_{i,t+1} = \begin{cases} (t_2 + 1, \dots, t_D + 1, 0) & \text{if failure at } t + 1, \\ (t_1 + 1, \dots, t_D + 1) & \text{otherwise.} \end{cases} \quad (2.17a)$$

$$(2.17b)$$

In (2.17a), note that t_1 is discarded. As $\mathbf{P}_{i,t} = (t_1, \dots, t_D)$ and $t_1 > \dots > t_D \geq 0$, we get that $t_1 \geq D - 1$. At time step $t + 1$, the part ordered from the failure at t_1 has arrived. Then, storing t_1 is not useful anymore. So if a failure occurs at $t + 1$, we can discard t_1 to make room for the new date of failure. This proves that it is enough to have $\mathbf{P}_{i,t}$ of size D to compute the replenishment of the stock as stated in §2.1.1. Note that the dates are not discarded if there is no failure (see (2.17b)), so it is possible to have $t_d > D$ for some $d \in \{1, \dots, D\}$. Such variables have no influence on the dynamics of the system.

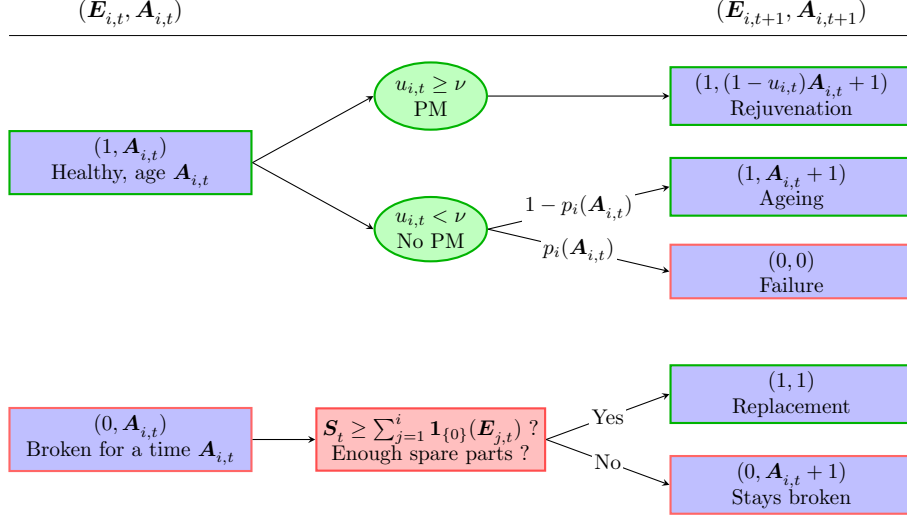


Figure 3: Dynamics of component i .

In Fig. 3, we summarize the dynamics of component i from t to $t + 1$. Recall that we have $\mathbf{X}_{i,t} = (\mathbf{E}_{i,t}, \mathbf{A}_{i,t}, \mathbf{P}_{i,t})$. We write the *dynamics of component i* on the whole time horizon as:

$$\Theta_i(\mathbf{X}_{1:i}, \mathbf{S}, u_i, \mathbf{W}_i) = 0, \quad (2.18)$$

where $\Theta_i = \{\Theta_{i,t}\}_{t \in \mathbb{T}}$ is such that:

$$\begin{cases} \Theta_{i,t+1}(\mathbf{X}_{1:i}, \mathbf{S}, u_i, \mathbf{W}_i) = \mathbf{X}_{i,t+1} - f_i(\mathbf{X}_{1:i,t}, \mathbf{S}_t, u_{i,t}, \mathbf{W}_{i,t+1}), & t \in \mathbb{T}_{-1}, \\ \Theta_{i,0}(\mathbf{X}_{1:i}, \mathbf{S}, u_i, \mathbf{W}_i) = \mathbf{X}_{i,0} - x_i, \end{cases} \quad (2.19)$$

with $x_i = (1, 0, \delta, \dots, \delta)^\top$ and f_i represents the dynamics we just described for component i . An explicit expression of f_i is given in Appendix A.

Note that there is a coupling between the dynamics of component i and the stock. There is also a coupling with components $j < i$. This is due to the choice (2.15) of replacing the broken components with the smallest indices first if there are not enough spare parts.

2.2.2 Dynamics of the stock

For the stock, the initial number of spare parts is $\mathbf{S}_0 = s$. As PMs can be anticipated, we consider that the required spares are ordered so that they arrive just on time for the scheduled maintenance. Therefore, they do not appear in the dynamics of the stock. A part is used for each CM and a new part is ordered only after the failure of a component. The number of time steps for the supply of a part is D . Hence, the part ordered after the d -th undiscarded failure of component i arrives in the stock at $t + 1$ if $\mathbf{P}_{i,t+1}^d = D$. This is equivalent to $\mathbf{P}_{i,t}^d = D - 1$. On the other hand, the number of

broken components is $\sum_{i=1}^n \mathbf{1}_{\{0\}}(\mathbf{E}_{i,t})$ and we replace as many of them as possible given the current level of stock \mathbf{S}_t . Thus, we have:

$$\mathbf{S}_{t+1} = \mathbf{S}_t + \sum_{i=1}^n \sum_{d=1}^D \mathbf{1}_{\{D-1\}}(\mathbf{P}_{i,t}^d) - \min \left\{ \mathbf{S}_t, \sum_{i=1}^n \mathbf{1}_{\{0\}}(\mathbf{E}_{i,t}) \right\}, \quad t \in \mathbb{T}_{-1}. \quad (2.20)$$

We write the *dynamics of the stock* in compact form as:

$$\Theta_{\mathbf{S}}(\mathbf{X}_{1:n}, \mathbf{S}) = 0, \quad (2.21)$$

where $\Theta_{\mathbf{S}} = \{\Theta_{\mathbf{S},t}\}_{t \in \mathbb{T}}$ is such that:

$$\begin{cases} \Theta_{\mathbf{S},t+1}(\mathbf{X}_{1:n}, \mathbf{S}) = \mathbf{S}_{t+1} - f_{\mathbf{S}}(\mathbf{X}_{1:n,t}, \mathbf{S}_t), & t \in \mathbb{T}_{-1}, \\ \Theta_{\mathbf{S},0}(\mathbf{X}_{1:n}, \mathbf{S}) = \mathbf{S}_0 - s, \end{cases}$$

with $f_{\mathbf{S}}$ corresponding to the right-hand side of (2.20). Note that \mathbf{S}_{t+1} depends on on the current level of stock \mathbf{S}_t but also on $\mathbf{X}_{i,t}$ for all $i \in \mathbb{I}$. The stock is coupling all the components of the system.

Finally, the *dynamics of the whole system* is summarized by the almost sure equality constraint $\Theta(\mathbf{X}, \mathbf{S}, u, \mathbf{W}) = 0$, where we have $\Theta : \mathcal{X} \times \mathcal{S} \times \mathbb{U} \times \mathcal{W} \rightarrow \mathcal{L}$, with $\Theta = \{\{\Theta_i\}_{i \in \mathbb{I}}, \Theta_{\mathbf{S}}\}$ and $\mathcal{L} = (\prod_{i=1}^n \mathcal{L}_i) \times \mathcal{L}_{\mathbf{S}}$ where \mathcal{L}_i is the set of random variables with range in $\mathbb{R}^{(D+2)(T+1)}$ and $\mathcal{L}_{\mathbf{S}}$ is the set of random variables with range in $\mathbb{R}^{(T+1)}$.

We have now completely described the dynamics of the system. In the next part we specify the costs associated to the system.

2.3 Costs generated by the system

The costs generated by the system are due to PMs, CMs and forced outages of the unit. In practice as PMs are scheduled in advance, they are cheaper than unpredictable CMs. A forced outage of the unit induces a loss of production. It is characterized by a yearly cost which is higher than that of a PM or a CM. We consider a discount rate τ meaning that a cost c occurring at time t will be valued $\eta_t c$ with the discount factor $\eta_t := \frac{1}{(1+\tau)^t}$. We introduce the following notations:

- $j_{i,t}^P(u_{i,t})$ is the PM cost incurred at time t for component i . Let C_i^P be the cost of a PM operation on component i . We set:

$$j_{i,t}^P(u_{i,t}) = \eta_t C_i^P u_{i,t}^2. \quad (2.22)$$

We use a quadratic cost as it is strongly convex and should favor numerical convergence. In particular, in the case where $0 < u_{i,t} < \nu$, which models a situation where no PM is performed, we have $j_{i,t}^P(u_{i,t}) > 0$.¹

- $j_{i,t}^C(\mathbf{X}_{i,t})$ is the CM cost. It is due at the time of the failure of a component, even if there is no spare part to perform the operation immediately. Hence it only occurs when $(\mathbf{E}_{i,t}, \mathbf{A}_{i,t}) = (0, 0)$. Let C_i^C be the cost of a CM operation on component i . We have:

$$j_{i,t}^C(\mathbf{X}_{i,t}) = \eta_t C_i^C \mathbf{1}_{\{0\}}(\mathbf{E}_{i,t}) \mathbf{1}_{\{0\}}(\mathbf{A}_{i,t}). \quad (2.23)$$

¹The fact that $j_{i,t}^P(u_{i,t}) > 0$ when $0 < u_{i,t} < \nu$ is favorable from a numerical point of view. For $0 < u_{i,t} < \nu$, we always have $j_{i,t}^P(u_{i,t}) > j_{i,t}^P(0)$ while no PM is performed. Hence, the system dynamics is the same with $u_{i,t} = 0$ and $0 < u_{i,t} < \nu$ while the cost is higher for $0 < u_{i,t} < \nu$. Therefore, the control $u_{i,t} = 0$ is always better to $0 < u_{i,t} < \nu$. This feature will allow us to clearly distinguish the steps where a PM is performed from the others.

- $j_t^F(\mathbf{X}_{1:n,t})$ is the forced outage cost. As all components belong to the same production unit, a forced outage occurs when at least one component is in a failed state and the CM has not been performed immediately because of a lack of spare part. Let C^F be the forced outage cost per time unit. We have:

$$j_t^F(\mathbf{X}_{1:n,t}) = \eta_t C^F \min \left\{ 1, \sum_{i=1}^n \mathbf{1}_{\{0\}}(\mathbf{E}_{i,t}) \mathbf{1}_{\mathbb{R}_+^*}(\mathbf{A}_{i,t}) \right\}. \quad (2.24)$$

In order to consider the previous costs over the whole study period we introduce:

- the *total maintenance cost* (preventive and corrective) generated by component $i \in \mathbb{I}$ on the studied period:

$$j_i(\mathbf{X}_i, u_i) = \sum_{t=0}^{T-1} j_{i,t}^P(u_{i,t}) + \sum_{t=0}^T j_{i,t}^C(\mathbf{X}_{i,t}), \quad (2.25)$$

- the *total forced outage cost* generated by the system during the studied period:

$$j^F(\mathbf{X}_{1:n}) = \sum_{t=0}^T j_t^F(\mathbf{X}_{1:n,t}), \quad (2.26)$$

2.4 Formulation of the maintenance optimization problem

The dynamics of the system is stochastic as it depends on the failure of the components, modelled by the random vector \mathbf{W} . The cost function is then stochastic as well. The objective is to find the deterministic maintenance strategy $u \in \mathbb{U}$ that minimizes the expected cost generated by the system over all failure scenarios. Hence, the industrial optimal maintenance scheduling problem is formulated as follows:

$$\begin{aligned} \min_{(\mathbf{X}, \mathbf{S}, u) \in \mathcal{X} \times \mathcal{S} \times \mathbb{U}} \mathbb{E} \left(\sum_{i=1}^n j_i(\mathbf{X}_i, u_i) + j^F(\mathbf{X}_{1:n}) \right) \\ \text{s.t. } \Theta(\mathbf{X}, \mathbf{S}, u, \mathbf{W}) = 0, \quad \mathbb{P}\text{-a.s.} \end{aligned} \quad (2.27)$$

The term $\mathbb{E}(\sum_{i=1}^n j_i(\mathbf{X}_i, u_i)) = \sum_{i=1}^n \mathbb{E}(j_i(\mathbf{X}_i, u_i))$ is additive with respect to the components whereas $\mathbb{E}(j^F(\mathbf{X}_{1:n}))$ induces a non-additive coupling between the components. In the theoretical part on the Auxiliary Problem Principle (Section 3), we will see that these two terms are treated in a different way for the design of a decomposition-coordination algorithm.

2.5 Complexity of the problem and limits of a direct approach

For maintenance scheduling problems that involve systems that are similar to the one considered in this paper, the algorithm MADS (Mesh Adaptive Direct Search) [7] can be used. In particular, MADS has been successfully applied for periodic maintenance problems in [4] and for a system with 4 components and monthly maintenance decisions in [1]. However, the problem considered here with 80 components and general maintenance strategies is more challenging than the two previous ones.

The industrial maintenance problem (2.27) is a non-smooth non-convex optimization problem. The number of components n in the system can be up to 80 and T is 40 years so \mathbb{U} has dimension up to $nT = 3200$ as we have one maintenance decision each year for each component. In §2.2, the dynamics is modelled with the vector Θ , taking values in $\mathcal{L} = \mathbb{R}^{(n(D+2)+1)(T+1)}$. Hence, there are $(n(D+2)+1)(T+1)$ constraints in Problem (2.27) *i.e.* 13161 constraints for the systems studied in Section 5 with a delay $D = 2$ years for the supply in spare parts. However, any maintenance strategy $u \in \mathbb{U}$ is feasible, as we can always choose whether or not to do a PM for each component at each

time step. The dynamics Θ then allows to compute the value of \mathbf{X} and \mathbf{S} given the strategy u and the occurrences of failures modelled by \mathbf{W} .

As the problem is non-smooth, non-convex and in high dimension, it is challenging for any frontal resolution approach, including the algorithm MADS. MADS is a blackbox algorithm, meaning that evaluation points are chosen iteratively without using the gradients of the objective function, which may not even be defined. This feature is particularly appealing as the cost function is not differentiable. The algorithm has been initially designed for continuous optimization, therefore it uses the modeling of §2.1 and the cost functions described in §2.3. In particular, the PM strategies are modelled with a continuous decision variable. When the number of components is not too large, say $n < 10$, the maintenance problem (2.27) is solved efficiently by MADS: the algorithm converges in around a minute. When the number of components is large, MADS needs more iterations to explore the high-dimensional space of solutions. As the volume of the search space grows exponentially with the number of components, MADS may not be able to find an effective maintenance strategy in reasonable time (less than a day) for the most demanding case with 80 components.

To overcome the difficulty of MADS when dealing with large systems, we implement a decomposition of Problem (2.27) component by component.

3 The Auxiliary Problem Principle for decomposition

When performing optimization on a large scale system, that is, a system which is described by a large number of variables or constraints, the computation is often highly expensive either in time or in memory requirement (or both). The idea of decomposition is to formulate subproblems involving only smaller subsystems of the original large system. Each subproblem is easier to solve than the global optimization problem and provides its “local” solution. Then the goal of coordination is to ensure that gathering the local solutions leads to a global solution. Decomposition-coordination methods usually result in an iterative process alternating an optimization step on the subsystems and a coordination step that updates the subproblems.

The main advantage of decomposition is that the resolution of the small subproblems is faster than the original problem. More than that, the computational complexity of an optimization problem is often superlinear or even exponential in the size of the problem. Hence, the sum of the computational efforts required for the resolution of all subproblems will be lower than for the global problem, even if the resolution of the subproblems must be carried out multiple times. Another feature of decomposition methods is that they are naturally adapted to parallelization as each subproblem is independent. This leads to a reduction of computation time.

In this section, we introduce the general framework of the Auxiliary Problem Principle (APP) that allows us to formulate a decomposition-coordination scheme for the optimization problem (2.27). The APP has first been introduced in [13] as a unified framework for decomposition methods but also for other classical iterative algorithms. This principle casts the resolution of an optimization problem into the resolution of a sequence of auxiliary problems whose solutions converge to the solution of the original problem. Appropriate choices for the auxiliary problems lead to decomposition-coordination schemes.

Based on [10], we present the main ideas of the APP. Consider the following problem, which we call the master problem:

$$\min_{u \in U^{\text{ad}}} J_{\Sigma}(u) + J_{\Delta}(u) \quad \text{such that } \Theta(u) \in -C, \quad (3.1)$$

where:

- U^{ad} is a non-empty, closed, convex subset of a Hilbert space $\mathbb{U} = \mathbb{U}_1 \times \dots \times \mathbb{U}_N$ and is decomposable as $U^{\text{ad}} = U_1^{\text{ad}} \times \dots \times U_N^{\text{ad}}$ where for all $i \in \mathbb{I}$, $U_i^{\text{ad}} \subset \mathbb{U}_i$ is a closed convex set;

- C is a pointed closed convex cone of a Hilbert space $\mathcal{C} = \mathcal{C}_1 \times \dots \times \mathcal{C}_N$ and is decomposable as $C = C_1 \times \dots \times C_N$ where for all $i \in \mathbb{I}$, $C_i \subset \mathcal{C}_i$ is a closed convex cone;
- $J_\Sigma : \mathbb{U} \rightarrow \mathbb{R}$ and $J_\Delta : \mathbb{U} \rightarrow \mathbb{R}$ are convex and lower semi-continuous (l.s.c.). The function $J_\Sigma + J_\Delta$ is coercive on U^{ad} . Moreover, J_Δ is differentiable and J_Σ is additive with respect to the decomposition of the admissible space, so that we have for $u = (u_1, \dots, u_N) \in U^{\text{ad}}$ with $u_i \in U_i^{\text{ad}}$ for all $i \in \mathbb{I}$:

$$J_\Sigma(u) = \sum_{i=1}^N J_{\Sigma,i}(u_i); \quad (3.2)$$

where $J_{\Sigma,i} : \mathbb{U}_i \rightarrow \mathbb{R}$.

- $\Theta : \mathbb{U} \rightarrow \mathcal{C}$ is differentiable and C -convex, meaning that:

$$\forall u, v \in \mathbb{U}, \forall \rho \in [0, 1], \rho\Theta(u) + (1 - \rho)\Theta(v) - \Theta(\rho u + (1 - \rho)v) \in C. \quad (3.3)$$

We write $\Theta(u) = (\Theta_1(u), \dots, \Theta_N(u))$ with $\Theta_i : \mathbb{U} \rightarrow \mathcal{C}_i$ for $i \in \mathbb{I}$.

The goal of the APP is to turn the resolution of the master problem on U^{ad} into the resolution of subproblems on the sets U_i^{ad} . In the master problem, there is a non-additive coupling in the cost due to J_Δ and a coupling in the constraint Θ .

We present the APP for the special case of the decomposition by prediction. The construction of a decomposable auxiliary problem relies on the following points:

1. The decomposition of the admissible space $U^{\text{ad}} = U_1^{\text{ad}} \times \dots \times U_N^{\text{ad}}$ defines the subspace on which each subproblem is solved.
2. The decomposition of the cone $C = C_1 \times \dots \times C_N$ specifies which part of the constraint is assigned to each subproblem.

Let $\bar{u} = (\bar{u}_1, \dots, \bar{u}_N) \in \mathbb{U}$ and $\bar{\lambda} = (\bar{\lambda}_1, \dots, \bar{\lambda}_N) \in C^*$ the dual cone of C . We recall that:

$$C^* = \{\lambda \in \mathcal{C}^*, \langle \lambda, \mu \rangle \geq 0 \text{ for all } \mu \in C\}, \quad (3.4)$$

with \mathcal{C}^* being the dual space of \mathcal{C} . Let K be an auxiliary cost and Φ be an auxiliary dynamics satisfying the following properties:

- $K : \mathbb{U} \rightarrow \mathbb{R}$ is convex, l.s.c., differentiable and additive: $K(u) = \sum_{i=1}^N K_i(u_i)$;
- $\Phi : \mathbb{U} \rightarrow \mathcal{C}$ is differentiable and block-diagonal: $\Phi(u) = (\Phi_1(u_1), \dots, \Phi_N(u_N))$.

The *auxiliary problem* for the master problem (3.1) arising from the choice of K and Φ is given by:

$$\begin{aligned} \min_{u \in U^{\text{ad}}} & K(u) + J_\Sigma(u) + \langle \nabla J_\Delta(\bar{u}) - \nabla K(\bar{u}), u \rangle + \langle \bar{\lambda}, (\Theta'(\bar{u}) - \Phi'(\bar{u})) \cdot u \rangle \\ \text{s.t.} & \Phi(u) - \Phi(\bar{u}) + \Theta(\bar{u}) \in -C. \end{aligned} \quad (3.5)$$

Choosing K additive and Φ block-diagonal ensures that Problem (3.5) decomposes in N independent subproblems that can be solved in parallel. For $i \in \mathbb{I}$, the i -th subproblem is given by:

$$\begin{aligned} \min_{u_i \in U_i^{\text{ad}}} & K_i(u_i) + J_{\Sigma,i}(u_i) + \langle \nabla_{u_i} J_\Delta(\bar{u}) - \nabla K_i(\bar{u}_i), u_i \rangle \\ & - \langle \bar{\lambda}_i, \Phi'_i(\bar{u}_i) \cdot u_i \rangle + \sum_{j=1}^N \langle \bar{\lambda}_j, \partial_{u_i} \Theta_j(\bar{u}) \cdot u_i \rangle \\ \text{s.t.} & \Phi_i(u_i) - \Phi_i(\bar{u}_i) + \Theta_i(\bar{u}) \in -C_i. \end{aligned} \quad (3.6)$$

This subproblem only depends on $u_i \in U_i^{\text{ad}}$ and inherits only the i -th component of the constraint.

Example 3.1. Let $\bar{u} \in \mathbb{U}$. A canonical choice for the additive auxiliary cost function K is:

$$K(u) = \sum_{i=1}^N K_i(u_i) \quad \text{with} \quad K_i(u_i) = J_{\Delta}(\bar{u}_{1:i-1}, u_i, \bar{u}_{i+1:n}). \quad (3.7)$$

where $u_{i:j} = (u_i, \dots, u_j)$ for $i \leq j$ and the convention that $u_{i:j}$ is empty if $j > i$. Similarly, a canonical choice for the block-diagonal auxiliary dynamics Φ is:

$$\Phi(u) = (\Phi_1(u_1), \dots, \Phi_N(u_N)) \quad \text{with} \quad \Phi_i(u_i) = \Theta_i(\bar{u}_{1:i-1}, u_i, \bar{u}_{i+1:n}). \quad (3.8)$$

The general idea is to construct the i -th term of the auxiliary function from the original function where only the i -th component is allowed to vary. \triangle

The following statement is the fundamental lemma for the theory of the APP.

Lemma 3.2. *Let u^{\sharp} be a solution of the auxiliary problem (3.5) and λ^{\sharp} be an optimal multiplier for its constraint. If $(u^{\sharp}, \lambda^{\sharp}) = (\bar{u}, \bar{\lambda})$, then u^{\sharp} is a solution of the master problem (3.1) and λ^{\sharp} is an optimal multiplier for its constraint.*

The proof consists in checking that if $(u^{\sharp}, \lambda^{\sharp}) = (\bar{u}, \bar{\lambda})$, then $(u^{\sharp}, \lambda^{\sharp})$ solves the variational inequalities that are satisfied by an optimal solution and an optimal multiplier of the master problem (3.1). More details can be found in [10, Section 4]. Lemma 3.2 suggests to use the APP fixed-point Algorithm 1.

Algorithm 1 APP fixed-point algorithm

- 1: Let $(\bar{u}, \bar{\lambda}) = (u^0, \lambda^0) \in U^{\text{ad}} \times C^*$ and set $k = 0$.
 - 2: At iteration $k + 1$:
 - Solve the auxiliary problem (3.5). Let u^{k+1} be a solution and λ^{k+1} be an optimal multiplier for its constraint.
 - Set $(\bar{u}, \bar{\lambda}) = (u^{k+1}, \lambda^{k+1})$.
 - 3: If the maximum number of iterations is reached or $\|u^{k+1} - u^k\| + \|\lambda^{k+1} - \lambda^k\|$ is sufficiently small then stop, else $k \leftarrow k + 1$ and go back to step 2.
-

A convergence result for Algorithm 1, is given in [14].

Theorem 3.3. [14, Theorem 5.1 with $\epsilon = 1$] *Assume that:*

- The admissible space U^{ad} is equal to the whole space \mathbb{U} ,
- The constraints are equality constraints, that is $C = \{0\}$,
- $K(u) = \frac{1}{2} \langle u, Ku \rangle$, $J_{\Sigma}(u) = 0$, $J_{\Delta}(u) = \frac{1}{2} \langle u, Ju \rangle + \langle j, u \rangle$ where K and J are linear self-adjoint strongly monotone and Lipschitz continuous operators, j is a vector in \mathbb{U} , and $2K - J$ is assumed to be strongly monotone,
- $\Phi(u) = Ou$, $\Theta(u) = Tu + t$ where the operators O and T are linear and surjective and t is a vector in \mathbb{C} ,
- (Geometric condition) The operator $2(TJ^{-1}O^{\top} + OJ^{-1}T^{\top}) - TJ^{-1}(2K + J)J^{-1}T^{\top}$ is strongly monotone.

Then, the sequence $\{(u^k, \lambda^k)\}_{k \in \mathbb{N}}$ generated by the APP fixed-point algorithm converges strongly to the unique optimal solution $(u^{\sharp}, \lambda^{\sharp})$ of the original problem (3.1).

In this section, we have studied a master optimization problem with couplings coming both from the part J_Δ of the cost function and the constraint Θ . The APP allows to turn the resolution of the master problem into the iterative resolution of an auxiliary problem that is decomposable into independent subproblems of smaller size. In the next section, we present the application of the APP to solve the maintenance optimization problem described in Section 2.

4 Application of the APP for the industrial maintenance problem

The application of the APP to Problem (2.27) is done as follows:

1. We specify a decomposition of the admissible space and of the space of constraints (§4.1).
2. We construct an auxiliary problem that is adapted to the proposed decomposition using a suitable auxiliary cost and auxiliary dynamics (§4.2).
3. The industrial system involves integer variables whereas the APP is based on variational techniques. Therefore, we proceed to a continuous relaxation of the system in order to compute the gradients of the dynamics that appear in the auxiliary problem (§4.3).
4. We give the explicit formulation of the independent subproblems that arise from the decomposition of the auxiliary problem (§4.4).
5. We see that the subproblem on the stock is easy to solve numerically. We take advantage of this feature to design an efficient implementation of the APP fixed-point algorithm mixing a parallel and sequential strategy for the resolution of the subproblems (§4.5).

The APP fixed-point algorithm presented in §4.5 is then applied to numerical test cases in Section 5.

4.1 Decomposition of the space by component

We start by specifying the decomposition of the admissible space and of the cone of constraints that is considered for the design of the decomposition by component.

Considering the physical nature of the industrial system composed of n components and a stock, we choose to decompose the problem in $n+1$ subproblems and call this decomposition a *decomposition by component*. More precisely, for $i \in \mathbb{I}$, the i -th subproblem is called *subproblem on component i* since it is solved on $\mathcal{X}_i \times \mathbb{U}_i$ and only involves the dynamics of component i . The $(n+1)$ -th subproblem is called *subproblem on the stock* since it is solved on \mathcal{S} and only involves the dynamics of the stock. This means that the admissible space $\mathcal{X} \times \mathcal{S} \times \mathbb{U}$ of Problem (2.27) is decomposed as a product of a $n+1$ subspaces:

$$\mathcal{X} \times \mathcal{S} \times \mathbb{U} = (\mathcal{X}_1 \times \mathbb{U}_1) \times \dots \times (\mathcal{X}_n \times \mathbb{U}_n) \times \mathcal{S} , \quad (4.1)$$

where, for $(\mathbf{X}, \mathbf{S}, u) = ((\mathbf{X}_1, \dots, \mathbf{X}_n), \mathbf{S}, (u_1, \dots, u_n)) \in \mathcal{X} \times \mathcal{S} \times \mathbb{U}$, we have

$$(\mathbf{X}_i, u_i) \in \mathcal{X}_i \times \mathbb{U}_i \text{ for all } i \in \mathbb{I} \text{ and } \mathbf{S} \in \mathcal{S} . \quad (4.2)$$

The constraints in Problem (2.27), that is $\Theta(\mathbf{X}, \mathbf{S}, u, \mathbf{W}) \in -C$ with $C = \{0\}_{\mathcal{L}}$ is decomposed through the following cone decomposition:

$$C = \{0\}_{\mathcal{L}} = \{0\}_{\mathcal{L}_1} \times \dots \times \{0\}_{\mathcal{L}_n} \times \{0\}_{\mathcal{L}_S} = C_1 \times \dots \times C_n \times C_S , \quad (4.3)$$

where for $i \in \mathbb{I}$, $\{0\}_{\mathcal{L}_i}$, $\{0\}_{\mathcal{L}_S}$ and $\{0\}_{\mathcal{L}}$ denote the null function of \mathcal{L}_i , \mathcal{L}_S and \mathcal{L} respectively. We recall that Θ_i takes values in \mathcal{L}_i and Θ_S takes values in \mathcal{L}_S .

4.2 Construction of an auxiliary problem

In this section, we choose the auxiliary functions that lead to the construction of a decomposable auxiliary problem.

Problem (2.27) is not directly decomposable by component because of couplings, highlighted in Section 2, that we recall now.

- The expected maintenance cost $\mathbb{E}(\sum_{i=1}^n j_i(\mathbf{X}_i, u_i))$ is additive with respect to the decomposition by component and can be identified with J_Σ in Section 3.
- The forced outage cost j^F induces a non-additive coupling between the components. The expected forced outage cost $\mathbb{E}(j^F(\mathbf{X}_{1:n}))$ can be identified with J_Δ in Section 3.
- The dynamics Θ_i of component i induces a coupling with the stock and all components with index $j < i$. The stock dynamics, Θ_S , is coupling the stock with all components.

In order to obtain a decomposition of Problem (2.27) by component, we use the canonical technique from Example 3.1. We define an additive mapping K and a block diagonal mapping Φ so that the resulting auxiliary problem is decomposable. We also choose to augment the auxiliary cost K with a strongly convex term in order to ease the numerical convergence of the method. Let $\bar{\mathbf{X}} = (\bar{\mathbf{X}}_1, \dots, \bar{\mathbf{X}}_n) \in \mathcal{X}$, $\bar{\mathbf{S}} \in \mathcal{S}$, $\bar{u} \in \mathbb{U}$, $\bar{\boldsymbol{\Lambda}} \in C^*$ and $\gamma_x, \gamma_s, \gamma_u > 0$. We consider:

- An additive auxiliary cost function $K : \mathcal{X} \times \mathcal{S} \times \mathbb{U} \rightarrow \mathbb{R}$:

$$K(\mathbf{X}, \mathbf{S}, u) = \sum_{i=1}^n K_i(\mathbf{X}_i, u_i) + K_S(\mathbf{S}), \quad (4.4)$$

with:

$$\begin{aligned} K_i(\mathbf{X}_i, u_i) &= \mathbb{E} \left(j^F(\bar{\mathbf{X}}_{1:i-1}, \mathbf{X}_i, \bar{\mathbf{X}}_{i+1:n}) + \frac{\gamma_x}{2} \|\mathbf{X}_i\|^2 + \frac{\gamma_u}{2} \|u_i\|^2 \right), \quad i \in \mathbb{I}, \\ K_S(\mathbf{S}) &= \mathbb{E} \left(\frac{\gamma_s}{2} \|\mathbf{S}\|^2 \right), \end{aligned} \quad (4.5)$$

- A block-diagonal auxiliary dynamics mapping $\Phi : \mathcal{X} \times \mathcal{S} \times \mathbb{U} \times \mathcal{W} \rightarrow \mathcal{L}$:

$$\Phi(\mathbf{X}, \mathbf{S}, u, \mathbf{W}) = (\Phi_1(\mathbf{X}_1, u_1, \mathbf{W}_1), \dots, \Phi_n(\mathbf{X}_n, u_n, \mathbf{W}_n), \Phi_S(\mathbf{S})), \quad (4.6)$$

with:

$$\begin{aligned} \Phi_i(\mathbf{X}_i, u_i, \mathbf{W}_i) &= \Theta_i(\bar{\mathbf{X}}_{1:i-1}, \mathbf{X}_i, \bar{\mathbf{S}}, u_i, \mathbf{W}_i), \quad i \in \mathbb{I}, \\ \Phi_S(\mathbf{S}) &= \Theta_S(\bar{\mathbf{X}}_{1:n}, \mathbf{S}), \end{aligned} \quad (4.7)$$

with $\bar{\mathbf{X}}_{1:0}$ being by convention an empty vector.

We can now write an auxiliary problem for (2.27). Assume that the dynamics Θ is differentiable. In this case, Φ is differentiable and the auxiliary problem writes:

$$\begin{aligned} \min_{(\mathbf{X}, \mathbf{S}, u) \in \mathcal{X} \times \mathcal{S} \times \mathbb{U}} \mathbb{E} \left(\sum_{i=1}^n \left(j_i(\mathbf{X}_i, u_i) + j^F(\bar{\mathbf{X}}_{1:i-1}, \mathbf{X}_i, \bar{\mathbf{X}}_{i+1:n}) \right) \right. \\ \left. + \frac{\gamma_x}{2} \|\mathbf{X} - \bar{\mathbf{X}}\|^2 + \frac{\gamma_s}{2} \|\mathbf{S} - \bar{\mathbf{S}}\|^2 + \frac{\gamma_u}{2} \|u - \bar{u}\|^2 \right. \\ \left. + \langle \bar{\boldsymbol{\Lambda}}, (\Theta'(\bar{\mathbf{X}}, \bar{\mathbf{S}}, \bar{u}, \mathbf{W}) - \Phi'(\bar{\mathbf{X}}, \bar{\mathbf{S}}, \bar{u}, \mathbf{W})) \cdot (\mathbf{X}, \mathbf{S}, u) \rangle \right) \\ \text{s.t. } \Phi(\mathbf{X}, \mathbf{S}, u, \mathbf{W}) = 0. \end{aligned} \quad (4.8)$$

By construction, the auxiliary problem (4.8) is decomposable with respect to the decompositions (4.1) and (4.3).

4.3 Relaxation of the system

In the industrial case, the assumptions of Theorem 3.3 are far from being satisfied: the forced outage cost j^F is not quadratic, nor even convex, the dynamics Θ is not linear, and it is hard to check if the geometric condition is satisfied. Moreover, the APP relies on variational techniques and requires the mappings Θ and Φ to be differentiable as the derivatives Θ' and Φ' appear in Problem (4.8). However, the dynamics Θ in Problem (2.27) involves integer variables so Θ' is not defined. To overcome this difficulty, we propose a continuous relaxation of the system with relaxed cost and dynamics that are differentiable almost everywhere. It is possible to use a differentiable relaxation of the system but this requires more implementation efforts. As we are far from the conditions of convergence of the algorithm, nothing ensures that a differentiable relaxation would give better results than the simple one that is defined below.

4.3.1 State variable relaxation

Let $i \in \mathbb{I}$ and $t \in \mathbb{T}$. Recall from §2.1 that:

- $\mathbf{X}_{i,t} = (\mathbf{E}_{i,t}, \mathbf{A}_{i,t}, \mathbf{P}_{i,t})$ takes values in $\{0, 1\} \times \mathbb{R} \times (\{\delta\} \cup \mathbb{R}_+)^D$,
- \mathbf{S}_t takes values in \mathbb{N} .

We relax the integrity constraint on $\mathbf{E}_{i,t}$ and \mathbf{S}_t , so we allow:

- $\mathbf{X}_{i,t} = (\mathbf{E}_{i,t}, \mathbf{A}_{i,t}, \mathbf{P}_{i,t})$ to take values in $[0, 1] \times \mathbb{R} \times (\{\delta\} \cup \mathbb{R}_+)^D$,
- \mathbf{S}_t to take values in \mathbb{R} .

Remark 4.1. We lose the physical interpretation of the relaxed variables.

- If $0 < \mathbf{E}_{i,t} < 1$, we could think that component i is in a degraded regime where the closer $\mathbf{E}_{i,t}$ is to 1 the healthier it is. This interpretation is however not exact as the health of a component is only characterized by its age $\mathbf{A}_{i,t}$. The probability of failure of a component is indeed only a function of $\mathbf{A}_{i,t}$.
- A value $\mathbf{S}_t \in \mathbb{R}$ means that there can be a non-integer number of parts in the stock. \diamond

4.3.2 Relaxation of the dynamics

The dynamics of the original system has been described in §2.2 and an explicit expression is given in Appendix A. This expression involves indicator functions $\mathbf{1}_{\mathcal{A}}$ for some set \mathcal{A} . The dynamics is then non-continuous. Replacing the original indicator function $\mathbf{1}_{\mathcal{A}}$ with a continuous relaxed version allows to define a relaxed dynamics for the system.

Definition 4.2. Let $\mathcal{A} \subset \mathbb{R}^p$, $x \in \mathbb{R}^p$ and $\alpha > 0$. We define a continuous relaxation $\mathbf{1}_{\mathcal{A}}^\alpha$ with parameter α of the indicator function $\mathbf{1}_{\mathcal{A}}$ as:

$$\mathbf{1}_{\mathcal{A}}^\alpha(x) = \begin{cases} 1 - 2\alpha \times \text{dist}(x, \mathcal{A}) & \text{if } \text{dist}(x, \mathcal{A}) \leq \frac{1}{2\alpha}, \\ 0 & \text{if } \text{dist}(x, \mathcal{A}) > \frac{1}{2\alpha}, \end{cases} \quad (4.9)$$

where $\text{dist}(x, \mathcal{A})$ is the Euclidean distance between x and the set \mathcal{A} .

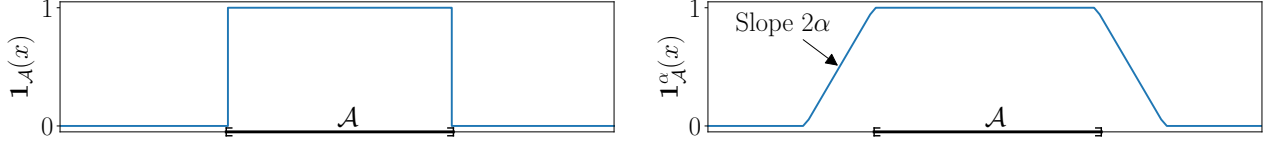


Figure 4: Illustration of the relaxation of the indicator function.

The relaxation of the indicator function is illustrated on Fig. 4.

The parameter α quantifies the stiffness of the relaxation. As $\alpha \rightarrow +\infty$, the relaxed indicator $\mathbf{1}_A^\alpha$ converges pointwise towards the original indicator function $\mathbf{1}_A$. As $\alpha \rightarrow 0$, the relaxed indicator $\mathbf{1}_A^\alpha$ converges pointwise towards the constant function 1. Note that for all $\alpha > 0$, the relaxed indicator $\mathbf{1}_A^\alpha$ is continuous and differentiable almost everywhere.

A continuous relaxation of the system dynamics of parameter α is obtained by replacing the occurrences of the indicator function by its relaxed version. For instance, the relaxed dynamics of the stock is obtained from (2.20):

$$\mathbf{S}_{t+1} = \mathbf{S}_t + \sum_{i=1}^n \sum_{d=1}^D \mathbf{1}_{\{D-1\}}^\alpha(\mathbf{P}_{i,t}^d) - \min\left(\mathbf{S}_t, \sum_{i=1}^n \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{i,t})\right). \quad (4.10)$$

Additional technical details and an explicit expression for the relaxed dynamics of the components are given in Appendix B. The relaxed dynamics and relaxed auxiliary dynamics are denoted by Θ^α and Φ^α respectively.

4.3.3 Cost relaxation

The relaxation of the maintenance cost and forced outage cost is constructed using the same technique as for the dynamics. Let $i \in \mathbb{I}$, $t \in \mathbb{T}$ and $\alpha > 0$ be given.

- The relaxed maintenance cost for component i at time t with parameter α is defined as:

$$\begin{cases} j_{i,t}^\alpha(\mathbf{X}_{i,t}, u_{i,t}) = \eta_t C_i^P u_{i,t}^2 + \eta_t C_i^C \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{i,t}) \mathbf{1}_{\{0\}}^\alpha(\mathbf{A}_{i,t}), & t \in \mathbb{T}_{-1}, \\ j_{i,T}^\alpha(\mathbf{X}_{i,T}) = \eta_T C_i^C \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{i,T}) \mathbf{1}_{\{0\}}^\alpha(\mathbf{A}_{i,T}). \end{cases} \quad (4.11)$$

- The relaxed forced outage cost at time t with parameter α is defined as:

$$j_t^{F,\alpha}(\mathbf{X}_{1:n,t}) = \eta_t C^F \min\left\{1, \sum_{i=1}^n \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{i,t}) \mathbf{1}_{\mathbb{R}_+}^\alpha(\mathbf{A}_{i,t})\right\}. \quad (4.12)$$

Similarly, as in (2.25) and (2.26), we set:

$$j_i^\alpha(\mathbf{X}_i, u_i) = \sum_{t=0}^{T-1} j_{i,t}^\alpha(\mathbf{X}_{i,t}, u_{i,t}) + j_{i,T}^\alpha(\mathbf{X}_{i,T}) \quad \text{and} \quad j^{F,\alpha}(\mathbf{X}_{1:n}) = \sum_{t=0}^{T-1} j_t^{F,\alpha}(\mathbf{X}_{1:n,t}). \quad (4.13)$$

We use the relaxed version of the dynamics Θ^α , of the auxiliary dynamics Φ^α , and of the costs j_i^α and $j^{F,\alpha}$ in the auxiliary problem (4.8). Hence, the objective function and the dynamics are continuous and differentiable almost everywhere with respect to $(\mathbf{X}, \mathbf{S}, u)$. Note that the choice of the relaxation parameter α plays an important role. We can guess that choosing a low value for α leads to a problem that may be easier to solve numerically than with a larger value of α . However, with a low value of α the relaxed dynamics and cost do not represent well the industrial problem whereas with a large α the original and relaxed dynamics are close. More details on the influence of the parameter α and how it is chosen in practice are given in §5.1.

4.4 Explicit expression of the subproblems

By construction, the auxiliary problem (4.8) can be decomposed into n independent subproblems on the components and a subproblem on the stock. In this section, we provide the explicit expression of these subproblems. The APP fixed-point algorithm will be applied on the relaxed system, so we use the relaxed dynamics and cost in the writing of the subproblems. Formally, the gradients of Θ^α and Φ^α are not defined everywhere. By abuse of notation, the subproblems are given as if Θ^α and Φ^α were differentiable. Appendix D gives details on how we handle the points where the relaxed indicator is not differentiable.

The subproblem on component $i \in \mathbb{I}$ is:

$$\begin{aligned}
\min_{(\mathbf{X}_i, u_i) \in \mathcal{X}_i \times \mathbb{U}_i} \mathbb{E} & \left(j_i^\alpha(\mathbf{X}_i, u_i) + j^{F, \alpha}(\bar{\mathbf{X}}_{1:i-1}, \mathbf{X}_i, \bar{\mathbf{X}}_{i+1:n}) \right. \\
& + \frac{\gamma_x}{2} \|\mathbf{X}_i - \bar{\mathbf{X}}_i\|^2 + \frac{\gamma_u}{2} \|u_i - \bar{u}_i\|^2 \\
& + \left\langle \bar{\Lambda}_{\mathbf{S}}, \partial_{\mathbf{X}_i} \Theta_{\mathbf{S}}^\alpha(\bar{\mathbf{X}}_{1:n}, \bar{\mathbf{S}}) \cdot \mathbf{X}_i \right\rangle \\
& + \sum_{j=i+1}^n \left\langle \bar{\Lambda}_j, \partial_{\mathbf{X}_i} \Theta_j^\alpha(\bar{\mathbf{X}}_{1:j}, \bar{\mathbf{S}}, \bar{u}_j, \mathbf{W}_j) \cdot \mathbf{X}_i \right\rangle \Big) \\
\text{s.t. } & \Phi_i^\alpha(\mathbf{X}_i, u_i, \mathbf{W}_i) = 0.
\end{aligned} \tag{4.14}$$

The subproblem on the stock is:

$$\begin{aligned}
\min_{\mathbf{S} \in \mathcal{S}} \mathbb{E} & \left(\frac{\gamma_s}{2} \|\mathbf{S} - \bar{\mathbf{S}}\|^2 + \sum_{i=1}^n \left\langle \bar{\Lambda}_i, \partial_{\mathbf{S}} \Theta_i^\alpha(\bar{\mathbf{X}}_{1:i}, \bar{\mathbf{S}}, \bar{u}_i, \mathbf{W}_i) \cdot \mathbf{S} \right\rangle \right) \\
\text{s.t. } & \Phi_{\mathbf{S}}^\alpha(\mathbf{S}) = 0.
\end{aligned} \tag{4.15}$$

4.5 The APP fixed-point algorithm for the industrial system

We can now solve the original maintenance optimization problem (2.27) using the fixed-point algorithm 1. In this section, we give details on the practical implementation of the algorithm and present an efficient mixed parallel and sequential strategy for the resolution of the subproblems, that is tailored for the industrial optimization problem.

At each iteration, we solve the auxiliary problem (4.8). In a fully parallel version of the APP fixed-point algorithm, solving the auxiliary problem (4.8) boils down to the parallel resolution of the $n + 1$ independent subproblems defined by (4.14)–(4.15).

1. The subproblems on the components (4.14) are solved with the blackbox algorithm MADS [7]. At iteration k , we solve subproblem $i \in \mathbb{I}$ with:

$$(\bar{\mathbf{X}}, \bar{\mathbf{S}}, \bar{u}) = (\mathbf{X}^k, \mathbf{S}^k, u^k) \quad \text{and} \quad \bar{\Lambda} = \Lambda^k. \tag{4.16}$$

MADS outputs a primal solution $(\mathbf{X}_i^{k+1}, u_i^{k+1})$. The optimal multiplier Λ_i^{k+1} is computed afterwards using the adjoint state. The full derivation of the backward recursion for the multipliers in the industrial case is carried out in Appendix C.

2. The subproblem on the stock (4.15) is very easy to solve numerically. At iteration k of the fully parallel APP fixed-point algorithm, we use:

$$(\bar{\mathbf{X}}, \bar{\mathbf{S}}, \bar{u}) = (\mathbf{X}^k, \mathbf{S}^k, u^k) \quad \text{and} \quad \bar{\Lambda} = \Lambda^k. \tag{4.17}$$

The constraint $\Phi_{\mathcal{S}}^{\alpha}(\mathbf{S}) = 0$ represents the dynamics of the stock with $\bar{\mathbf{X}} = \mathbf{X}^k$ being fixed. The value of $\bar{\mathbf{X}}$ completely determines the dynamics of the stock. Hence, solving the subproblem on the stock just boils down to simulate its dynamics, we get a primal solution \mathbf{S}^{k+1} . The optimal multiplier $\mathbf{\Lambda}_{\mathcal{S}}^{k+1}$ is also computed using the adjoint state, see Appendix C.

The features of the subproblem on the stock suggest to change the fully parallel strategy into a mixed parallel/sequential strategy.

1. At iteration k , the n subproblems on the components are still solved in parallel using:

$$(\bar{\mathbf{X}}, \bar{\mathbf{S}}, \bar{u}) = (\mathbf{X}^k, \mathbf{S}^k, u^k) \quad \text{and} \quad \bar{\mathbf{\Lambda}} = \mathbf{\Lambda}^k. \quad (4.18)$$

This yields a solution $(\mathbf{X}_i^{k+1}, u_i^{k+1}, \mathbf{\Lambda}_i^{k+1})$ for each subproblem $i \in \mathbb{I}$.

2. The difference arises for the subproblem on the stock. At iteration k , we immediately use the output of the subproblems on the components. This means that we set before solving the subproblem on the stock at iteration k :

$$(\bar{\mathbf{X}}, \bar{u}) = (\mathbf{X}^{k+1}, u^{k+1}) \quad \text{and} \quad (\bar{\mathbf{\Lambda}}_1, \dots, \bar{\mathbf{\Lambda}}_n) = (\mathbf{\Lambda}_1^{k+1}, \dots, \mathbf{\Lambda}_n^{k+1}). \quad (4.19)$$

This implies that the subproblem on the stock at iteration k can be solved only after all subproblems on the components have been solved.

With this strategy, we see experimentally that the number of iterations for convergence is reduced without penalizing the computation time per iteration as the subproblem on the stock can be solved in negligible time. This results in an overall speed up of the algorithm. The APP fixed-point algorithm with the mixed parallel/sequential strategy is presented in Algorithm 2. The termination criterion is a maximum number of iterations $M \in \mathbb{N}$. This is the version that is used for the numerical experiments.

Algorithm 2 APP fixed-point algorithm with a mixed parallel/sequential strategy

- 1: Start with $(\bar{\mathbf{X}}, \bar{\mathbf{S}}, \bar{u}) = (\mathbf{X}^0, \mathbf{S}^0, u^0)$, $\bar{\mathbf{\Lambda}} = \mathbf{\Lambda}^0$
 - 2: **for** $k = 0, \dots, M - 1$ **do**:
 - 3: **for all** $i \in \{1, \dots, n\}$ **do in parallel**:
 - 4: Solve the subproblem (4.14) on component i .
 - 5: Let $(\mathbf{X}_i^{k+1}, u_i^{k+1})$ be a solution and $\mathbf{\Lambda}_i^{k+1}$ be an optimal multiplier.
 - 6: **end for**
 - 7: Set $(\bar{\mathbf{X}}, \bar{u}) = ((\mathbf{X}_1^{k+1}, \dots, \mathbf{X}_n^{k+1}), (u_1^{k+1}, \dots, u_n^{k+1}))$
 - 8: Set $(\bar{\mathbf{\Lambda}}_1, \dots, \bar{\mathbf{\Lambda}}_n) = (\mathbf{\Lambda}_1^{k+1}, \dots, \mathbf{\Lambda}_n^{k+1})$
 - 9: Solve the subproblem (4.15) on the stock.
 - 10: Let \mathbf{S}^{k+1} be a solution and $\mathbf{\Lambda}_{\mathcal{S}}^{k+1}$ be an optimal multiplier.
 - 11: Set $\bar{\mathbf{S}} = \mathbf{S}^{k+1}$ and $\bar{\mathbf{\Lambda}}_{\mathcal{S}} = \mathbf{\Lambda}_{\mathcal{S}}^{k+1}$
 - 12: **end for**
 - 13: **return** the maintenance strategy u^M
-

5 Numerical results

In this part, we present the results of the decomposition methodology applied to Problem (2.27). The expectation in (2.27) cannot be evaluated exactly, so we solve a Monte-Carlo approximation of

the problem with Q fixed failure scenarios $\omega_1, \dots, \omega_Q \in \Omega$:

$$\begin{aligned} \min_{(\mathbf{X}, \mathbf{S}, u) \in \mathcal{X} \times \mathcal{S} \times \mathbb{U}} \quad & \frac{1}{Q} \sum_{q=1}^Q \left(\sum_{i=1}^n j_i(\mathbf{X}_i(\omega_q), u_i) + j^F(\mathbf{X}_{1:n}(\omega_q)) \right) \\ \text{s.t.} \quad & \Theta(\mathbf{X}(\omega_q), \mathbf{S}(\omega_q), u, \mathbf{W}(\omega_q)) = 0, \quad q \in \{1, \dots, Q\}. \end{aligned} \quad (5.1)$$

The aim of this work is to show that the decomposition by prediction can efficiently solve a maintenance problem with a large number of components. We consider two test cases with the characteristics given in Table 1. As a maintenance strategy is parametrized by the vector $u \in \mathbb{U}$ given in (2.8), the optimization problem for the 80-component case involves $nT = 80 \times 40 = 3200$ variables.

The synthetic systems described in Table 1 do not exactly represent a real situation but are meant to be a proxy for the real cases presented at the beginning of §2.1:

1. The situation where we consider turbines of the same production unit is simpler than the synthetic systems as it has fewer components (up to 10 for the real case and 80 in the synthetic case).
2. On the other hand, the situation where we consider turbines across several production units is slightly more complicated than the synthetic systems: in our model all components belong to the same unit, therefore the forced outage cost is simpler than in reality. However, the decomposition methodology can be applied to this real case exactly as for the synthetic cases.
3. For the case of a system with several families of components with a stock of spare parts for each family, the decomposition methodology is still applicable and will result in an additional subproblem for each stock. However, we have seen in §4.5 that these subproblems are solved in negligible time as they just amount to the simulation of the dynamics of the corresponding stock. Hence, these additional subproblems do not slow down the method compared to the synthetic cases. The expression of the forced outage cost may also be slightly more complicated than for the synthetic cases.

The costs and the failure distributions used in Table 1 are of the same order of magnitude as what is encountered in practice. In the first test case, the failure distributions are chosen so that the mean time to failure of the components is in the lower limit of the real cases, which induces a large number of failures in the system. This is the situation where the PM strategy has the most significant influence on the economic performance of the system. In the second test case, we consider components with a longer lifespan but the stock has fewer parts than in the first case. This induces a strong coupling between the stock and the components. These cases can be considered to be stress tests for the optimization algorithms.

5.1 Parameter tuning

Several parameters have to be tuned in order to apply the APP fixed-point algorithm. The parameters $\gamma_x, \gamma_s, \gamma_u$ appear in the auxiliary problem and α characterizes the relaxation of the system. These parameters are used in the APP fixed-point algorithm but not in the reference algorithm as the latter uses the original dynamics of the system to perform the optimization.

As the maintenance threshold ν is used both in the reference algorithm and the decomposition method, it is fixed to a value giving good performance with the reference algorithm. The same value of ν is used in the APP fixed-point algorithm. We do not consider changing the value of ν as it would mean that the reference algorithm changes for each different value of ν , making a fair comparison harder and the results less clear to analyze.

Parameter	Case 1	Case 2
Number of components n	80	80
Initial number of spare parts \mathbf{S}_0	16	5
Horizon T	40 years	40 years
Time step Δt	1 year	1 year
Number of time steps for supply D	2	2
Discount rate τ	0.08	0.08
Maintenance threshold ν	0.9	0.9
Yearly forced outage cost C^F	10000 k€/ year	10000 k€/ year
Number of failure scenarios Q	100	300
Components characteristics		
PM cost C^P	50 k€	50 k€
CM cost C^C	200 k€	200 k€
Failure distribution	Weib(3, 10)	Weib(3, 20)
Mean time to failure	8.93 years	17.86 years

Table 1: Characteristics of the test cases. Bold data represent the features that differ between the two cases.

5.1.1 Description of the parameters

In the auxiliary problem (4.8), the value of $\gamma = (\gamma_x, \gamma_s, \gamma_u)$ influences the numerical behavior of the algorithm. We choose to increase the values of γ_x, γ_s and γ_u at each iteration of the APP fixed-point algorithm. The insight is that we can use low values of γ in the first iterations to get close to a good solution. Then, we use high values of γ to avoid oscillations of the solution of the auxiliary problem. Indeed, with a large value of γ , the solution $(\mathbf{X}^{k+1}, \mathbf{S}^{k+1}, u^{k+1})$ of the auxiliary problem at iteration $k+1$ is close to the previous solution $(\mathbf{X}^k, \mathbf{S}^k, u^k)$. The value of γ_u evolves from iteration k to $k+1$ of the APP fixed-point algorithm with an additive step $\Delta\gamma > 0$, so that:

$$\gamma_u^{k+1} = \gamma_u^k + \Delta\gamma. \quad (5.2)$$

Then, γ_x and γ_s are chosen to be proportional to γ_u with ratios $r_x > 0$ and $r_s > 0$ respectively, so that:

$$\gamma_x^{k+1} = \gamma_u^{k+1}/r_x \quad \text{and} \quad \gamma_s^{k+1} = \gamma_u^{k+1}/r_s. \quad (5.3)$$

The motivation for this choice is that the vectors \mathbf{X} , \mathbf{S} and u need some rescaling so that their norms are of the same order of magnitude. The parameters $\gamma_u^0, r_x, r_s, \Delta\gamma$ have to be tuned.

The other parameter that requires attention is the relaxation parameter α . Similarly as for γ , we choose to increase the value of α at each iteration. A low value of α makes the problem easier to solve numerically but does not represent well the real problem. As α increases, the relaxed problem is closer and closer to the real one but becomes harder to solve. To ease the resolution, we use the solution of the auxiliary problem at iteration k as a warm start in MADS for iteration $k+1$. The value of α varies from iteration k to $k+1$ of the APP fixed-point algorithm with a step $\Delta\alpha > 0$ so that:

$$\alpha^{k+1} = \alpha^k + \Delta\alpha. \quad (5.4)$$

The values of α^0 and $\Delta\alpha$ have to be tuned. We denote by:

$$p = (\gamma_u^0, r_x, r_s, \Delta\gamma, \alpha^0, \Delta\alpha) \in \mathbb{R}^6, \quad (5.5)$$

the vector of parameters that have to be adjusted for the algorithm.

5.1.2 Tuning methodology

Choosing a good value for p is difficult in practice. We cannot afford to test the fixed-point algorithm with many different values of p directly on the 80-component case, as one run of the optimization takes more than a day. In order to find an appropriate value for p , we rely on a smaller system than the 80-component case. This *small system* is designed so that the runs of the fixed-point algorithm take a reasonable amount of time, 4 hours in our case. The small system consists of 10 components with 2 spare parts initially. All the other characteristics of this small system are the same as Case 1 in Table 1. In a way, the 10-component system can be seen as a downscaling of the 80-component case of interest.

The idea of the tuning procedure is to run the decomposition method on the 10-component system several times, but with a different value for p at each run. To do so, we start by defining bounds on the value of the parameters, they are given in Table 2. These bounds are chosen to be wide in order to ensure that they contain good values of the parameters. Then, 200 samples of p are drawn

Parameter	γ_u^0	r_x	r_s	$\Delta\gamma$	α^0	$\Delta\alpha$
Bounds	[1, 100]	[1, 10 ⁴]	[1, 10 ³]	[0, 100]	[2, 200]	[0, 200]

Table 2: Bounds for the parameters of the of the fixed-point algorithm.

with an optimized Latin Hypercube Sampling [15]. We run the fixed-point algorithm with each of the sampled values on the 10-component system. The value of p which gives the best result on the small system is used for the case with 80 components. The chosen value is:

$$p = (17.32, 7434, 815.3, 1.360 \times 10^{-1}, 46.51, 135.5). \quad (5.6)$$

We give the value of p with 4 significant digits to emphasize that the performance of the APP fixed-point algorithm is very sensitive to this value. A sensitivity analysis has been performed using a Morris method [9] but no clear pattern for a good choice of p has been identified. More details on the methodology and results of the sensitivity analysis can be found in [8, §10.7].

5.2 Comparison of the decomposition method with MADS

Now that all the parameters of the fixed-point algorithm 2 are set, we can solve Problem (5.1). The performance of the fixed-point algorithm is compared with a reference algorithm, which is the blackbox algorithm MADS applied directly on Problem (5.1). We consider the systems of 80 components described in Table 1. Parameters of the computation are given in Table 3. When running the optimization, the reference algorithm uses the original dynamics as it does not use gradient information. The APP fixed-point algorithm runs with the value of p in (5.6) that parametrizes the auxiliary problem and the relaxation of the system.

	Decomposition	MADS
Fixed-point iterations	50	/
Cost function calls	10 ³ /subproblem/iteration	10 ⁶
Cost and dynamics	Relaxed	Original
Processor model	Intel® Xeon® Processor E5-2680 v4, 2.4 GHz	

Table 3: Parameters of the computation for the two algorithms.

Remark 5.1. The APP fixed-point algorithm solves a decomposable auxiliary problem at each iteration, this algorithm is designed to be parallelized. It runs on 80 processors so that the subproblems on the components are solved in parallel. The reference algorithm MADS runs only on one processor. Note that it is also possible to parallelize MADS [6], although the implementation is not as straightforward as for the decomposition method. The parallel version of MADS has not been tested. \diamond

The maintenance strategies returned by the two algorithms are then evaluated on a common set of 10^5 failure scenarios, distinct from those used for the optimization. For the two strategies, the evaluation is done with the original dynamics of the system in order to ensure a fair comparison. The two algorithms return a maintenance strategy with $u_{i,t} \in [0, 1]$ for $(i, t) \in \mathbb{I} \times \mathbb{T}_{-1}$. From the operational perspective, PMs make the components as good as new. Hence, for the evaluation of the strategy, the controls are projected on $\{0, 1\}$: we consider that if $u_{i,t} \geq \nu$, then the PM makes the component as good as new, otherwise no PM is performed. The comparison between the two maintenance strategies is fair as we use the same procedure for their evaluation.

5.2.1 Results for Case 1

The mean cost is 14509 k€ with MADS and 12855 k€ with the decomposition which represents a gain of 12%. The values of some quantiles are gathered in Table 4 and the distribution of the cost is represented on Fig. 5. Figure 6 outlines that both the CM cost and the PM cost are

	1%	5%	25%	50%	75%	95%	99%
Decomposition	11641	11981	12469	12838	13221	13788	14182
MADS	13339	13663	14138	14494	14846	15406	15817

Table 4: Quantiles of the cost of the two maintenance strategies (k€).

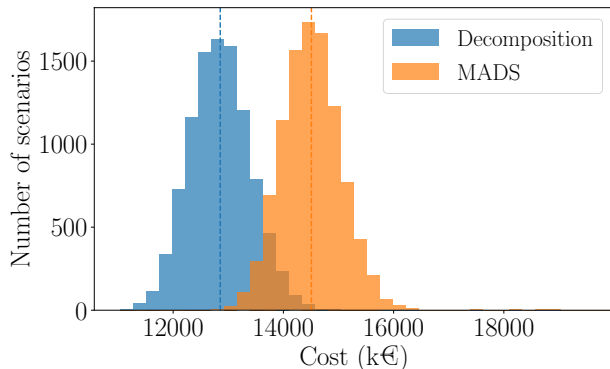


Figure 5: Distribution of the cost for the two maintenance strategies. The dashed lines represent the expected cost for both strategies.

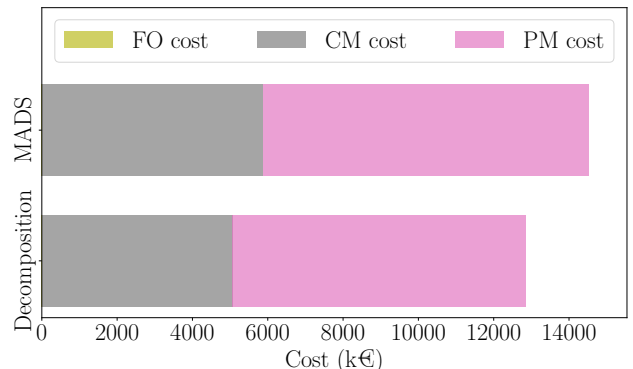


Figure 6: Part of the PM, CM and forced outage cost in the total expected cost.

lower with the decomposition strategy. In fact the decomposition strategy performs more PMs than MADS strategy (Table 5) but these PMs occur later in time (Fig. 7). Because of the discount factor, the decomposition strategy is cheaper even if more PMs are performed. The forced outage cost is not visible on Fig. 6 as it represents 8.7 k€ for MADS strategy and 2.7 k€ for the decomposition strategy: there are very few scenarios where a forced outage occurs, again with an advantage for the decomposition strategy (6 occurrences in 10^5 failure scenarios for the decomposition versus 36 for

MADS). Overall, the decomposition strategy performs PMs more effectively than MADS, both in terms of cost and in terms of number of failures and forced outages of the system.

Table 5 also provides the computation time for both methods. As we design a long term maintenance schedule, the focus is on the performance of the maintenance strategy and not on the running time of the methods that are acceptable for our purpose. The parameters of the algorithms have been tuned to allow similar numbers of function calls for both methods. The decomposition is slower as it uses the relaxed dynamics which is slower to compute than the original dynamics used by MADS.

	Decomposition	MADS
Total number of PMs	669	595
Mean number of PMs / component	8.4	7.4
Mean number of failures / component	1.18	1.48
Number of forced outages / Number of scenarios	6/10000	36/10000
Computation time	2 days, 1 hour	1 day, 7 hours

Table 5: Number of PMs, failures and forced outages, and computation time for each strategy.

The cumulative number of PMs can be visualized on Fig. 7. A striking feature with the decomposition strategy is that there are almost no PM in the first three years. This exploits the fact that the components are new. The reference algorithm MADS applied directly on the original problem does not detect this feature. In fact, the region of the space corresponding to not doing any PM in first three years jointly for all components is a very small subset of the admissible space of the original problem and is not explored by MADS. On the other hand, the subproblems in the APP fixed-point algorithm act on an individual component, it is then easier to figure out that doing no PM in the first three years is profitable.

On the second half of the period of study, the decomposition strategy performs more PMs than MADS. Because of the discount factor, these PMs are cheaper than PMs performed in the first years. Thus, the decomposition strategy manages to avoid more failures than MADS strategy in the second half of the horizon. We see that the decomposition strategy has a time-dependent behavior which is not the case for MADS strategy that always performs PMs at the same rate. The design of an effective time-dependent strategy is only possible with the decomposition as it solves subproblems of dimension 40 on individual components. The algorithm MADS applied on the whole problem cannot detect these features as the search space of dimension 3200 is too large.

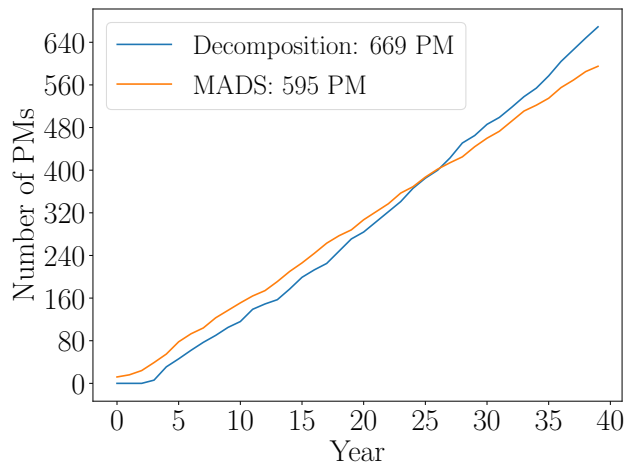


Figure 7: Cumulative number of PMs.

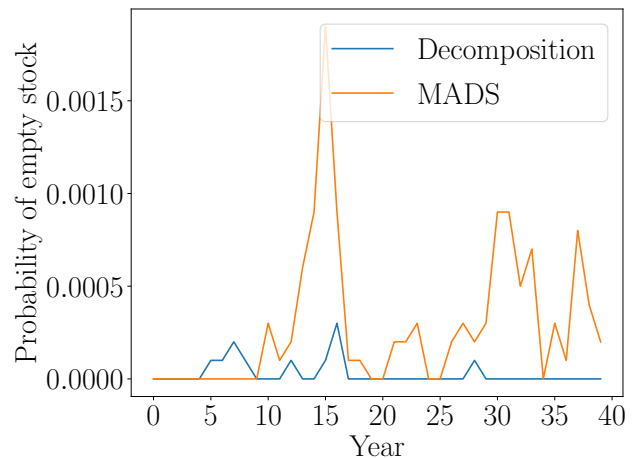


Figure 8: Evolution of the probability of having an empty stock.

Another indicator that is monitored by decision makers is the level of stock. A necessary condition for the occurrence of a forced outage is that the stock is empty. Hence, we look at the probability of having an empty stock. The higher this probability, the higher the probability of forced outage. The probability of having an empty stock is very low on the whole horizon for the decomposition strategy (Fig. 8). For MADS strategy this probability is often higher in particular around year 15 and after year 30. This feature again supports that the the decomposition strategy is more robust than MADS strategy.

Overall on this case, the strategy obtained by decomposition is more cost effective than MADS strategy. In the case of extreme events, the decomposition strategy is more robust than MADS strategy, as shown by the 99% quantile in Table 4, the number of failures and of forced outages in Table 5 and the probability of having an empty stock in Fig. 8.

5.2.2 Results for Case 2

In the second case presented in Table 1, the components have a longer lifespan but the stock has fewer parts than in the first case. The coupling between the components and the stock is then more important than in Case 1. The mean cost of the maintenance strategy returned by MADS is 10815 k€ and it is 9749 k€ for the strategy returned by the decomposition, which represents a gain of 10%. The values of some quantiles are gathered in Table 6 and the distribution of the cost is represented on Fig. 9. The situation is quite different from the first case as the decomposition strategy has a more important variance than MADS strategy and presents several modes.

	1%	5%	25%	50%	75%	95%	99%
Decomposition	5381	5601	5999	9054	13014	18421	23069
MADS	10221	10357	10568	10733	10916	11266	13650

Table 6: Quantiles of the cost of the two maintenance strategies (k€).

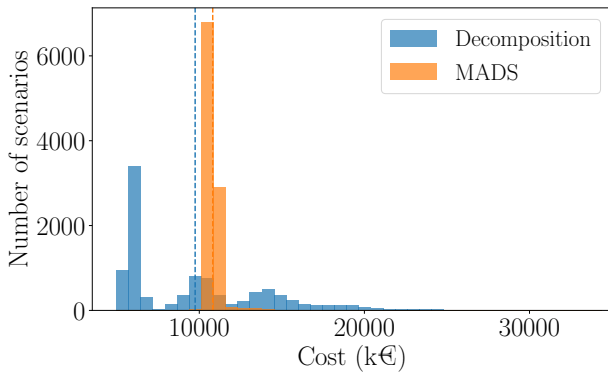


Figure 9: Distribution of the cost for the two maintenance strategies. The dashed lines represent the expected cost for both strategies.

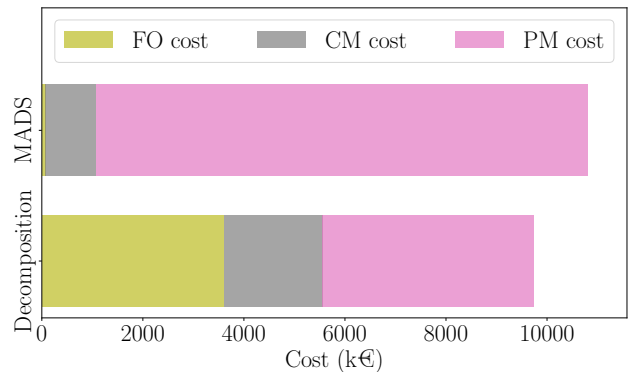


Figure 10: Part of the PM, CM and forced outage cost in the total expected cost.

The repartition of the cost in Fig. 10 confirms that the two maintenance strategies have a different approach. The strategy returned by MADS is similar than in the first case with a high number of PMs, few failures and forced outages, see Table 7. On the other hand, with the decomposition strategy the PM cost is much lower, as fewer PMs are performed (Table 7). The counterpart is a higher CM cost and a notable contribution of the forced outage cost. We see in Table 7 that the number of forced outages is indeed significantly higher with the decomposition strategy, with almost

one forced outage per scenario in average. Thus, the approaches of the two maintenance strategies are completely different: the MADS strategy is conservative with a lot of PMs (similarly as in the first case) whereas the decomposition strategy is more risky with fewer PMs and more forced outages.

	Decomposition	MADS
Total number of PMs	503	644
Mean number of PMs / component	6.3	8.1
Mean number of failures / component	0.38	0.26
Number of forced outages / scenario	0.94	0.04
Computation time	5 days, 12 hours	3 days, 12 hours

Table 7: Number of PMs, failures and forced outages, and computation time for each strategy.

The cumulative number of PMs can be visualized on Fig. 11. Similarly as in the first case, we see a time-dependent behavior of the decomposition strategy with almost no maintenance in the first seven years. In Case 1, the decomposition strategy does not perform PMs in the first three years only but the components have a shorter average lifespan (around 9 years in Case 1 and 18 years in Case 2). In the first half of the time horizon, the decomposition strategy performs fewer PMs than MADS strategy, whereas in the second half of the horizon the decomposition performs slightly more PMs than MADS. We see in Fig. 12, that the decomposition strategy results in a high probability of having an empty stock between years 10 and 15, where most of the forced outages occur. In the first years and in the second half of the horizon, both strategies present a low risk of having an empty stock.

In Table 7, we see that the computation time are around 3 times larger than in Case 1. This is due to the fact that we have used $Q = 300$ scenarios here against $Q = 100$ scenarios in Case 1 for the estimation of the expected cost (see Table 1). As the decomposition designs a risky strategy, if Q is kept too low, the maintenance strategy may not perform well when evaluated on the 10^5 validation scenarios as extreme events are not taken into account during the optimization.

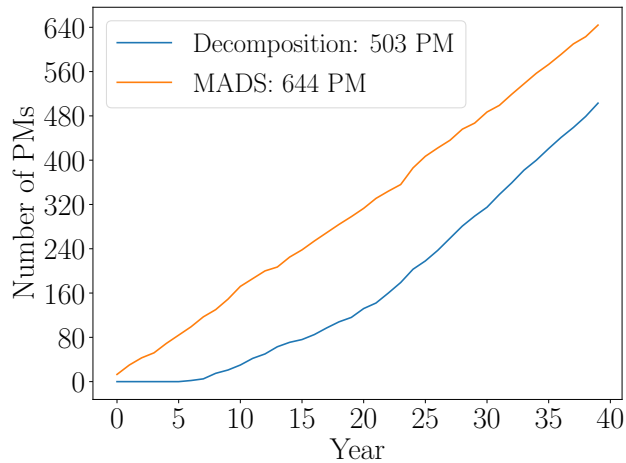


Figure 11: Cumulative number of PMs.

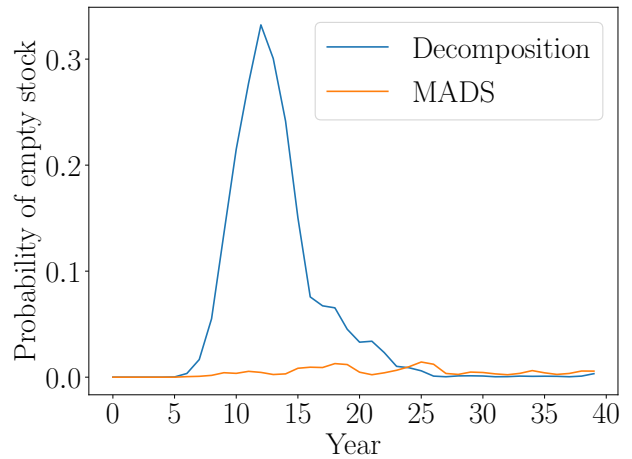


Figure 12: Evolution of the probability of having an empty stock.

Overall for this case, the decomposition strategy performs better in expectation, which is the criterion that the problem has been designed to optimize. However, the decomposition strategy has a higher variance than MADS strategy with an important risk of forced outage. Thus, from an operational point of view, a decision maker will probably choose the MADS strategy. Our work

is meant to be a decision support tool and is not design to take the final operational decision. The analysis of Fig. 11 and 12 suggest to investigate a maintenance strategy that is similar to the decomposition strategy in the first years and in the second half of the horizon but with more PMs between years 10 and 15 to reduce the number of forced outages. We have the insight that this new strategy could improve the robustness of the decomposition strategy and perform similarly or even better in expectation.

6 Conclusion

In this work we study a maintenance scheduling optimization problem for hydropower plants management. We set up a decomposition method to find a deterministic preventive maintenance strategy for a system of physical components sharing a common stock of spare parts. The decomposition relies on the Auxiliary Problem Principle. We construct a sequence of auxiliary problems that are solved iteratively. The auxiliary problems are decomposable into independent subproblems of smaller dimension that are solved in parallel. Each subproblem involves only one component of the system or the stock. A relaxation of the system is necessary as the gradients of the dynamics and the cost are required to compute the coordination terms. Then, the fixed-point algorithm is implemented with an efficient mixed parallel/sequential strategy. This implementation is particularly adapted to the structure of the industrial problem, for which the subproblem on the stock is easy to solve numerically.

On the two industrial test cases, the decomposition method outperforms in expectation the black-box algorithm MADS applied directly on the full problem. The reason is that when MADS is used to solve subproblems of small dimension within the decomposition algorithm, it manages to detect features to design an efficient time-dependent PM strategy. These features – such as not replacing components in the first years of the horizon – cannot be detected when MADS is applied directly on the full problem as the corresponding strategies only represent a very small subset of the search space which is not explored by the algorithm.

The approach of the decomposition strategy differs between the two test cases. In the first case the strategy is conservative and robust to extreme events whereas in the second test case, it is more risky as more forced outages can occur. Our work is meant to be a decision support tool and even if it is unlikely that a decision maker chooses a risky maintenance strategy, the results on the second case can give insights for the design of an efficient robust maintenance strategy.

This work proves the interest of the modeling effort needed to apply the decomposition method. Some challenges still remain for an application in an operational context. In order to force the design of a robust maintenance strategy, some risk criteria should be added in the mathematical formulation of the problem, either in the cost function or in the constraint. Another point that must be noticed is that the dynamics is simulated with a time step of one year. A smaller time step must be used for an accurate evaluation of the costs. This will not increase the complexity of the problem as maintenance decisions are always made on a yearly basis, so that the space of admissible maintenance strategies is still of the same dimension. However, the time needed for the evaluation of the cost function will increase. It is also possible to model more complex systems, by adding a control on the time of the order of spare parts or dependence between the failures of the components for instance. We could also consider imperfect preventive maintenance. A balance must be found between the simplicity of the model and its adequation to reality given the industrial application in mind.

A Explicit expression of the dynamics of the original system

In this part, we give an explicit expression for the dynamics f_i of component $i \in \mathbb{I}$ that appears in (2.19). We can write:

$$\mathbf{X}_{i,t+1} = \begin{pmatrix} \mathbf{E}_{i,t+1} \\ \mathbf{A}_{i,t+1} \\ \mathbf{P}_{i,t+1} \end{pmatrix} = \begin{pmatrix} f_{i,\mathbf{E}}(\mathbf{X}_{1:i,t}, \mathbf{S}_t, u_{i,t}, \mathbf{W}_{i,t+1}) \\ f_{i,\mathbf{A}}(\mathbf{X}_{1:i,t}, \mathbf{S}_t, u_{i,t}, \mathbf{W}_{i,t+1}) \\ f_{i,\mathbf{P}}(\mathbf{X}_{1:i,t}, \mathbf{S}_t, u_{i,t}, \mathbf{W}_{i,t+1}) \end{pmatrix}, \quad (\text{A.1})$$

so that $f_i = (f_{i,\mathbf{E}}, f_{i,\mathbf{A}}, f_{i,\mathbf{P}})$. We give an explicit formula for $f_{i,\mathbf{E}}$, $f_{i,\mathbf{A}}$ and $f_{i,\mathbf{P}}$.

A.1 Dynamics of the regime $\mathbf{E}_{i,t}$

Using Fig. 3, we can write:

$$\begin{aligned} \mathbf{E}_{i,t+1} &= f_{i,\mathbf{E}}(\mathbf{X}_{1:i,t}, \mathbf{S}_t, u_{i,t}, \mathbf{W}_{i,t+1}) \\ &= \mathbf{1}_{\mathbb{R}_+}(\mathbf{S}_t - \sum_{j=1}^i \mathbf{1}_{\{0\}}(\mathbf{E}_{j,t})) \mathbf{1}_{\{0\}}(\mathbf{E}_{i,t}) \\ &\quad + (\mathbf{1}_{\mathbb{R}_+}(u_{i,t} - \nu) + \mathbf{1}_{\mathbb{R}_+}(\mathbf{W}_{i,t+1} - p_i(\mathbf{A}_{i,t})) \mathbf{1}_{\mathbb{R}_+^*}(\nu - u_{i,t})) \mathbf{1}_{\{1\}}(\mathbf{E}_{i,t}). \end{aligned} \quad (\text{A.2})$$

The first part of (A.2) means that if the component is broken at t and we have enough spares to repair it, it is then functioning at $t+1$. The second part means that if the component is functioning at t and we do a PM, it is still functioning at $t+1$. Finally, if we do not do a PM, the regime depends on the occurrence of a failure between t and $t+1$.

A.2 Dynamics of the age $\mathbf{A}_{i,t}$

Again using Fig. 3, we can write:

$$\begin{aligned} \mathbf{A}_{i,t+1} &= f_{i,\mathbf{A}}(\mathbf{X}_{1:i,t}, \mathbf{S}_t, u_{i,t}, \mathbf{W}_{i,t+1}) \\ &= (\mathbf{A}_{i,t} + 1) \left(\mathbf{1}_{\mathbb{R}_+^*} \left(\sum_{j=1}^i \mathbf{1}_{\{0\}}(\mathbf{E}_{j,t}) - \mathbf{S}_t \right) \mathbf{1}_{\{0\}}(\mathbf{E}_{i,t}) \right. \\ &\quad \left. + \mathbf{1}_{\mathbb{R}_+}(\mathbf{W}_{i,t+1} - p_i(\mathbf{A}_{i,t})) \mathbf{1}_{\mathbb{R}_+^*}(\nu - u_{i,t}) \mathbf{1}_{\{1\}}(\mathbf{E}_{i,t}) \right) \\ &\quad + \mathbf{1}_{\mathbb{R}_+}(\mathbf{S}_t - \sum_{j=1}^i \mathbf{1}_{\{0\}}(\mathbf{E}_{j,t})) \mathbf{1}_{\{0\}}(\mathbf{E}_{i,t}) + \left((1 - u_{i,t}) \mathbf{A}_{i,t} + 1 \right) \mathbf{1}_{\mathbb{R}_+}(u_{i,t} - \nu) \mathbf{1}_{\{1\}}(\mathbf{E}_{i,t}). \end{aligned} \quad (\text{A.3})$$

If the component is broken at t , it stays broken if there are not enough spares in the stock (first line of (A.3)). In this case the time $\mathbf{A}_{i,t}$ increases by 1. If the component is healthy at t , it ages if no PM is done and no failure occurs (second line of (A.3)). When the component is broken and that there are enough spares (first term in the third line of (A.3)), a CM is performed and we have $\mathbf{A}_{i,t+1} = 1$. In the case of a PM (second term in the third line of (A.3)), the component is rejuvenated and its age becomes $\mathbf{A}_{i,t+1} = (1 - u_{i,t}) \mathbf{A}_{i,t} + 1$. If there is a failure, we have $\mathbf{A}_{i,t+1} = 0$.

A.3 Dynamics of the vector of times since last failures $\mathbf{P}_{i,t}$

The expression of the dynamics of $\mathbf{P}_{i,t} = (\mathbf{P}_{i,t}^1, \dots, \mathbf{P}_{i,t}^D)$ is more complex. We write:

$$\mathbf{P}_{i,t+1} = \begin{pmatrix} \mathbf{P}_{i,t+1}^1 \\ \vdots \\ \mathbf{P}_{i,t+1}^D \end{pmatrix} = \begin{pmatrix} f_{i,\mathbf{P}}^1(\mathbf{X}_{1:i,t}, \mathbf{S}_t, u_{i,t}, \mathbf{W}_{i,t+1}) \\ \vdots \\ f_{i,\mathbf{P}}^D(\mathbf{X}_{1:i,t}, \mathbf{S}_t, u_{i,t}, \mathbf{W}_{i,t+1}) \end{pmatrix} = f_{i,\mathbf{P}}(\mathbf{X}_{1:i,t}, \mathbf{S}_t, u_{i,t}, \mathbf{W}_{i,t+1}), \quad (\text{A.4})$$

so that $f_{i,\mathbf{P}} = (f_{i,\mathbf{P}}^1, \dots, f_{i,\mathbf{P}}^D)$. We give the expression of $f_{i,\mathbf{P}}^d$ for $d \in \{1, \dots, D\}$:

$$\begin{aligned} \mathbf{P}_{i,t+1}^d &= \left((\mathbf{P}_{i,t}^d + 1) \mathbf{1}_{\mathbb{R}_+}(\mathbf{P}_{i,t}^d) + \delta \mathbf{1}_{\{\delta\}}(\mathbf{P}_{i,t}^d) \right) \left(1 - \mathbf{1}_{\{1\}}(\mathbf{E}_{i,t}) \mathbf{1}_{\{0\}}(\mathbf{E}_{i,t+1}) \right) \\ &\quad + \left((\mathbf{P}_{i,t}^d + 1) \mathbf{1}_{\mathbb{R}_+}(\mathbf{P}_{i,t}^d) \mathbf{1}_{\{\delta\}}(\mathbf{P}_{i,t}^D) + \delta \mathbf{1}_{\{\delta\}}(\mathbf{P}_{i,t}^{d-1}) \mathbf{1}_{[2,D]}(d) \right. \\ &\quad \left. + (\mathbf{P}_{i,t}^{d+1} + 1) \mathbf{1}_{\mathbb{R}_+}(\mathbf{P}_{i,t}^D) \mathbf{1}_{[1,D-1]}(d) \right) \mathbf{1}_{\{1\}}(\mathbf{E}_{i,t}) \mathbf{1}_{\{0\}}(\mathbf{E}_{i,t+1}). \end{aligned} \quad (\text{A.5})$$

The first line represents the case where there is no failure. Then $\mathbf{P}_{i,t}$ increases by one if it is different from δ , otherwise it keeps the value δ . When there is a failure, if component i has undergone fewer than D failures, the evolution of $\mathbf{P}_{i,t}$ is described by (2.16a). This case is represented by the second line of (A.5). When the component has already undergone D failures, the evolution of $\mathbf{P}_{i,t}$ is described by (2.17a). This case is represented by the third line of (A.5). Note that this expression of $\mathbf{P}_{i,t+1}^d$ depends on $\mathbf{E}_{i,t+1}$. It is possible to express $\mathbf{P}_{i,t+1}^d$ only with variables describing component i at time t , this can be done by replacing $\mathbf{E}_{i,t+1}$ by its expression (A.2).

A.4 Dynamics of the stock \mathbf{S}_t

We recall the explicit dynamics of the stock that is already given in (2.20):

$$\mathbf{S}_{t+1} = \mathbf{S}_t + \sum_{i=1}^n \sum_{d=1}^D \mathbf{1}_{\{D-1\}}(\mathbf{P}_{i,t}^d) - \min \left\{ \mathbf{S}_t, \sum_{i=1}^n \mathbf{1}_{\{0\}}(\mathbf{E}_{i,t}) \right\}. \quad (\text{A.6})$$

The dynamics of the whole system has now been explicitly described.

B Explicit expression of the dynamics of the relaxed system

The expression of the relaxed dynamics of parameter α is obtained from Equations (A.2), (A.3), (A.5) and (A.6) by replacing the indicator function with its relaxed version.

We do not always substitute directly the indicator with its relaxation. The dynamics often involves conditions on complementary events. For example, the condition *if the component is broken* is represented by $\mathbf{1}_{\{0\}}(\mathbf{E}_{i,t})$. On the other hand, the condition *if the component is healthy* is represented by $\mathbf{1}_{\{1\}}(\mathbf{E}_{i,t})$. For the original dynamics as $\mathbf{E}_{i,t} \in \{0, 1\}$ we always have

$$\mathbf{1}_{\{0\}}(\mathbf{E}_{i,t}) + \mathbf{1}_{\{1\}}(\mathbf{E}_{i,t}) = 1. \quad (\text{B.1})$$

This relation is not true anymore using directly the relaxed version of the indicator with the relaxed variables. Take for example $\alpha = 2$, and suppose $\mathbf{E}_{i,t} = \frac{1}{2}$, then

$$\mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{i,t}) + \mathbf{1}_{\{1\}}^\alpha(\mathbf{E}_{i,t}) = 0. \quad (\text{B.2})$$

If we replace directly all indicator functions by their relaxation, the consequence would be in this case that $\mathbf{E}_{i,t+1} = 0$ no matter the control $u_{i,t}$. This means that even if we do a PM with $u_{i,t} = 1$, the

component is down at $t + 1$. This does not represent the dynamics of the system as we would expect. To design a coherent relaxed dynamics, complementary conditions $\mathbf{1}_{\mathcal{A}}$ and $\mathbf{1}_{\mathcal{A}^c}$ are represented using the relaxed version $\mathbf{1}_{\mathcal{A}}^\alpha$ of the indicator function for the first condition and the function $1 - \mathbf{1}_{\mathcal{A}}^\alpha$ for the complementary condition.

B.1 Relaxed dynamics of the regime $E_{i,t}$

The relaxed dynamics of the regime of parameter $\alpha > 0$ is given by:

$$\begin{aligned} E_{i,t+1} &= f_{i,E}^\alpha(\mathbf{X}_{1:i,t}, \mathbf{S}_t, u_{i,t}, \mathbf{W}_{i,t+1}) \\ &= \mathbf{1}_{\mathbb{R}_+}^\alpha(\mathbf{S}_t - \sum_{j=1}^i \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{j,t})) \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{i,t}) + \left(\mathbf{1}_{\mathbb{R}_+}^\alpha(u_{i,t} - \nu) \right. \\ &\quad \left. + \mathbf{1}_{\mathbb{R}_+}^\alpha(\mathbf{W}_{i,t+1} - p_i(\mathbf{A}_{i,t}))(1 - \mathbf{1}_{\mathbb{R}_+}^\alpha(u_{i,t} - \nu)) \right) \left(1 - \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{i,t}) \right). \end{aligned} \quad (\text{B.3})$$

We use the relaxed version of the indicator for $\mathbf{1}_{\{0\}}(\mathbf{E}_{i,t})$ and $\mathbf{1}_{\mathbb{R}_+}(u_{i,t} - \nu)$. We relax $\mathbf{1}_{\{1\}}(\mathbf{E}_{i,t})$ and $\mathbf{1}_{\mathbb{R}_+^*}(\nu - u_{i,t})$ as $1 - \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{i,t})$ and $1 - \mathbf{1}_{\mathbb{R}_+}^\alpha(u_{i,t} - \nu)$ respectively.

B.2 Relaxed dynamics of the age $A_{i,t}$

The relaxed dynamics of the age of parameter $\alpha > 0$ is given by:

$$\begin{aligned} A_{i,t+1} &= f_{i,A}^\alpha(\mathbf{X}_{1:i,t}, \mathbf{S}_t, u_{i,t}, \mathbf{W}_{i,t+1}) \\ &= (\mathbf{A}_{i,t} + 1) \left[\mathbf{1}_{\mathbb{R}_+^*}^\alpha \left(\sum_{j=1}^i \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{j,t}) - \mathbf{S}_t \right) \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{i,t}) \right. \\ &\quad \left. + \mathbf{1}_{\mathbb{R}_+}^\alpha(\mathbf{W}_{i,t+1} - p_i(\mathbf{A}_{i,t}))(1 - \mathbf{1}_{\mathbb{R}_+}^\alpha(u_{i,t} - \nu))(1 - \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{i,t})) \right] \\ &\quad + (1 - \mathbf{1}_{\mathbb{R}_+^*}^\alpha \left(\sum_{j=1}^i \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{j,t}) - \mathbf{S}_t \right)) \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{i,t}) \\ &\quad + \left((1 - u_{i,t}) \mathbf{A}_{i,t} + 1 \right) \mathbf{1}_{\mathbb{R}_+}^\alpha(u_{i,t} - \nu) (1 - \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{i,t})). \end{aligned} \quad (\text{B.4})$$

B.3 Relaxed dynamics of the vector of last failures $P_{i,t}$

The relaxed dynamics of parameter $\alpha > 0$ of the d -th element of the vector of last failures is given by:

$$\begin{aligned} P_{i,t+1}^d &= f_{i,P}^{d,\alpha}(\mathbf{X}_{1:i,t}, \mathbf{S}_t, u_{i,t}, \mathbf{W}_{i,t+1}) \\ &= \left((P_{i,t}^d + 1)(1 - \mathbf{1}_{\{\delta\}}^\alpha(P_{i,t}^d)) + \delta \mathbf{1}_{\{\delta\}}^\alpha(P_{i,t}^d) \right) \left(1 - \mathbf{1}_{\{1\}}^\alpha(\mathbf{E}_{i,t}) \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{i,t+1}) \right) \\ &\quad + \left((P_{i,t}^d + 1)(1 - \mathbf{1}_{\{\delta\}}^\alpha(P_{i,t}^d)) \mathbf{1}_{\{\delta\}}^\alpha(P_{i,t}^D) + \delta \mathbf{1}_{\{\delta\}}^\alpha(P_{i,t}^{d-1}) \mathbf{1}_{[2,D]}(d) \right) \\ &\quad + (P_{i,t}^{d+1} + 1)(1 - \mathbf{1}_{\{\delta\}}^\alpha(P_{i,t}^D)) \mathbf{1}_{[1,D-1]}(d) \mathbf{1}_{\{1\}}^\alpha(\mathbf{E}_{i,t}) \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{i,t+1}). \end{aligned} \quad (\text{B.5})$$

We do not relax $\mathbf{1}_{[2,D]}(d)$ and $\mathbf{1}_{[1,D-1]}(d)$. The reason is that these indicator functions do not arise from a discontinuity in the original dynamics. They are just used to take into account in the same equation the cases of $P_{i,t}^1$ and $P_{i,t}^D$ that have a slightly different expression than $P_{i,t}^d$ for $1 < d < D$.

B.4 Relaxed dynamics of the stock \mathbf{S}_t

The relaxed dynamics of the stock of parameter $\alpha > 0$ is given by:

$$\mathbf{S}_{t+1} = \mathbf{S}_t + \sum_{i=1}^n \sum_{d=1}^D \mathbf{1}_{\{D-1\}}^\alpha(\mathbf{P}_{i,t}^d) - \min \left\{ \mathbf{S}_t, \sum_{i=1}^n \mathbf{1}_{\{0\}}^\alpha(\mathbf{E}_{i,t}) \right\}. \quad (\text{B.6})$$

C Computation of optimal multipliers

At iteration k of the APP fixed-point algorithm, the subproblem on component $i \in \mathbb{I}$ is solved with the blackbox algorithm MADS [7]. MADS directly solves the constrained problem and outputs a primal solution $(\mathbf{X}_i^{k+1}, u_i^{k+1})$. Finding the primal solution \mathbf{S}^{k+1} of the subproblem on the stock just requires a simulation of the dynamics. For each subproblem we also have to compute optimal multipliers $\Lambda_1^{k+1}, \dots, \Lambda_n^{k+1}, \Lambda_{\mathbf{S}}^{k+1}$ to update the coordination term at the end of each iteration.

Suppose that the optimal solution and optimal multiplier of the auxiliary problem (4.8) are uniquely defined. As we know the primal solution, we can compute the optimal multiplier using the stationarity of the Lagrangian. In the following computation, we use the relaxed cost and dynamics to be able to compute the different gradients that appear, however for the sake of readability we drop the superscript α .

The Lagrangian L of the auxiliary problem (4.8) is given by:

$$\begin{aligned} L(\mathbf{X}, \mathbf{S}, u, \Lambda) = & \mathbb{E} \left(\sum_{i=1}^n \left(j_i(\mathbf{X}_i, u_i) + j^F(\bar{\mathbf{X}}_{1:i-1}, \mathbf{X}_i, \bar{\mathbf{X}}_{i+1:n}) \right) \right) \\ & + \frac{\gamma_x}{2} \|\mathbf{X} - \bar{\mathbf{X}}\|^2 + \frac{\gamma_s}{2} \|\mathbf{S} - \bar{\mathbf{S}}\|^2 + \frac{\gamma_u}{2} \|u - \bar{u}\|^2 \\ & + \langle \bar{\Lambda}, (\Theta'(\bar{\mathbf{X}}, \bar{\mathbf{S}}, \bar{u}, \mathbf{W}) - \Phi'(\bar{\mathbf{X}}, \bar{\mathbf{S}}, \bar{u}, \mathbf{W})) \cdot (\mathbf{X}, \mathbf{S}, u) \rangle \\ & + \langle \Lambda, \Phi(\mathbf{X}, \mathbf{S}, u, \mathbf{W}) \rangle. \end{aligned} \quad (\text{C.1})$$

At the saddle point $(\mathbf{X}^\sharp, \mathbf{S}^\sharp, u^\sharp, \Lambda^\sharp)$ of L we have:

$$\nabla L(\mathbf{X}^\sharp, \mathbf{S}^\sharp, u^\sharp, \Lambda^\sharp) = 0. \quad (\text{C.2})$$

Recall that for $i \in \{1, \dots, n, \mathbf{S}\}$, we have:

$$\Lambda_i^{k+1} = (\Lambda_{i,0}^{k+1}, \dots, \Lambda_{i,T}^{k+1}). \quad (\text{C.3})$$

Using (C.2) and knowing the solution $(\mathbf{X}^\sharp, \mathbf{S}^\sharp, u^\sharp)$ of the auxiliary problem, we can update the multiplier Λ^\sharp with a backward recursion.

Proposition C.1. *Let $i \in \mathbb{I}$. For the dynamics of component i in the auxiliary problem (4.8), the optimal multiplier $\Lambda_i^\sharp = (\Lambda_{i,0}^\sharp, \dots, \Lambda_{i,T}^\sharp)$ can be computed with the following backward recursion for $t \in \mathbb{T}$:*

$$\begin{aligned} \Lambda_{i,T}^\sharp = & -\nabla_{\mathbf{X}_{i,T}} j_{i,T}(\mathbf{X}_{i,T}^\sharp, u_{i,T}^\sharp) - \nabla_{\mathbf{X}_{i,T}} j_T^F(\bar{\mathbf{X}}_{1:i-1,T}, \mathbf{X}_{i,T}^\sharp, \bar{\mathbf{X}}_{i+1:n,T}) \\ & - \gamma_x(\mathbf{X}_{i,T}^\sharp - \bar{\mathbf{X}}_{i,T}), \\ \Lambda_{i,t}^\sharp = & -\nabla_{\mathbf{X}_{i,t}} j_{i,t}(\mathbf{X}_{i,t}^\sharp, u_{i,t}^\sharp) - \nabla_{\mathbf{X}_{i,t}} j_t^F(\bar{\mathbf{X}}_{1:i-1,t}, \mathbf{X}_{i,t}^\sharp, \bar{\mathbf{X}}_{i+1:n,t}) \\ & - \gamma_x(\mathbf{X}_{i,t}^\sharp - \bar{\mathbf{X}}_{i,t}) - \sum_{j=i+1}^n \partial_{\mathbf{X}_{i,t}} \Theta_{j,t+1}(\bar{\mathbf{X}}_{1:j}, \bar{\mathbf{S}}, \bar{u}_j, \mathbf{W}_j)^\top \cdot \bar{\Lambda}_{j,t+1} \\ & - \partial_{\mathbf{X}_{i,t}} \Theta_{\mathbf{S},t+1}(\bar{\mathbf{X}}_{1:n}, \bar{\mathbf{S}})^\top \cdot \bar{\Lambda}_{\mathbf{S},t+1} - \partial_{\mathbf{X}_{i,t}} \Phi_{i,t+1}(\mathbf{X}_i^\sharp, u_i^\sharp, \mathbf{W}_i)^\top \cdot \Lambda_{i,t+1}^\sharp. \end{aligned} \quad (\text{C.4})$$

Proof. The gradient of the Lagrangian L with respect to $\mathbf{X}_{i,t}$, $t \in \mathbb{T}$ is given by:

$$\begin{aligned}
\nabla_{\mathbf{X}_{i,T}} L(\mathbf{X}^\sharp, \mathbf{S}^\sharp, u^\sharp, \Lambda^\sharp) &= \nabla_{\mathbf{X}_{i,T}} j_{i,T}(\mathbf{X}_{i,T}^\sharp, u_{i,T}^\sharp) \\
&\quad + \nabla_{\mathbf{X}_{i,T}} j_T^F(\bar{\mathbf{X}}_{1:i-1,T}, \mathbf{X}_{i,T}^\sharp, \bar{\mathbf{X}}_{i+1:n,T}) \\
&\quad + \gamma_x(\mathbf{X}_{i,T}^\sharp - \bar{\mathbf{X}}_{i,T}) + \partial_{\mathbf{X}_{i,T}} \Phi_{i,T}(\mathbf{X}_i^\sharp, u_i^\sharp, \mathbf{W}_i)^\top \cdot \Lambda_{i,T}^\sharp, \\
\nabla_{\mathbf{X}_{i,t}} L(\mathbf{X}^\sharp, \mathbf{S}^\sharp, u^\sharp, \Lambda^\sharp) &= \nabla_{\mathbf{X}_{i,t}} j_{i,t}(\mathbf{X}_{i,t}^\sharp, u_{i,t}^\sharp) + \nabla_{\mathbf{X}_{i,t}} j_t^F(\bar{\mathbf{X}}_{1:i-1,t}, \mathbf{X}_{i,t}^\sharp, \bar{\mathbf{X}}_{i+1:n,t}) \\
&\quad + \gamma_x(\mathbf{X}_{i,t}^\sharp - \bar{\mathbf{X}}_{i,t}) + \partial_{\mathbf{X}_{i,t}} \Theta_{\mathbf{S},t+1}(\bar{\mathbf{X}}_{1:n}, \bar{\mathbf{S}})^\top \cdot \bar{\Lambda}_{\mathbf{S},t+1} \\
&\quad + \sum_{j=i+1}^n \partial_{\mathbf{X}_{i,t}} \Theta_{j,t+1}(\bar{\mathbf{X}}_{1:j}, \bar{\mathbf{S}}, \bar{u}_j, \mathbf{W}_j)^\top \cdot \bar{\Lambda}_{j,t+1} \\
&\quad + \partial_{\mathbf{X}_{i,t}} \Phi_{i,t}(\mathbf{X}_i^\sharp, u_i^\sharp, \mathbf{W}_i)^\top \cdot \Lambda_{i,t}^\sharp \\
&\quad + \partial_{\mathbf{X}_{i,t}} \Phi_{i,t+1}(\mathbf{X}_i^\sharp, u_i^\sharp, \mathbf{W}_i)^\top \cdot \Lambda_{i,t+1}^\sharp.
\end{aligned} \tag{C.5}$$

Using

$$\nabla_{\mathbf{X}_{i,t}} L(\mathbf{X}^\sharp, \mathbf{S}^\sharp, u^\sharp, \Lambda^\sharp) = 0, \quad t \in \mathbb{T}, \tag{C.6}$$

$$\partial_{\mathbf{X}_{i,t}} \Phi_{i,t}(\mathbf{X}_i^\sharp, u_i^\sharp, \mathbf{W}_i) = \mathbf{I}, \quad t \in \mathbb{T}, \tag{C.7}$$

where \mathbf{I} is the identity matrix of appropriate size, we get the formula (C.4).

Proposition C.2. *The optimal multiplier $\Lambda_{\mathbf{S}}^\sharp = (\Lambda_{\mathbf{S},0}^\sharp, \dots, \Lambda_{\mathbf{S},T}^\sharp)$ associated to the dynamics of the stock in the auxiliary problem (4.8) can be computed with the following backward recursion for $t \in \mathbb{T}$:*

$$\begin{aligned}
\Lambda_{\mathbf{S},T}^\sharp &= -\gamma_s(\mathbf{S}_T^\sharp - \bar{\mathbf{S}}_T), \\
\Lambda_{\mathbf{S},t}^\sharp &= -\gamma_s(\mathbf{S}_t^\sharp - \bar{\mathbf{S}}_t) - \sum_{i=1}^n \partial_{\mathbf{S}_t} \Theta_{i,t+1}(\bar{\mathbf{X}}_{1:i}, \bar{\mathbf{S}}, \bar{u}_i, \mathbf{W}_i)^\top \cdot \bar{\Lambda}_{i,t+1} \\
&\quad - \partial_{\mathbf{S}_t} \Phi_{\mathbf{S},t+1}(\mathbf{S}^\sharp)^\top \cdot \Lambda_{\mathbf{S},t+1}^\sharp.
\end{aligned} \tag{C.8}$$

Proof. The gradient of the Lagrangian L with respect to \mathbf{S}_t , $t \in \mathbb{T}$ is given by:

$$\begin{aligned}
\nabla_{\mathbf{S}_T} L(\mathbf{X}^\sharp, \mathbf{S}^\sharp, u^\sharp, \Lambda^\sharp) &= \gamma(\mathbf{S}_T^\sharp - \bar{\mathbf{S}}_T) + \partial_{\mathbf{S}_T} \Phi_{\mathbf{S},T}(\mathbf{S}^\sharp)^\top \cdot \Lambda_{\mathbf{S},T}^\sharp, \\
\nabla_{\mathbf{S}_t} L(\mathbf{X}^\sharp, \mathbf{S}^\sharp, u^\sharp, \Lambda^\sharp) &= \gamma(\mathbf{S}_t^\sharp - \bar{\mathbf{S}}_t) \\
&\quad + \sum_{i=1}^n \partial_{\mathbf{S}_t} \Theta_{i,t+1}(\bar{\mathbf{X}}_{1:i}, \bar{\mathbf{S}}, \bar{u}_i, \mathbf{W}_i)^\top \cdot \bar{\Lambda}_{i,t+1} \\
&\quad + \partial_{\mathbf{S}_t} \Phi_{\mathbf{S},t+1}(\mathbf{S}^\sharp)^\top \cdot \Lambda_{\mathbf{S},t+1}^\sharp + \partial_{\mathbf{S}_t} \Phi_{\mathbf{S},t}(\mathbf{S}^\sharp)^\top \cdot \Lambda_{\mathbf{S},t}^\sharp.
\end{aligned} \tag{C.9}$$

Using

$$\nabla_{\mathbf{S}_t} L(\mathbf{X}^\sharp, \mathbf{S}^\sharp, u^\sharp, \Lambda^\sharp) = 0, \quad t \in \mathbb{T}, \tag{C.10}$$

$$\partial_{\mathbf{S}_t} \Phi_{\mathbf{S},t}(\mathbf{S}^\sharp) = \mathbf{1}, \quad t \in \mathbb{T}, \tag{C.11}$$

we get the backward recursion (C.8).

D Derivative of the relaxed indicator function

We give some details about the derivative of the relaxed indicator function. The relaxed indicator function $\mathbf{1}_{\mathcal{A}}^\alpha$, where $\alpha > 0$, appears in the dynamics and cost with three main cases for the set $\mathcal{A} \subset \mathbb{R}$. Note that $\mathbf{1}_{\mathcal{A}}^\alpha$ is not differentiable at $x \in \mathbb{R}$ if $d(\mathcal{A}, x) = \frac{1}{2\alpha}$. At such point, the derivative is taken to be 0. The following situations occur:

1. \mathcal{A} is a singleton $\{a\}$, then for $x \in \mathbb{R}$:

$$\mathbf{1}_{\{a\}}^\alpha(x) = \begin{cases} 1 - 2\alpha|x - a| & \text{if } |x - a| \leq \frac{1}{2\alpha}, \\ 0 & \text{if } |x - a| > \frac{1}{2\alpha}. \end{cases} \quad (\text{D.1})$$

Hence the derivative $\mathbf{1}'_{\{a\}}^\alpha$ is given by:

$$\mathbf{1}'_{\{a\}}^\alpha(x) = \begin{cases} 2\alpha & \text{if } a - \frac{1}{2\alpha} < x < a, \\ -2\alpha & \text{if } a < x < a + \frac{1}{2\alpha}, \\ 0 & \text{otherwise.} \end{cases} \quad (\text{D.2})$$

2. $\mathcal{A} = \mathbb{R}_+$ then for $x \in \mathbb{R}$ we have:

$$\mathbf{1}_{\mathbb{R}_+}^\alpha(x) = \begin{cases} 2\alpha x + 1 & \text{if } -\frac{1}{2\alpha} < x < 0, \\ 1 & \text{if } x \geq 0, \\ 0 & \text{if } x \leq -\frac{1}{2\alpha}. \end{cases} \quad \mathbf{1}'_{\mathbb{R}_+}^\alpha(x) = \begin{cases} 2\alpha & \text{if } -\frac{1}{2\alpha} < x < 0, \\ 0 & \text{otherwise.} \end{cases} \quad (\text{D.3})$$

3. $\mathcal{A} = \mathbb{R}_+^*$: if we strictly apply Definition 4.2, we would have $\mathbf{1}_{\mathbb{R}_+^*}^\alpha = \mathbf{1}_{\mathbb{R}_+}^\alpha$. However with this definition we would not have pointwise convergence of $\mathbf{1}_{\mathbb{R}_+^*}^\alpha$ towards $\mathbf{1}_{\mathbb{R}_+^*}$ as α goes to 0. Indeed, for all $\alpha > 0$ we would have $\mathbf{1}_{\mathbb{R}_+^*}^\alpha(0) = 1$ but $\mathbf{1}_{\mathbb{R}_+^*}(0) = 0$. To overcome this issue we define $\mathbf{1}_{\mathbb{R}_+^*}^\alpha$ as follows:

$$\mathbf{1}_{\mathbb{R}_+^*}^\alpha(x) = \begin{cases} 2\alpha x & \text{if } 0 < x < \frac{1}{2\alpha}, \\ 1 & \text{if } x \geq \frac{1}{2\alpha}, \\ 0 & \text{if } x \leq 0. \end{cases} \quad \mathbf{1}'_{\mathbb{R}_+^*}^\alpha(x) = \begin{cases} 2\alpha & \text{if } 0 < x < \frac{1}{2\alpha}, \\ 0 & \text{otherwise.} \end{cases} \quad (\text{D.4})$$

Using these formulas for the derivative of the relaxed indicator function and the explicit expressions of the relaxed cost function and relaxed dynamics given in §4.3.3 and Appendix B respectively, all the gradients that appear either in the objective function of the subproblems or in the backward recursion for the multiplier update can be computed.

References

- [1] S. ALARIE, C. AUDET, P.-Y. BOUCHET, AND S. L. DIGABEL, *Optimization of noisy blackboxes with adaptive precision*, arXiv:1911.05846 [math], (2019). arXiv: 1911.05846.

- [2] A. ALMAKHLAFI AND J. KNOWLES, *Benchmarks for maintenance scheduling problems in power generation*, in 2012 IEEE Congress on Evolutionary Computation, Brisbane, Australia, June 2012, pp. 1–8.
- [3] A. ALRABGHI AND A. TIWARI, *State of the art in simulation-based optimisation for maintenance systems*, *Computers & Industrial Engineering*, 82 (2015), pp. 167–182.
- [4] H. AOUDJIT, *Planification de la maintenance d'un parc de turbines-alternateurs par programmation mathématique*, PhD thesis, Université de Montréal, 2010.
- [5] K. J. ARROW AND L. HURWICZ, *Decentralization and Computation in Resource Allocation*, Stanford University, Department of Economics, 1960.
- [6] C. AUDET, J. E. DENNIS, AND S. LE DIGABEL, *Parallel Space Decomposition of the Mesh Adaptive Direct Search Algorithm*, *SIAM Journal on Optimization*, 19 (2008), pp. 1150–1170.
- [7] C. AUDET AND J. E. DENNIS, JR, *Mesh Adaptive Direct Search Algorithms for Constrained Optimization*, *SIAM Journal on Optimization*, 17 (2006), pp. 188–217.
- [8] T. BITTAR, *Stochastic Optimization of Maintenance Scheduling: Blackbox Methods, Decomposition Approaches - Theoretical and Numerical Aspects*, PhD thesis, Ecole des Ponts ParisTech, 2021.
- [9] F. CAMPOLONGO, J. CARIBONI, AND A. SALTELLI, *An effective screening design for sensitivity analysis of large models*, *Environmental Modelling & Software*, 22 (2007), pp. 1509–1518.
- [10] P. CARPENTIER AND G. COHEN, *Décomposition-coordination en optimisation déterministe et stochastique*, vol. 81 of *Mathématiques et Applications*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2017.
- [11] M.-C. CHEN, C.-M. HSU, AND S.-W. CHEN, *Optimizing joint maintenance and stock provisioning policy for a multi-echelon spare parts logistics network*, *Journal of the Chinese Institute of Industrial Engineers*, 23 (2006), pp. 289–302.
- [12] D. I. CHO AND M. PARLAR, *A survey of maintenance models for multi-unit systems*, *European Journal of Operational Research*, 51 (1991), pp. 1–23.
- [13] G. COHEN, *Optimization by decomposition and coordination: A unified approach*, *IEEE Transactions on Automatic Control*, 23 (1978), pp. 222–232.
- [14] G. COHEN, *Auxiliary problem principle and decomposition of optimization problems*, *Journal of Optimization Theory and Applications*, 32 (1980), pp. 277–305.
- [15] G. DAMBLIN, M. COUPLET, AND B. IOOSS, *Numerical studies of space-filling designs: optimization of Latin Hypercube Samples and subprojection properties*, *Journal of Simulation*, 7 (2013), pp. 276–289.
- [16] A. DIABAT, J.-P. RICHARD, AND C. W. CODRINGTON, *A Lagrangian relaxation approach to simultaneous strategic and tactical planning in supply chain design*, *Annals of Operations Research*, 203 (2013), pp. 55–80.
- [17] S.-H. DING AND S. KAMARUDDIN, *Maintenance policy optimization—literature review and directions*, *The International Journal of Advanced Manufacturing Technology*, 76 (2015), pp. 1263–1283.

- [18] M. FATTAHI, M. MAHOOTCHI, H. MOSADEGH, AND F. FALLAHI, *A new approach for maintenance scheduling of generating units in electrical power systems based on their operational hours*, Computers & Operations Research, 50 (2014), pp. 61–79.
- [19] A. FROGER, M. GENDREAU, J. E. MENDOZA, E. PINSON, AND L.-M. ROUSSEAU, *Maintenance scheduling in the electricity industry: A literature review*, European Journal of Operational Research, 251 (2016), pp. 695–706.
- [20] A. GRIGORIEV, J. VAN DE KLUNDERT, AND F. C. SPIEKSMAN, *Modeling and solving the periodic maintenance problem*, European Journal of Operational Research, 172 (2006), pp. 783–797.
- [21] T. KAIHARA, N. FUJII, A. TSUJIBE, AND Y. NONAKA, *Proactive maintenance scheduling in a re-entrant flow shop using Lagrangian decomposition coordination method*, CIRP Annals, 59 (2010), pp. 453–456.
- [22] L. S. LASDON AND J. D. SCHOEFFLER, *A multi-level technique for optimization*, Joint Automatic Control Conference, 3 (1965), pp. 85–92.
- [23] R. LUSBY, L. F. MULLER, AND B. PETERSEN, *A solution approach based on Benders decomposition for the preventive maintenance scheduling problem of a stochastic large-scale energy system*, Journal of Scheduling, 16 (2013), pp. 605–628.
- [24] M. D. MESAROVIĆ, D. MACKO, AND Y. TAKAHARA, *Theory of Hierarchical, Multilevel Systems*, vol. 68 of Mathematics in Science and Engineering, Academic Press, 1970.
- [25] R. P. NICOLAI AND R. DEKKER, *Optimal Maintenance of Multi-component Systems: A Review*, in Complex System Maintenance Handbook, Springer London, 2008, pp. 263–286.
- [26] F. J. NOGALES, F. J. PRIETO, AND A. J. CONEJO, *A Decomposition Methodology Applied to the Multi-Area Optimal Power Flow Problem*, Annals of Operations Research, 120 (2003), pp. 99–116.
- [27] M. P. NOWAK AND W. RÖMISCH, *Stochastic Lagrangian Relaxation Applied to Power Scheduling in a Hydro-Thermal System under Uncertainty*, Annals of Operations Research, 100 (2000), pp. 251–272.
- [28] M. C. OLDE KEIZER, S. D. P. FLAPPER, AND R. H. TEUNTER, *Condition-based maintenance policies for systems with multiple dependent components: A review*, European Journal of Operational Research, 261 (2017), pp. 405–420.
- [29] M. PATRIKSSON, A.-B. STRÖMBERG, AND A. WOJCIECHOWSKI, *The stochastic opportunistic replacement problem, part II: a two-stage solution approach*, Annals of Operations Research, 224 (2015), pp. 51–75.
- [30] S. H. A. RAHMATI, A. AHMADI, AND K. GOVINDAN, *A novel integrated condition-based maintenance and stochastic flexible job shop scheduling problem: simulation-based optimization approach*, Annals of Operations Research, 269 (2018), pp. 583–621.
- [31] N. REZG, X. XIE, AND Y. MATI, *Joint optimization of preventive maintenance and inventory control in a production line using simulation*, International Journal of Production Research, 42 (2004), pp. 2029–2046.
- [32] O. ROUX, M. A. JAMALI, D. A. KADI, AND E. CHÂTELET, *Development of simulation and optimization platform to analyse maintenance policies performances for manufacturing systems*, International Journal of Computer Integrated Manufacturing, 21 (2008), pp. 407–414.

- [33] N. SADATI AND A. BABAZADEH, *Optimal control of robot manipulators with a new two-level gradient-based approach*, *Electrical Engineering*, 88 (2006), pp. 383–393.
- [34] R. SARKER AND A. HAQUE, *Optimization of maintenance and spare provisioning policy using simulation*, *Applied Mathematical Modelling*, 24 (2000), pp. 751–760.
- [35] K. SURESH AND N. KUMARAPPAN, *Hybrid improved binary particle swarm optimization approach for generation maintenance scheduling problem*, *Swarm and Evolutionary Computation*, 9 (2013), pp. 69–89.
- [36] M. SÁNCHEZ-SILVA, D. M. FRANGOPOL, J. PADGETT, AND M. SOLIMAN, *Maintenance and Operation of Infrastructure Systems: Review*, *Journal of Structural Engineering*, 142 (2016), p. F4016004.
- [37] Y. TAKAHARA, *Multilevel Approach to Dynamic Optimization*, Tech. Report SRC-50- C-64-18, Case Western Reserve University, Systems Research Center, 1964.