Decomposition Methods for Monotone Two-Time-Scale Stochastic Optimization Problems

Tristan Rigaut, Pierre Carpentier, Jean-Philippe Chancelier, Michel de Lara

Decomposition Methods for Monotone Two-Time-Scale Stochastic Optimization Problems

Tristan Rigaut[*], Pierre Carpentier[†], Jean-Philippe Chancelier[‡], Michel De Lara[‡]

August 29, 2023

Abstract

It is common that strategic investment decisions are made at a slow time-scale, whereas operational decisions are made at a fast time-scale. Hence, the total number of decision stages may be huge. In this paper, we consider multistage stochastic optimization problems with two time-scales, and we propose a time block decomposition scheme to address them numerically. More precisely, i) we write recursive Bellman-like equations at the slow time-scale and ii), under a suitable monotonicity assumption, we propose computable upper and lower bounds — relying respectively on primal and dual decomposition — for the corresponding slow time-scale Bellman functions. With these functions, we are able to design policies. We assess the methods tractability and validate their efficiency by solving a battery management problem where the fast time-scale operational decisions have an impact on the storage current capacity, hence on the strategic decisions to renew the battery at the slow time-scale.

1 Introduction, motivation and context

In energy management, it is common that strategic investment decisions (storage capacities, production units) are made at a slow time-scale, whereas operational decisions (storage management, production) are made at a fast time-scale. The total number of decision stages may be huge, which leads to numerically untractable optimization problems — for instance, a two-time-scale stochastic optimal problem where fast controlled stochastic dynamics (e.g. a change every fifteen or thirty minutes) affect a controlled long term stochastic behavior (e.g. a change every day or every week) over several years. How can we nevertheless provide numerical solutions (policies) to such problems?

1.1 Literature review

Stochastic Dynamic Programming (SDP) based on the Bellman equation [4] is a standard method to solve a multistage stochastic optimization problem by time decomposition. This method suffers the so called *curses of dimensionality* as introduced in [4, 7, 20]. In particular the complexity of the most classical implementation of SDP (that discretizes the state space) is exponential in the number of state variables.

[*]Schneider Electric, Grenoble, France
[†]UMA, ENSTA Paris, IP Paris, Palaiseau, France
[‡]CERMICS, École des Ponts, Marne-la-Vallée, France

A major contribution to handle a large number of state variables is the well-known Stochastic Dual Dynamic Programming (SDDP) algorithm [18]. This method is adapted to problems with linear dynamics and convex costs. Other similar methods have been developed such as Mixed Integer Dynamic Approximation Scheme (MIDAS) [19] or Stochastic Dual Dynamic Integer Programming (SDDiP) [26] for nonconvex problems, in particular those displaying binary variables. The performance of these algorithms is sensitive to the number of time steps [15, 19].

Other classical stochastic optimization methods are even more sensitive to the number of time stages. It is well-known that solving a multistage stochastic optimization problem on a scenario tree displays a complexity exponential in the number of time steps.

Problems displaying a large number of time stages, in particular problems with multiple time-scales, require to design specific methods. A class of stochastic optimization problems to deal with two time-scales has been introduced in [14] and further formalized in [16]. It is called Multi-Horizon Stochastic Optimization Problems and it frames problems where uncertainty is modeled using multi-horizon scenario trees as rigorously studied in [25]. Several authors have studied stochastic optimization problems with interdependent strategic/operational decisions or intrastage/interstage problems [3, 1, 2, 24, 21], but most of the time the developed methods to tackle the difficulties are problem-dependent. In [14] the authors present different particular cases where the two time-scales (called operational and strategic decision problems) can be easily decomposed. In [16] a formal definition of a Multi Horizon Stochastic Program is given and methods to compute bounds are developed: formal Multi Horizon Stochastic Program is a stochastic optimization problem with linear cost and dynamics where uncertainties are modeled as multi time-scale scenario trees.

1.2 Paper contributions and organization

In this paper, we propose a framework to formally define stochastic optimization problems naturally displaying two time-scales, that is, a slow time-scale (like days) and a fast time-scale (like half hours). The ultimate goal is to design tractable algorithms for such problems with hundreds of thousands of time steps, without requiring a stationary/infinite horizon assumption (contrarily to [12]) and in a stochastic setting (which extends [13]).

The paper is organized as follows. In Sect. 2, we outline the setting of a generic two-time-scale multistage stochastic optimization problem. In Sect. 3, we show how to write Bellman equations at the slow time-scale (the resulting Dynamic Programming equation is referred to as the Bellman equation by time blocks, and is detailed in [9, Sect. 5]). If we suppose slow time-scale stagewise independence of the noise process, the corresponding Bellman functions provide both the optimal cost and optimal policies. If not, we nevertheless are able to derive feasible policies from the Bellman functions, which is our main objective. Then, under a monotonicity-inducing assumption, we obtain a more tractable version of the Bellman equation, by relaxing the problem dynamics without changing the slow time-scale Bellman functions. In Sect. 4, we devise two algorithms. The first algorithm, akin to the so-called price decomposition, gives a lower bound of the slow time-scale Bellman functions, whereas the second algorithm, based on resource decomposition, gives an upper bound. This upper bound is relevant, that is, not almost surely equal to $+\infty$, for monotone multistage stochastic optimization problems. In Sect. 5, we indicate how to obtain policies, and we discuss optimality. In Sect. 6, we present an application of the above method to a battery management problem incorporating a very large number of time steps. We finally discuss how to take advantage of periodicity properties at the slow time-scale in Appendix A, we give some insights on the numerical complexity of the decomposition methods in Appendix B, and we prove the desired monotonicity-inducing assumptions for the battery problem in Appendix C.

2 Two-time-scale stochastic optimization problems

We present a formal definition of a two-time-scale stochastic optimization problem, that is, with a slow time-scale and a fast time-scale.

2.1 Notations for two time-scales

Given two natural numbers $r \leq s$, we use either the notation $[\![r, s]\!]$ or the notation $r{:}s$ for the set $\{r, r+1, \dots, s-1, s\}$.

To properly handle two time-scales, we adopt the following notations. For a given constant time interval $\Delta t > 0$, let $M \in \mathbb{N}^*$ be such that $(M+1)$ is the number of time steps in the slow time step, e.g. for $\Delta t = 30$ minutes, $M + 1 = 48$, when the slow time step correspond to a day. A decision-maker has to make decisions on two time-scales over a given number of slow time steps $(D+1) \in \mathbb{N}^*$:

1. one type of (say, operational) decision every fast time step $m \in [\![0, M]\!]$ of every slow time step $d \in [\![0, D]\!]$,
2. another type of (say, strategic) decision every slow time step $d \in [\![0, D]\!] \cup \{D{+}1\}$.

In our model the time flows between two slow time steps d and $d+1$ as follows:

$$d, 0 \xrightarrow[\Delta t]{} d, 1 \xrightarrow[\Delta t]{} \dots \xrightarrow[\Delta t]{} d, M \xrightarrow[\Delta t]{} d+1, 0$$

A variable z will have two time indexes $z_{d,m}$ if it changes every fast time step m of every slow time step d. An index (d, m) belongs to the set

$$\mathbb{T} = [\![0, D]\!] \times [\![0, M]\!] \cup \{(D{+}1, 0)\} , \tag{1}$$

which is a totally ordered set when equipped with the following lexicographical order $\preceq$:

$$(d, m) \preceq (d', m') \iff (d \leq d') \text{ or } \big(d = d' \text{ and } m \leq m'\big) . \tag{2}$$

We also use the following notations for describing sequences of variables and sequences of spaces. For (d, m) and $(d, m') \in \mathbb{T}$, with $m \leq m'$:

- the notation $z_{d,m:m'}$ refers to the sequence of variables $\big(z_{d,m}, z_{d,m+1}, \dots, z_{d,m'-1}, z_{d,m'}\big)$,
- the notation $\mathbb{Z}_{d,m:m'}$ refers to the Cartesian product $\prod_{k=m}^{m'} \mathbb{Z}_{d,k}$ of spaces $\{\mathbb{Z}_{d,k}\}_{k \in [\![m,m']\!]}$.

2.2 Two-time-scale multistage stochastic optimization setting

We consider a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Random variables are denoted using bold letters, and we denote by $\sigma(\mathbf{Z})$ the σ-algebra generated by the random variable $\mathbf{Z}$.

We consider an exogenous noise process $\mathbf{W} = \{\mathbf{W}_d\}_{d \in [\![0,D]\!]}$ at the slow time-scale, as detailed below. For any $d \in [\![0,D]\!]$, the random variable $\mathbf{W}_d$ consists of a sequence of random variables $\{\mathbf{W}_{d,m}\}_{m \in [\![0,M]\!]}$ at the fast time-scale:

$$\mathbf{W}_d = (\mathbf{W}_{d,0}, \dots, \mathbf{W}_{d,m}, \dots, \mathbf{W}_{d,M}) \ . \tag{3}$$

Each random variable $\mathbf{W}_{d,m} : \Omega \to \mathbb{W}_{d,m}$ takes values in a *Borel space*[1] $\mathbb{W}_{d,m}$ ("uncertainty" space), so that $\mathbf{W}_d : \Omega \to \mathbb{W}_d$ takes values in the product space $\mathbb{W}_d = \mathbb{W}_{d,0:M}$. For any $(d,m) \in \mathbb{T}$, we denote by $\mathcal{F}_{d,m}$ the σ-field generated by all noises up to time (d,m), that is,

$$\mathcal{F}_{d,m} = \sigma\big(\mathbf{W}_0, \dots, \mathbf{W}_{d-1}, \mathbf{W}_{d,0}, \dots, \mathbf{W}_{d,m}\big) \ . \tag{4a}$$

We also introduce the filtration $\mathcal{F}_{[\![0,D]\!]}$ at the slow time-scale:

$$\mathcal{F}_{[\![0,D]\!]} = \{\mathcal{F}_{d,M}\}_{d \in [\![0,D]\!]} = \big(\mathcal{F}_{0,M}, \dots, \mathcal{F}_{D,M}\big) \ . \tag{4b}$$

In the same vein, we introduce a decision process $\mathbf{U} = \{\mathbf{U}_d\}_{d \in [\![0,D]\!]}$ at the slow time-scale, where each $\mathbf{U}_d$ consists of a sequence $\{\mathbf{U}_{d,m}\}_{m \in [\![0,M]\!]}$ of decision variables at the fast time-scale:

$$\mathbf{U}_d = (\mathbf{U}_{d,0}, \dots, \mathbf{U}_{d,m}, \dots, \mathbf{U}_{d,M}) \ . \tag{5}$$

Each random variable $\mathbf{U}_{d,m} : \Omega \to \mathbb{U}_{d,m}$ takes values in a Borel space $\mathbb{U}_{d,m}$ ("control" space), and we denote by $\mathbb{U}_d$ the Cartesian product $\mathbb{U}_{d,0:M}$. We finally introduce a state process $\mathbf{X} = \{\mathbf{X}_d\}_{d \in [\![0,D+1]\!]}$ at the slow time-scale, where each random variable $\mathbf{X}_d : \Omega \to \mathbb{X}_d$ takes values in a Borel space $\mathbb{X}_d$ ("state" space). Note that, unlike processes $\mathbf{W}$ and $\mathbf{U}$, *the state process* $\mathbf{X}$ *is defined only at the slow time-scale*. Thus, for any $d \in [\![0,D{+}1]\!]$, the random variable $\mathbf{X}_d$ represents the system state at time $(d,0)$.

We also consider Borel spaces $\mathbb{Y}_d$ such that, for each $d \in [\![0,D{+}1]\!]$, $\mathbb{X}_d$ and $\mathbb{Y}_d$ are paired spaces when equipped with a bilinear form $\langle \cdot \, , \cdot \rangle$. In this paper, we assume that each state space $\mathbb{X}_d$ is the vector space $\mathbb{R}^{n_d}$, so that $\mathbb{Y}_d = \mathbb{R}^{n_d}$, the bilinear form $\langle \cdot \, , \cdot \rangle$ being the standard scalar product.

For each $d \in [\![0,D]\!]$, we introduce a nonnegative Borel-measurable instantaneous cost function $L_d : \mathbb{X}_d \times \mathbb{U}_d \times \mathbb{W}_d \to [0,+\infty]$ and a Borel-measurable dynamics $f_d : \mathbb{X}_d \times \mathbb{U}_d \times \mathbb{W}_d \to \mathbb{X}_{d+1}$. Note that both the instantaneous cost L_d and the dynamics f_d depend on all the fast time-scale decision and noise variables constituting the slow time step d. We also introduce a nonnegative Borel-measurable final cost function $K : \mathbb{X}_{D+1} \to [0,+\infty]$.[2]

With all these ingredients, we write a two-time-scale stochastic optimization problem

$$V^{\mathrm{e}}(x) = \inf_{\mathbf{X},\mathbf{U}} \mathbb{E}\bigg[\sum_{d=0}^{D} L_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d) + K(\mathbf{X}_{D+1})\bigg] \ , \tag{6a}$$

$$\text{s.t} \quad \mathbf{X}_0 = x \ , \ \ \mathbf{X}_{d+1} = f_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d) \ , \ \ \forall d \in [\![0,D]\!] \ , \tag{6b}$$

$$\sigma(\mathbf{U}_{d,m}) \subset \mathcal{F}_{d,m} \ , \ \ \forall (d,m) \in [\![0,D]\!] \times [\![0,M]\!] \ . \tag{6c}$$

The expected cost value in (6) is well defined, as all functions are nonnegative measurable. Constraint (6c) — where $\sigma(\mathbf{U}_{d,m})$ is the σ-field generated by the random variable $\mathbf{U}_{d,m}$ —

[1]We call Borel space $(\mathbb{X}, \mathcal{B}_{\mathbb{X}})$ a Borel set $\mathbb{X}$ equipped with its Borel σ-field $\mathcal{B}_{\mathbb{X}}$ ([8, Definition 7.7, p. 118]).

[2]We could also consider either bounded function, or uniformly bounded below function. However, for the sake of simplicity, we deal in the sequel with nonnegative cost functions L_d and K.

expresses the fact that each decision $\mathbf{U}_{d,m}$ is $\mathcal{F}_{d,m}$-measurable, that is, is nonanticipative. The function V^{e} is called the *optimal value function* of Problem (6).

The notation $V^{\mathrm{e}}(x)$ for the optimal value of Problem (6) emphasizes the fact that the dynamics equations (6b) correspond to equality constraints (as is classical). We also introduce a relaxation of Problem (6). For this purpose, we consider the following multistage stochastic optimization problem:

$$V^{\mathrm{i}}(x) = \inf_{\mathbf{X},\mathbf{U}} \mathbb{E}\bigg[\sum_{d=0}^{D} L_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d) + K(\mathbf{X}_{D+1})\bigg] , \tag{7a}$$

$$\text{s.t} \quad \mathbf{X}_0 = x \ , \ \ f_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d) \geq \mathbf{X}_{d+1} \ , \ \ \forall d \in [\![0, D]\!] \ , \tag{7b}$$

$$\sigma(\mathbf{U}_{d,m}) \subset \mathcal{F}_{d,m} \ , \ \ \forall (d,m) \in [\![0, D]\!] \times [\![0, M]\!] \ , \tag{7c}$$

$$\sigma(\mathbf{X}_{d+1}) \subset \mathcal{F}_{d,M} \ , \ \ \forall d \in [\![0, D]\!] \ . \tag{7d}$$

We have relaxed the dynamic equality constraints (6b) into inequality constraints (7b). Thus, Problem (7) is less constrained than Problem (6), so that the optimal value function V^{i} of Problem (7) is less than the optimal value function V^{e} of Problem (6):

$$V^{\mathrm{i}}(x) \leq V^{\mathrm{e}}(x) \ , \ \ \forall x \in \mathbb{X}_0 \ . \tag{8}$$

Remark 1 *We just consider as explicit constraints the dynamic constraints* (6b) *and the nonanticipativity constraints* (6c)*, but other constraints involving the state and the control can be incorporated in the instantaneous cost L_d and in the final cost K by means of indicator functions*[3] *as L_d and K can take the value $+\infty$.*

Problem (6) seems very similar to a classical discrete time multistage stochastic optimization problem. But an important difference appears in the nonanticipativity constraints (6c) that express the fact that the decision vector $\mathbf{U}_d = (\mathbf{U}_{d,0}, \ldots, \mathbf{U}_{d,M})$ at every slow time step d does not display the same measurability for each component (information grows every fast time step). This point of view is not referred to in the literature and is one of the novelty of our approach.

3 Time block decomposition, Bellman functions and monotonicity assumptions

In §3.1, we introduce Bellman functions at the slow time scale, as a way to decompose a two-time-scale stochastic optimization problem in time blocks. In §3.2, we introduce assumptions on the data of Problem (6) which allow us to make the link between the sequence of Bellman functions associated with Problem (6) and the sequence of Bellman functions associated with Problem (7).

3.1 Time block decomposition and Bellman functions at the slow time-scale

Stochastic Dynamic Programming, based on Bellman optimality principle, is a classical way to decompose multistage stochastic optimization problems into multiple but smaller static

[3]The indicator function δ_A of a set A is defined as $\delta_A(a) = 0$ if $a \in A$ and $\delta_A(a) = +\infty$ if $a \notin A$.

optimization problems. In this paragraph, we apply the Bellman recursion by time blocks to decompose the multistage two-time-scale stochastic optimization Problem (7) into multiple smaller problems that are stochastic optimization problems over a single slow time step.

We first introduce a sequence $\{V_d^{\mathrm{e}}\}_{d\in[\![0,D+1]\!]}$ of *slow time-scale Bellman functions* associated with Problem (6). These functions are defined by backward induction as follows. At time $D{+}1$, we set $V_{D+1}^{\mathrm{e}} = K$, and then, for $d \in [\![0, D]\!]$ and for all $x \in \mathbb{X}_d$, we set

$$
\begin{aligned}
V_d^{\mathrm{e}}(x) = \inf_{\mathbf{X}_{d+1}, \mathbf{U}_d} \ & \mathbb{E}\Big[L_d(x, \mathbf{U}_d, \mathbf{W}_d) + V_{d+1}^{\mathrm{e}}(\mathbf{X}_{d+1})\Big] \ , && \text{(9a)} \\
\text{s.t} \quad & \mathbf{X}_{d+1} = f_d(x, \mathbf{U}_d, \mathbf{W}_d) \ , && \text{(9b)} \\
& \sigma(\mathbf{U}_{d,m}) \subset \sigma(\mathbf{W}_{d,0:m}) \ , \ \ \forall m \in [\![0, M]\!] \ , && \text{(9c)}
\end{aligned}
$$

the expectation in (9a) being taken with respect to the marginal probability of the random vector $\mathbf{W}_d$. We also introduce a sequence of *slow time-scale Bellman functions* $\{V_d^{\mathrm{i}}\}_{d\in[\![0,D+1]\!]}$ associated with Problem (7). At time $D{+}1$, we set $V_{D+1}^{\mathrm{i}} = K$, and then, for $d \in [\![0, D]\!]$ and for all $x \in \mathbb{X}_d$, we set

$$
\begin{aligned}
V_d^{\mathrm{i}}(x) = \inf_{\mathbf{X}_{d+1}, \mathbf{U}_d} \ & \mathbb{E}\Big[L_d(x, \mathbf{U}_d, \mathbf{W}_d) + V_{d+1}^{\mathrm{i}}(\mathbf{X}_{d+1})\Big] \ , && \text{(10a)} \\
\text{s.t} \quad & f_d(x, \mathbf{U}_d, \mathbf{W}_d) \geq \mathbf{X}_{d+1} \ , && \text{(10b)} \\
& \sigma(\mathbf{U}_{d,m}) \subset \sigma(\mathbf{W}_{d,0:m}) \ , \ \ \forall m \in [\![0, M]\!] \ , && \text{(10c)} \\
& \sigma(\mathbf{X}_{d+1}) \subset \sigma(\mathbf{W}_{d,0:M}) \ . && \text{(10d)}
\end{aligned}
$$

Problem (10) is less constrained than Problem (9) because the (dynamics) equality constraints (9b) are more binding than the inequality constraints (10b), and also because (9b) implies the new constraint (10d). Since $V_{D+1}^{\mathrm{e}} = V_{D+1}^{\mathrm{i}} = K$, we obtain by backward induction that the Bellman functions (10) associated with Problem (7) are lower bounds to the Bellman functions (9) associated with Problem (6):

$$
V_d^{\mathrm{i}} \leq V_d^{\mathrm{e}} \ , \ \ \forall d \in [\![0, D{+}1]\!] \ . \tag{11}
$$

3.2 Bellman functions under monotonicity-inducing assumption

We introduce assumptions on the data of Problem (6) which allow us to make the link between the sequence of Bellman functions $\{V_d^{\mathrm{e}}\}_{d\in[\![0,D+1]\!]}$ associated with Problem (6) and the sequence of Bellman functions $\{V_d^{\mathrm{i}}\}_{d\in[\![0,D+1]\!]}$ associated with Problem (7).

We first formulate an assumption that we call *monotonicity-inducing assumption* as it is the key ingredient to obtain both the monotonicity of the $\{V_d^{\mathrm{e}}\}_{d\in[\![0,D+1]\!]}$ Bellman functions and, for $d \in [\![0, D{+}1]\!]$, the inequality $V_d^{\mathrm{i}} \geq V_d^{\mathrm{e}}$ — the opposite of inequality (11). It is worth noting that this assumption, which seems an *ad hoc* trick for proving that inequality (11) is in fact an equality, is satisfied in the case study (developed in Sect. 6) which motivates this paper. The fact that this assumption is satisfied for the case study is shown in Appendix C.

Assumption 1 (Monotonicity-inducing) *We assume that the data of Problem* (9) *satisfies the following properties.*

1. *The final cost function K is nonincreasing on its effective domain:*

$$
\forall (x, x') \in (\mathrm{dom} K)^2 \ , \ \ x \leq x' \implies K(x) \geq K(x') \ . \tag{12a}
$$

2. *For all $d \in [\![0, D]\!]$, the effective domain of the Bellman function V_d^{e} is induced by the effective domain of the instantaneous cost function L_d, namely*

$$\mathrm{dom} V_d^{\mathrm{e}} = \{x \in \mathbb{X} \,|\, \exists \, \mathbf{U} \text{ satisfying (9c) s.t. } \mathbb{E}[L_d(x, \mathbf{U}_d, \mathbf{W}_d)] < +\infty\} \ . \tag{12b}$$

3. *For all $d \in [\![0, D]\!]$, for any two states $x' \geq x$ both in $\mathrm{dom} V_d^{\mathrm{e}}$ (where the Bellman function V_d^{e} is given by (9)) and for any (control) random variable $\mathbf{U}_d$ satisfying (9c) and such that $\mathbb{E}\,[L_d(x, \mathbf{U}_d, \mathbf{W}_d)] < +\infty$, there exists a (control) random variable $\widetilde{\mathbf{U}}_d$ satisfying (9c) such that (almost surely)*

$$f_d(x', \widetilde{\mathbf{U}}_d, \mathbf{W}_d) \in \mathrm{dom} V_{d+1}^{\mathrm{e}} \text{ and } f_d(x', \widetilde{\mathbf{U}}_d, \mathbf{W}_d) \geq f_d(x, \mathbf{U}_d, \mathbf{W}_d) \tag{12c}$$

$$L_d(x', \widetilde{\mathbf{U}}_d, \mathbf{W}_d) \leq L_d(x, \mathbf{U}_d, \mathbf{W}_d) \ . \tag{12d}$$

Proposition 2 *We suppose that monotonicity-inducing Assumption 1 holds true. Then, for all $d \in [\![0, D{+}1]\!]$, the (original) Bellman function V_d^{e} given by backward induction (9) is nonincreasing on its effective domain, that is,*

$$\forall d \in [\![0, D{+}1]\!] \ , \ \ \forall (x, x') \in \mathrm{dom} V_d^{\mathrm{e}} \times \mathrm{dom} V_d^{\mathrm{e}} \ , \ \ x \leq x' \implies V_d^{\mathrm{e}}(x) \geq V_d^{\mathrm{e}}(x') \ .$$

Proof. The proof is done by backward induction. At time $D{+}1$, the Bellman function $V_{D+1}^{\mathrm{e}} = K$ is nonincreasing on its effective domain by Condition 1 of Assumption 1. For $d \in [\![0, D]\!]$, assume that V_{d+1}^{e} is nonincreasing on its effective domain. Let $(x, x') \in \mathrm{dom} V_d^{\mathrm{e}} \times \mathrm{dom} V_d^{\mathrm{e}}$ such that $x \leq x'$. For any $\epsilon > 0$, let $\mathbf{U}_d$ be an ϵ-solution of Problem (9) starting at state x. We have that

$$V_d^{\mathrm{e}}(x) + \epsilon \geq \mathbb{E}\left[L_d(x, \mathbf{U}_d, \mathbf{W}_d) + V_{d+1}^{\mathrm{e}}\big(f_d(x, \mathbf{U}_d, \mathbf{W}_d)\big)\right] , \tag{13}$$

which implies $L_d(x, \mathbf{U}_d, \mathbf{W}_d) < +\infty$ and $f_d(x, \mathbf{U}_d, \mathbf{W}_d) \in \mathrm{dom} V_{d+1}^{\mathrm{e}}$, $\mathbb{P}$-a.s.. From Condition 3 of Assumption 1, there exists a random variable $\widetilde{\mathbf{U}}_d$ satisfying the measurability constraint (9c) and satisfying $\mathbb{P}$-a.s. Equations (12c) and (12d). Using the induction assumption and Equation (12c) we obtain that $V_{d+1}^{\mathrm{e}}\big(f_d(x', \widetilde{\mathbf{U}}_d, \mathbf{W}_d)\big) \leq V_{d+1}^{\mathrm{e}}\big(f_d(x, \mathbf{U}_d, \mathbf{W}_d)\big)$ almost surely which, combined with Equation (13), implies that the random variable $V_{d+1}^{\mathrm{e}}\big(f_d(x', \widetilde{\mathbf{U}}_d, \mathbf{W}_d)\big)$ is integrable. Using Equation (12d) combined with Equation (13) we also obtain that the random variable $L_d(x', \widetilde{\mathbf{U}}_d, \mathbf{W}_d)$ is integrable and smaller than $L_d(x, \mathbf{U}_d, \mathbf{W}_d)$. We therefore have

$$\begin{aligned} V_d^{\mathrm{e}}(x) + \epsilon &\geq \mathbb{E}\left[L_d(x, \mathbf{U}_d, \mathbf{W}_d) + V_{d+1}^{\mathrm{e}}\big(f_d(x, \mathbf{U}_d, \mathbf{W}_d)\big)\right] && \text{(by (13))} \\ &\geq \mathbb{E}\left[L_d(x', \widetilde{\mathbf{U}}_d, \mathbf{W}_d) + V_{d+1}^{\mathrm{e}}\big(f_d(x', \widetilde{\mathbf{U}}_d, \mathbf{W}_d)\big)\right] && \text{(as shown above)} \\ &\geq V_d^{\mathrm{e}}(x') \ . && \text{(as } \widetilde{\mathbf{U}}_d \text{ satisfies (9c))} \end{aligned}$$

This ends the proof. □

We are now able to formulate the main proposition of this section.

Proposition 3 *We suppose that monotonicity-inducing Assumption 1 holds true. Then, for any $d \in [\![0, D{+}1]\!]$, the (original) Bellman function V_d^{e} in (9) coincides with the (relaxed) Bellman function V_d^{i} in (10):*

$$V_d^{\mathrm{i}} = V_d^{\mathrm{e}} \ , \ \ \forall d \in [\![0, D{+}1]\!] \ .$$

Proof. By Equation (11), we have that $V_d^{\mathrm{i}} \leq V_d^{\mathrm{e}}$ for all $d \in [\![0, D{+}1]\!]$. To obtain the reverse inequality, we proceed by backward induction. At time $D+1$, the two functions V_{D+1}^{e} and V_{D+1}^{i} are both equal to the function K. Let d be fixed in $[\![0, D]\!]$ and assume that $V_{d+1}^{\mathrm{i}} = V_{d+1}^{\mathrm{e}}$. For any $x \in \mathrm{dom} V_d^{\mathrm{i}}$ and for any $\epsilon > 0$, let $(\mathbf{X}_{d+1}, \mathbf{U}_d)$ be an ϵ-optimal solution of Problem (10). We have that $f_d(x, \mathbf{U}_d, \mathbf{W}_d) \geq \mathbf{X}_{d+1}$ by Equation (10b) and

$$
\begin{aligned}
V_d^{\mathrm{i}}(x) + \epsilon \geq \; & \mathbb{E}\left[L_d(x, \mathbf{U}_d, \mathbf{W}_d) + V_{d+1}^{\mathrm{i}}(\mathbf{X}_{d+1})\right] , \\
= \; & \mathbb{E}\left[L_d(x, \mathbf{U}_d, \mathbf{W}_d) + V_{d+1}^{\mathrm{e}}(\mathbf{X}_{d+1})\right] ,
\end{aligned} \tag{14}
$$

by the induction assumption. From (14), we deduce that $\mathbb{E}\big[L_d(x, \mathbf{U}_d, \mathbf{W}_d)\big] < +\infty$ and that $\mathbf{X}_{d+1} \in \mathrm{dom} V_{d+1}^{\mathrm{e}}$, $\mathbb{P}$-a.s.. Using Condition 2 of Assumption 1 we obtain that $x \in \mathrm{dom} V_d^{\mathrm{e}}$ and using Condition 3 of Assumption 1 with $x' = x$ we obtain that there exists a random variable $\widetilde{\mathbf{U}}_d$ satisfying the measurability constraint (9c) and satisfying both Equations (12c) and (12d). Using Equation (12c) we obtain that, $\mathbb{P}$-a.s.,

$$
f_d(x, \widetilde{\mathbf{U}}_d, \mathbf{W}_d) \geq f_d(x, \mathbf{U}_d, \mathbf{W}_d) \geq \mathbf{X}_{d+1} \text{ and } f_d(x, \widetilde{\mathbf{U}}_d, \mathbf{W}_d) \in \mathrm{dom} V_{d+1}^{\mathrm{e}} . \tag{15}
$$

Now, we obtain successively

$$
\begin{aligned}
V_d^{\mathrm{i}}(x) + \epsilon \geq \; & \mathbb{E}\left[L_d(x, \mathbf{U}_d, \mathbf{W}_d) + V_{d+1}^{\mathrm{e}}(\mathbf{X}_{d+1})\right] && \text{(by (14))} \\
\geq \; & \mathbb{E}\left[L_d(x, \widetilde{\mathbf{U}}_d, \mathbf{W}_d) + V_{d+1}^{\mathrm{e}}(f_d(x, \widetilde{\mathbf{U}}_d, \mathbf{W}_d))\right] && \text{(by (12d), (15) and Proposition 2)} \\
\geq \; & V_d^{\mathrm{e}}(x) . && \text{(as } \widetilde{\mathbf{U}}_d \text{ satisfies (9c))}
\end{aligned}
$$

We thus obtain the reverse inequality $V_d^{\mathrm{i}} \geq V_d^{\mathrm{e}}$, hence the result. □

The issue is that performing the backward induction (10) requires to solve a multistage stochastic optimization problem at the fast time-scale for each $d \in [\![0, D]\!]$ and for each $x \in \mathbb{X}_d$. In the next section, we present two methods to compute bounds of the Bellman functions V_d^{i} at the slow time-scale, that allow to simplify the backward induction.

4 Price/resource decomposition of the dynamics in the Bellman functions

We aim at finding tractable algorithms to numerically solve the backward induction (10) and obtain the corresponding sequence of Bellman functions $\{V_d^{\mathrm{i}}\}_{d \in [\![0, D+1]\!]}$. Indeed, these Bellman functions are not easily obtained. The main issue is that each optimization problem (10) is a multistage stochastic optimization problem at the fast time-scale, that has to be solved for every $d \in [\![0, D]\!]$ and every $x \in \mathbb{X}_d$, and each numerical solving might be in itself hard.

To tackle this issue, we propose in §4.1 and §4.2 to compute respectively lower and upper bounds of the Bellman functions at the slow time-scale. These Bellman functions bounds can then be used to design admissible two-time-scale optimization policies (see Sect. 5). The two algorithms are based on so-called price and resource decomposition techniques (see [5, Chap. 6] and [11]) applied to Problem (10).

Both algorithms involve the computation of auxiliary functions that gather the fast time-scale computations, and that are numerically appealing because they allow to exploit some potential periodicity of two-time-scale problems, as well as parallel computation. This point is developed in Appendix A.

4.1 Lower bounds of the Bellman functions

We present lower bounds for the Bellman functions $\{V_d^{\mathrm{i}}\}_{d\in[\![0,D+1]\!]}$ given by Equation (10). These bounds derive from an algorithm which appears to be connected to the one developed in [13], called "adaptive weights algorithm". We extend the results of [13] in a stochastic setting and in a more general framework, as we are not tied to a battery management problem and as we use a more direct way to reach similar conclusions.

To obtain lower bounds of the sequence $\{V_d^{\mathrm{i}}\}_{d\in[\![0,D+1]\!]}$ of Bellman functions, we dualize the dynamic equations (10b) with Lagrange multipliers, and we use weak duality. The multipliers (called prices here) could be chosen in the class of nonpositive $\mathcal{F}_{[\![0,D]\!]}$-adapted processes but it is enough, to get lower bounds, to stick to deterministic price processes. Following these lines, we obtain a lower bound as follows.

For each $d \in [\![0, D]\!]$, we define the function $L_d^{\mathrm{P}} : \mathbb{X}_d \times \mathbb{Y}_{d+1} \to \mathbb{R}\cup\{\pm\infty\}$ by[4]

$$L_d^{\mathrm{P}}(x_d, p_{d+1}) = \inf_{\mathbf{U}_d} \mathbb{E}\left[L_d(x_d, \mathbf{U}_d, \mathbf{W}_d) + \big\langle p_{d+1},\ f_d(x_d, \mathbf{U}_d, \mathbf{W}_d)\big\rangle\right] , \tag{16a}$$

$$\text{s.t.} \quad \sigma(\mathbf{U}_{d,m}) \subset \sigma(\mathbf{W}_{d,0:m}) \ , \ \ \forall m \in [\![0, M]\!] \ , \tag{16b}$$

where L_d and f_d are respectively the instantaneous cost function and the dynamics of Problem (6).

Proposition 4 *Consider the sequence $\{\underline{V}_d^{\mathrm{P}}\}_{d\in[\![0,D+1]\!]}$ of Bellman functions which is defined by $\underline{V}_{D+1}^{\mathrm{P}} = K$ and for all $d \in [\![0, D]\!]$, and for all $x \in \mathbb{X}_d$ by*[5]

$$\underline{V}_d^{\mathrm{P}}(x) = \sup_{p_{d+1}\le 0} \Big(L_d^{\mathrm{P}}(x, p_{d+1}) - \big(\underline{V}_{d+1}^{\mathrm{P}}\big)^{\star}(p_{d+1})\Big) , \tag{17}$$

where $\big(\underline{V}_{d+1}^{\mathrm{P}}\big)^{\star} : \mathbb{Y}_{d+1} \to \mathbb{R}\cup\{\pm\infty\}$ is the Fenchel conjugate of $\underline{V}_{d+1}^{\mathrm{P}}$ (see [22]). Then, the Bellman functions $\{\underline{V}_d^{\mathrm{P}}\}_{[\![0,D+1]\!]}$ given by Equation (17) are lower bounds of the corresponding Bellman functions $\{V_d^{\mathrm{i}}\}_{d\in[\![0,D+1]\!]}$ given by Equation (10), that is,

$$\underline{V}_d^{\mathrm{P}} \le V_d^{\mathrm{i}} \ , \ \ \forall d \in [\![0, D{+}1]\!] \ . \tag{18}$$

Proof. We start the proof by a preliminary interchange result. We consider a subset $\mathcal{X}$ of the space of random variables taking values in a Borel space $\mathbb{X}$ and a measurable function $\varphi : \mathbb{X} \to \mathbb{R}\cup\{\pm\infty\}$. We assume that $\mathcal{X}$ contains all the constant random variables. We prove that

$$\inf_{\mathbf{X}\in\mathcal{X}} \mathbb{E}\big[\varphi(\mathbf{X})\big] = \inf_{x\in\mathbb{X}} \varphi(x) \ . \tag{19}$$

- The $\le$ inequality $\inf_{\mathbf{X}\in\mathcal{X}} \mathbb{E}\big[\varphi(\mathbf{X})\big] \le \inf_{x\in\mathbb{X}} \varphi(x)$ is clear as $\mathcal{X}$ contains all the constant random variables.
- The reverse inequality holds true if $\inf_{x\in\mathbb{X}} \varphi(x) = -\infty$ since $\inf_{\mathbf{X}\in X} \mathbb{E}\big[\varphi(\mathbf{X})\big] \le \inf_{x\in\mathbb{X}} \varphi(x)$. Assume now that $\inf_{x\in\mathbb{X}} \varphi(x) = \underline{\varphi} > -\infty$. Then $\varphi(\mathbf{X}) \ge \underline{\varphi}$ $\mathbb{P}$-a.s. for all $\mathbf{X} \in \mathcal{X}$ and

[4] recall that $\mathbb{Y}_{d+1} = \mathbb{X}_{d+1} = \mathbb{R}^{n_{d+1}}$

[5] As we manipulate functions with values in $\overline{\mathbb{R}} = [-\infty, +\infty]$, we need to take care with the addition of extended reals. When not explicitly specified, we adopt by default that $+$ is the Moreau lower addition $\dot{+}$ [17], which extends the usual addition to extended reals by $(+\infty)+(-\infty) = (-\infty)+(+\infty) = -\infty$. So Equation (17) has to be understood as $\underline{V}_d^{\mathrm{P}}(x) = \sup_{p_{d+1}\le 0}\Big(L_d^{\mathrm{P}}(x, p_{d+1}) \dot{+} (-\big(\underline{V}_{d+1}^{\mathrm{P}}\big)^{\star}(p_{d+1}))\Big)$.

hence $\inf_{\mathbf{X}\in\mathcal{X}} \mathbb{E}\big[\varphi(\mathbf{X})\big] \geq \underline{\varphi}$. Consider an arbitrary $\epsilon > 0$ and $\mathbf{X}_\epsilon$ such that $\mathbb{E}\big[\varphi(\mathbf{X}_\epsilon)\big] \leq \inf_{\mathbf{X}\in X} \mathbb{E}\big[\varphi(\mathbf{X})\big] + \epsilon$. We successively obtain $\inf_{x\in\mathbb{X}} \varphi(x) = \mathbb{E}\big[\inf_{x\in\mathbb{X}} \varphi(x)\big] \leq \mathbb{E}\big[\varphi(\mathbf{X}_\epsilon)\big] \leq \inf_{\mathbf{X}\in X} \mathbb{E}\big[\varphi(\mathbf{X})\big] + \epsilon$. Thus, the reverse inequality $\inf_{\mathbf{X}\in\mathcal{X}} \mathbb{E}\big[\varphi(\mathbf{X})\big] \geq \inf_{x\in\mathbb{X}} \varphi(x)$ follows, hence the equality in (19).

We turn now to the proof of (18), that we do by backward induction. First, we have that $\underline{V}^{\mathrm{P}}_{D+1} = K = V^{\mathrm{i}}_{D+1}$. Second, consider $d \in [\![0, D]\!]$ and assume that $\underline{V}^{\mathrm{P}}_{d+1} \leq V^{\mathrm{i}}_{d+1}$. Explicitly using the Moreau lower addition $\dot{+}$ (see Footnote 5), we successively have[6]

$$
\begin{aligned}
\underline{V}^{\mathrm{P}}_d(x) &= \sup_{p_{d+1}\leq 0} \Big(L^{\mathrm{P}}_d(x, p_{d+1}) \dot{+} \big(-\big(\underline{V}^{\mathrm{P}}_{d+1}\big)^\star(p_{d+1})\big)\Big) && \text{(by (17))} \\
&= \sup_{p_{d+1}\leq 0} \Big(L^{\mathrm{P}}_d(x, p_{d+1}) \dot{+} \inf_{x_{d+1}} \big(-\langle p_{d+1},\, x_{d+1}\rangle + \underline{V}^{\mathrm{P}}_{d+1}(x_{d+1})\big)\Big) && \text{(definition of } \big(\underline{V}^{\mathrm{P}}_{d+1}\big)^\star) \\
&\leq \sup_{p_{d+1}\leq 0} \Big(L^{\mathrm{P}}_d(x, p_{d+1}) \dot{+} \inf_{x_{d+1}} \big(-\langle p_{d+1},\, x_{d+1}\rangle + \underline{V}^{\mathrm{i}}_{d+1}(x_{d+1})\big)\Big) && \text{(induction assumption)} \\
&= \sup_{p_{d+1}\leq 0} \Big(L^{\mathrm{P}}_d(x, p_{d+1}) \dot{+} \inf_{\mathbf{X}_{d+1}} \mathbb{E}\big[-\langle p_{d+1},\, \mathbf{X}_{d+1}\rangle + \underline{V}^{\mathrm{i}}_{d+1}(\mathbf{X}_{d+1})\big]\Big) && \text{(interchange result)} \\
&\leq \sup_{p_{d+1}\leq 0} \inf_{\mathbf{U}_d} \mathbb{E}\Big[L_d(x_d, \mathbf{U}_d, \mathbf{W}_d) + \big\langle p_{d+1},\, f_d(x, \mathbf{U}_d, \mathbf{W}_d)\big\rangle\Big] \\
&\qquad \dot{+} \inf_{\mathbf{X}_{d+1}} \mathbb{E}\big[-\langle p_{d+1},\, \mathbf{X}_{d+1}\rangle + \underline{V}^{\mathrm{i}}_{d+1}(\mathbf{X}_{d+1})\big]
\end{aligned}
$$

by substituting (16), and by using subadditivity of the infimum operation with respect to the Moreau lower addition $\dot{+}$

$$
\begin{aligned}
&= \sup_{p_{d+1}\leq 0} \inf_{\mathbf{U}_d, \mathbf{X}_{d+1}} \mathbb{E}\Big[L_d(x_d, \mathbf{U}_d, \mathbf{W}_d) + \big\langle p_{d+1},\, f_d(x, \mathbf{U}_d, \mathbf{W}_d) - \mathbf{X}_{d+1}\big\rangle + \underline{V}^{\mathrm{i}}_{d+1}(\mathbf{X}_{d+1})\Big] \\
&\leq \inf_{\mathbf{U}_d, \mathbf{X}_{d+1}} \sup_{p_{d+1}\leq 0} \mathbb{E}\Big[L_d(x_d, \mathbf{U}_d, \mathbf{W}_d) + \big\langle p_{d+1},\, f_d(x, \mathbf{U}_d, \mathbf{W}_d) - \mathbf{X}_{d+1}\big\rangle + \underline{V}^{\mathrm{i}}_{d+1}(\mathbf{X}_{d+1})\Big] \\
&\leq \inf_{\mathbf{X}_{d+1}, \mathbf{U}_d} \mathbb{E}\big[L_d(x, \mathbf{U}_d, \mathbf{W}_d) + V^{\mathrm{i}}_{d+1}(\mathbf{X}_{d+1})\big] \quad \text{s.t} \quad f_d(x, \mathbf{U}_d, \mathbf{W}_d) \geq \mathbf{X}_{d+1} \quad \text{(by weak duality)} \\
&= \underline{V}^{\mathrm{i}}_d(x) \; .
\end{aligned}
$$

This ends the proof. □

Remark 5 *In the presentation above, we could have defined the sequence* $\big\{\underline{V}^{\mathrm{P}}_d\big\}_{d\in[\![0,D+1]\!]}$ *as*

$$
\underline{V}^{\mathrm{P}}_d(x) = \sup_{p_{d+1}\in\mathbb{Y}_{d+1}} \Big(L^{\mathrm{P}}_d(x, p_{d+1}) - \big(\underline{V}^{\mathrm{P}}_{d+1}\big)^\star(p_{d+1})\Big) \; ,
$$

that is, by maximizing over the whole space $\mathbb{Y}_{d+1}$ *instead of the set of nonpositive prices. Then, in Proposition 4, we would have obtained the inequalities*

$$
\underline{V}^{\mathrm{P}}_d \leq V^{\mathrm{e}}_d \; , \;\; \forall d \in [\![0, D{+}1]\!] \; ,
$$

the Bellman functions V^{e}_d *replacing the* V^{i}_d*. We did not do that in order to be coherent with the computation of the upper bounds in the next section, in which using the Bellman functions* V^{i}_d *is mandatory.*

[6]Here below, we use $\dot{+}$ to stress that there might be an addition of two conflicting $\pm\infty$. When we leave the notation $+$, it is because either we sum real numbers or we sum a real number with $\pm\infty$ or we sum elements of $[0, +\infty]$.

4.2 Upper bounds of the Bellman functions

We present upper bounds for the Bellman functions $\{V_d^{\mathrm{i}}\}_{d\in[\![0,D+1]\!]}$ given by Equation (10). They are obtained using a kind of resource decomposition scheme associated with the dynamic equations, that is, by requiring that the state at time $d+1$ be set at a prescribed deterministic value, so that new constraints have to be added. This is made possible by the fact that we relax the almost sure target equality constraint (6b) into the inequality constraint (7b).

We define the function $L_d^{\mathrm{R}} : \mathbb{X}_d \times \mathbb{X}_{d+1} \to [0, +\infty]$ by

$$L_d^{\mathrm{R}}(x_d, r_{d+1}) = \inf_{\mathbf{U}_d} \mathbb{E}\Big[L_d(x_d, \mathbf{U}_d, \mathbf{W}_d)\Big] , \tag{20a}$$

$$\text{s.t.} \quad f_d(x_d, \mathbf{U}_d, \mathbf{W}_d) \geq r_{d+1} , \tag{20b}$$

$$\sigma(\mathbf{U}_{d,m}) \subset \sigma(\mathbf{W}_{d,0:m}) , \ \ \forall m \in [\![0, M]\!] , \tag{20c}$$

where L_d and f_d are respectively the instantaneous cost function and the dynamics of Problem (6). Note that the function L_d^{R} can take the value $+\infty$ since Constraint (20b) may lead to an empty admissibility set. Having replaced the equality constraint (6b) by the inequality constraint (7b) in Problem (7) makes it possible to have the inequality constraint (20b) in the definition of the function L_d^{R}. This last inequality ensures that a random variable is almost surely greater or equal to a deterministic quantity, a much more easier situation that ensuring the equality between a random variable and a deterministic quantity.

Proposition 6 *Consider the sequence $\{\overline{V}_d^{\mathrm{R}}\}_{d\in[\![0,D+1]\!]}$ of Bellman functions defined inductively by $\overline{V}_{D+1}^{\mathrm{R}} = K$ and for all $d \in [\![0, D]\!]$ and for all $x \in \mathbb{X}_d$ by*

$$\overline{V}_d^{\mathrm{R}}(x) = \inf_{r_{d+1}\in\mathbb{X}_{d+1}} \Big(L_d^{\mathrm{R}}(x, r_{d+1}) + \overline{V}_{d+1}^{\mathrm{R}}(r_{d+1})\Big) . \tag{21}$$

Then, the Bellman functions $\{\overline{V}_d^{\mathrm{R}}\}_{[\![0,D+1]\!]}$ given by Equation (21) are upper bounds of the Bellman functions $\{V_d^{\mathrm{i}}\}_{d\in[\![0,D+1]\!]}$ given by Equation (10), that is,

$$V_d^{\mathrm{i}} \leq \overline{V}_d^{\mathrm{R}} , \ \ \forall d \in [\![0, D+1]\!] . \tag{22}$$

Proof. The proof is done by backward induction. We first have that $\overline{V}_{D+1}^{\mathrm{R}} = K = V_{D+1}^{\mathrm{i}}$. Now, consider $d \in [\![0, D]\!]$ and assume that $V_{d+1}^{\mathrm{i}} \leq \overline{V}_{d+1}^{\mathrm{R}}$. We successively have

$$\begin{aligned}
V_d^{\mathrm{i}}(x) &= \inf_{\mathbf{X}_{d+1},\mathbf{U}_d} \mathbb{E}\big[L_d(x, \mathbf{U}_d, \mathbf{W}_d) + V_{d+1}^{\mathrm{i}}(\mathbf{X}_{d+1})\big] \\
&\qquad \text{s.t} \quad f_d(x, \mathbf{U}_d, \mathbf{W}_d) \geq \mathbf{X}_{d+1}, \text{ and (10c)–(10d)}, && \text{(by (10))} \\
&\leq \inf_{r_{d+1}\in\mathbb{X}_{d+1}} \inf_{\mathbf{U}_d} \mathbb{E}\big[L_d(x, \mathbf{U}_d, \mathbf{W}_d)\big] + V_{d+1}^{\mathrm{i}}(r_{d+1}) \\
&\qquad \text{s.t} \quad f_d(x, \mathbf{U}_d, \mathbf{W}_d) \geq r_{d+1} \text{ and (10c)} && \text{(by considering only constant r.v. } \mathbf{X}_{d+1}) \\
&= \inf_{r_{d+1}\in\mathbb{X}_{d+1}} \big(L_d^{\mathrm{R}}(x_d, r_{d+1}) + V_{d+1}^{\mathrm{i}}(r_{d+1})\big) && \text{(by (20))} \\
&\leq \inf_{r_{d+1}\in\mathbb{X}_{d+1}} \big(L_d^{\mathrm{R}}(x_d, r_{d+1}) + \overline{V}_{d+1}^{\mathrm{R}}(r_{d+1})\big) && \text{(induction assumption)} \\
&= \overline{V}_d^{\mathrm{R}}(x) . && \text{(by (21))}
\end{aligned}$$

This ends the proof. □

5 Mixing time block and price/resource decomposition of the dynamics in the Bellman functions

In §5.1, we show how we can design (not necessarily optimal) policies by means of Bellman functions as obtained in Sect. 3 and Sect. 4. In §5.2, we discuss optimality.

5.1 Computation of policies

We assume that we have at disposal Bellman functions $\{\widetilde{V}_d\}_{d\in[\![0,D+1]\!]}$ obtained either by resource decomposition ($\widetilde{V}_d = \overline{V}^{\mathrm{R}}_d$), or by price decomposition ($\widetilde{V}_d = \underline{V}^{\mathrm{P}}_d$). The computation of the $\widetilde{V}_d$'s, that is, the computation of the $\overline{V}^{\mathrm{R}}_d$'s or $\underline{V}^{\mathrm{P}}_d$'s, constitutes the *offline* part of the optimization procedure, as described in Algorithms 1 and 2.

Then, for a given slow time step $d \in [\![0, D]\!]$ and a given current state $x_d \in \mathbb{X}_d$, we can use $\widetilde{V}_{d+1}$ as an approximation of the Bellman function V^{e}_{d+1} in order to state a new fast-time-scale problem starting at d for computing the decisions to apply at d. This constitutes the *online* part of the procedure (in this paper, we do not discuss conditions ensuring that the control below is indeed a random variable, see [8]):

$$\mathbf{U}^*_d \in \operatorname*{arg\,min}_{\mathbf{U}_d} \mathbb{E}\Big[L_d(x_d, \mathbf{U}_d, \mathbf{W}_d) + \widetilde{V}_{d+1}\big(f_d(x_d, \mathbf{U}_d, \mathbf{W}_d)\big)\Big] , \tag{23a}$$

$$\text{s.t.} \quad \sigma(\mathbf{U}_{d,m}) \subset \sigma(\mathbf{W}_{d,0:m}) , \ \forall m \in [\![0, M]\!] . \tag{23b}$$

This fast-time-scale optimization problem can be solved by any method that provides an online policy as presented in [6]. The presence of a final cost $\widetilde{V}_d$ ensures that the effects of decisions made at the fast time-scale are taken into account at the slow time-scale.

Nevertheless, it would be time-consuming to produce online policies using the numerical solving of Problem (23) for every slow time step of the horizon in simulation. We present in the next two paragraphs how to obtain two-time-scale policies with prices or resources in a smaller amount of time.

Obtaining a policy using prices. In the case where we decompose the problem using deterministic prices, we possibly solve Problem (16) for every couple of initial state and deterministic price $(x_d, p_{d+1}) \in \mathbb{X}_d \times \mathbb{Y}_{d+1}$ and for every $d \in [\![0, D]\!]$. This process produces, for each $d \in [\![0, D]\!]$ and for each $(x_d, p_{d+1}) \in \mathbb{X}_d \times \mathbb{Y}_{d+1}$, an optimal policy $\pi^{\mathrm{P}}_d[x_d, p_{d+1}] : \mathbb{W}_d \to \mathbb{U}_d$ and an optimal value $L^{\mathrm{P}}_d(x_d, p_{d+1})$.

At the beginning of a slow time step d in a state $x_d \in \mathbb{X}_d$, we compute a price p_{d+1} solving the following optimization problem

$$p_{d+1} \in \operatorname*{arg\,max}_{p \leq 0} \Big(L^{\mathrm{P}}_d(x_d, p) - \big(\underline{V}^{\mathrm{P}}_{d+1}\big)^\star(p) \Big) . \tag{24}$$

Thanks to this deterministic price p_{d+1}, we apply the corresponding policy $\pi^{\mathrm{P}}_d[x_d, p_{d+1}]$ to simulate decisions and states drawing a scenario w_d out of the random process $\mathbf{W}_d$. The next state x_{d+1} at the beginning of the slow time step $d+1$ is $x_{d+1} = f_d\big(x_d, \pi^{\mathrm{P}}_d[x_d, p_{d+1}](w_d), w_d\big)$.

Obtaining a policy using resources. In the case where we decompose the problem using deterministic resources, we possibly solve Problem (20) for every couple of initial state and deterministic resource $(x_d, x_{d+1}) \in \mathbb{X}_d \times \mathbb{X}_{d+1}$ and for every $d \in [\![0, D]\!]$. This leads, for each $d \in [\![0, D]\!]$ and for each $(x_d, x_{d+1}) \in \mathbb{X}_d \times \mathbb{X}_{d+1}$, to an optimal policy $\pi_d^{\mathrm{R}}[x_d, x_{d+1}] : \mathbb{W}_d \to \mathbb{U}_d$ and to an optimal value $L_d^{\mathrm{R}}(x_d, x_{d+1})$.

At the beginning of a slow time step d in a state $x_d \in \mathbb{X}_d$, we compute a resource (state) x_{d+1} solving the following optimization problem

$$x_{d+1} \in \operatorname*{arg\,min}_{x \in \mathbb{X}_{d+1}} \Big(L_d^{\mathrm{R}}(x_d, x) + \overline{V}_{d+1}^{\mathrm{R}}(x) \Big) , \tag{25}$$

and we apply the corresponding policy $\pi_d^{\mathrm{R}}[x_d, x_{d+1}]$ to simulate decisions and states drawing a scenario w_d out of $\mathbf{W}_d$. The next state x_{d+1} at the beginning of the slow time step $d+1$ is then $x_{d+1} = f_d\big(x_d, \pi_d^{\mathrm{R}}[x_d, x_{d+1}](w_d), w_d\big)$.

5.2 Discussion on optimality

Without any specific assumption (independence, monotonicity), we have obtained by Propositions 4 and 6 that the price Bellman functions $\underline{V}_d^{\mathrm{P}}$ and the resource Bellman functions $\overline{V}_d^{\mathrm{R}}$ are respectively lower and upper bounds for the Bellman functions associated with (relaxed) Problem (7):

$$\underline{V}_d^{\mathrm{P}} \le V_d^{\mathrm{i}} \le \overline{V}_d^{\mathrm{R}} , \ \ \forall d \in [\![0, D{+}1]\!] . \tag{26}$$

Then, if monotonicity-inducing Assumption 1 holds true, we have by Proposition 3 that these price Bellman functions and resource Bellman functions are also lower and upper bounds for the Bellman functions associated with (original) Problem (6):

$$\underline{V}_d^{\mathrm{P}} \le V_d^{\mathrm{e}} \le \overline{V}_d^{\mathrm{R}} , \ \ \forall d \in [\![0, D{+}1]\!] . \tag{27}$$

The link between the optimal value functions V^{e} (resp. V^{i}) and the Bellman functions V_d^{e} (resp. V_d^{e}) is obtained thanks to a specific independence assumption for the noise process $\mathbf{W}$.

Assumption 2 *(White noise assumption) The sequence of random vectors $\{\mathbf{W}_d\}_{d \in [\![0,D]\!]}$ is white, that is, $\big\{\big(\mathbf{W}_{d,0}, \ldots, \mathbf{W}_{d,m}, \ldots, \mathbf{W}_{d,M}\big)\big\}_{d \in [\![0,D]\!]}$ is a sequence of $D{+}1$ independent random vectors.*

Then the Bellman's principle of optimality applies at the slow time-scale for the optimization problem (6), leading to a Stochastic Dynamic Programming equation at the slow time-scale.

Remark 7 *We do not assume that each random vector $\mathbf{W}_d = (\mathbf{W}_{d,0}, \ldots, \mathbf{W}_{d,M})$ is itself composed of independent random variables.*

Proposition 8 *Under the white noise Assumption 2, the optimal value function V^{e} (resp. V^{i}) solution of (original) Problem* (6) *(resp. solution of (relaxed) Problem* (7)*) coincides with the Bellman function V_0^{e} at time $t = 0$ (resp. V_0^{i}) given by Bellman equations* (9) *(resp.* (10)*). More explicitly, we have that*

$$V^{\mathrm{e}}(x) = V_0^{\mathrm{e}}(x) \quad \textit{and} \quad V^{\mathrm{i}}(x) = V_0^{\mathrm{i}}(x) \ , \ \ \forall x \in \mathbb{X}_0 \ .$$

Proof. The fact that the function V^{e} is equal to the function V_0^{e} is a consequence of [9, Proposition 4.1] where the machinery for establishing a Dynamic Programming equation in a two-time-scale multistage stochastic optimization setting is developed. To establish the equality between the functions V^{i} and V_0^{i}, we proceed as follows. First, it is easily established that Problem (7) is equivalent to Problem (28) stated below which involves a new decision process $\mathbf{\Delta} = \{\mathbf{\Delta}_{d+1}\}_{d\in[\![0,D]\!]}$, each control variable $\mathbf{\Delta}_{d+1}$ taking values in $\mathbb{X}_{d+1}$:

$$
\begin{aligned}
V^{\mathrm{i}}(x) = \inf_{\mathbf{X},\mathbf{U},\mathbf{\Delta}} \ & \mathbb{E}\Bigg[\sum_{d=0}^{D} L_d(\mathbf{X}_d,\mathbf{U}_d,\mathbf{W}_d) + K(\mathbf{X}_{D+1})\Bigg] , && \text{(28a)}\\
\text{s.t} \quad & \mathbf{X}_0 = x \ , \ \ \mathbf{X}_{d+1} = f_d(\mathbf{X}_d,\mathbf{U}_d,\mathbf{W}_d) - \mathbf{\Delta}_{d+1} \ , \ \ \forall d \in [\![0,D]\!] \ , && \text{(28b)}\\
& \mathbf{\Delta}_{d+1} \geq 0 \ , \ \ \forall d \in [\![0,D]\!] \ , && \text{(28c)}\\
& \sigma(\mathbf{U}_{d,m}) \subset \mathcal{F}_{d,m} \ , \ \ \forall (d,m) \in [\![0,D]\!] \times [\![0,M]\!] \ , && \text{(28d)}\\
& \sigma(\mathbf{\Delta}_{d+1}) \subset \mathcal{F}_{d,M} \ , \ \ \forall d \in [\![0,D]\!] \ . && \text{(28e)}
\end{aligned}
$$

Second, Problem (28) involves standard equality constraints in the dynamics, so that the machinery developed in [9, Proposition 4.1] applies to it. We therefore obtain a Dynamic Programming equation associated with Problem (28) involving the new decision process $\mathbf{\Delta}$. This last Dynamic Programming equation reduces to the Bellman equation (10) when replacing the extra nonnegative decision variables by inequality constraints. □

As an immediate consequence of Propositions 8 and 3, we obtain the following proposition which is the main result of this section.

Proposition 9 *Suppose that both monotonicity-inducing Assumption 1 and white noise Assumption 2 hold true. Then, the optimal value function V^{e} of Problem* (6) *can be computed by solving Problem* (7) *at the slow time-scale by the Bellman backward induction* (10)*, that is,*

$$
V^{\mathrm{i}}(x) = V_0^{\mathrm{i}}(x) = V_0^{\mathrm{e}}(x) = V^{\mathrm{e}}(x) \ , \ \ \forall x \in \mathbb{X}_0 \ .
$$

Moreover, if both monotonicity-inducing Assumption 1 and white noise Assumption 2 hold true, we have by Proposition 8 that the price Bellman function $\underline{V}_0^{\mathrm{P}}$ and the resource Bellman function $\overline{V}_0^{\mathrm{R}}$ at time $d = 0$ are respectively lower and upper bounds for the optimal value function of Problem (6):

$$
\underline{V}_0^{\mathrm{P}} \leq V^{\mathrm{e}} \leq \overline{V}_0^{\mathrm{R}} \ . \tag{29}
$$

Equation (29) provides an interval in which the optimal value of the original problem (6) lies. But this interval is valid only under the time-block independence Assumption 2. This last assumption is generally not satisfied in practical cases, and we cannot therefore guarantee the quality of the Bellman functions obtained by the price and resource decomposition algorithms. This being so, the price Bellman functions $\underline{V}_d^{\mathrm{P}}$ and the resource Bellman functions $\overline{V}_d^{\mathrm{R}}$ always allow to compute admissible policies for Problem (6), as explained in §5.1.

6 Case study

In this section, we apply the previous theoretical results to a long term aging and battery renewal management problem. In §6.1, we formulate the problem. In §6.2, we simplify the intraday problems. In §6.3, we describe the data used for the numerical experiments. Finally, in §6.4, we sketch how to apply resource and price decomposition algorithms, and we compare the results given by each of these methods.

6.1 Problem formulation

We consider the following energy storage management problem over 20 years. We manage the charge and discharge of an battery every time step m of 30 minutes. A decision of battery replacement is taken every day, so that the number of days considered in the problem is $20 \times 365 = 7300$. Since the number of time steps during a day is $24 \times 2 = 48$, the total number of time steps of the problem is $48 \times 7300 = 350{,}400$. The state of charge of the battery has to remain between prescribed bounds at each time step. We also consider the evolution over time of the amount of remaining exchangeable energy in the battery (related to the number of cycles remaining), that is, the health of the battery. Once this variable reaches zero, the battery is considered unusable. In addition to the battery, the studied system includes a local renewable energy production unit and a local energy consumption: the net demand (consumption minus production) at each time step is an exogenous random variable affecting the system. Finally we pay for the local system energy consumption, that is, net demand minus energy exchanged with the battery. When this quantity is negative (excess energy production), the energy surplus is assumed to be wasted. The aim of the problem is to minimize the energy bill over the whole time horizon by providing an optimal strategy for the storage charge and the battery renewal.

As we are dealing with the energy storage management problem of a battery over a very long term (20 years) involving two time scales, we adopt the notations defined in §2.1. The total number of slow time steps (days) in the time horizon is denoted by $D+1$ ($D = 20 \times 365 = 7300$), and each slow time interval $[d, d+1[$ contains $M+1$ fast time steps (half hour), hence $M + 1 = 24 \times 2 = 48$.

At the fast time-scale, the system control is the energy $\mathbf{U}_{d,m}$ transferred in and out of the battery. We denote the charge of the battery by $\mathbf{U}^{+}_{d,m} = \max\{0, \mathbf{U}_{d,m}\}$, and the discharge of the battery by $\mathbf{U}^{-}_{d,m} = \max\{0, -\mathbf{U}_{d,m}\}$. For all time[7] $(d, m) \in [\![0, D]\!] \times [\![0, M{+}1]\!]$, the state of the battery consists of

- the amount of energy $\mathbf{S}_{d,m}$ in the battery (state of charge), whose dynamics is given by the simple storage dynamics equation

$$\mathbf{S}_{d,m+1} = \mathbf{S}_{d,m} + \rho^{\mathrm{c}} \mathbf{U}^{+}_{d,m} - \rho^{\mathrm{d}} \mathbf{U}^{-}_{d,m} \ , \ \ \forall m \in [\![0, M]\!] \ , \tag{30a}$$

  where ρ^{c} and ρ^{d} are the charge and discharge coefficients of the battery,

- the amount of remaining exchangeable energy $\mathbf{H}_{d,m}$ (health of the battery), with

$$\mathbf{H}_{d,m+1} = \mathbf{H}_{d,m} - \mathbf{U}^{+}_{d,m} - \mathbf{U}^{-}_{d,m} \ , \ \ \forall m \in [\![0, M]\!] \ , \tag{30b}$$

  so that the battery health decreases with any energy exchange,

- the capacity $\mathbf{C}_{d,m}$ of the battery (assumed to be constant at the fast time-scale)

$$\mathbf{C}_{d,m+1} = \mathbf{C}_{d,m} \ , \ \ \forall m \in [\![0, M]\!] \ . \tag{30c}$$

[7] There is here a slight difference with the notations presented in §2.1: we have added a new time step $(d, M{+}1)$ at the end of day d in order to apply the last fast control of day d and the slow control of day $d{+}1$ at distinct time steps, hence the introduction of a fictitious time step — denoted by $(d, M{+}1)$ — between (d, M) and $(d{+}1, 0)$ (see Equation (31) and comments above).

These equations at the fast time-scale are gathered as

$$(\mathbf{S}_{d,m+1}, \mathbf{H}_{d,m+1}, \mathbf{C}_{d,m+1}) = \varphi\big(\mathbf{S}_{d,m}, \mathbf{H}_{d,m}, \mathbf{C}_{d,m}, \mathbf{U}_{d,m}\big) \ , \ \ \forall (d,m) \in [\![0, D]\!] \times [\![0, M]\!] \ . \tag{30d}$$

At the slow time-scale, that is, for each slow time step d, there exists another control $\mathbf{B}_d$ modeling the possible renewal of the battery at the end of the slow time step. To take it into account, we add a fictitious time step $(d, M{+}1)$ between (d, M) and $(d{+}1, 0)$. The dynamics of the battery for this specific time step are

$$\mathbf{S}_{d+1,0} = \begin{cases} 0 & \text{if } \mathbf{B}_d > 0 \ , \\ \mathbf{S}_{d,M+1} & \text{otherwise} \ , \end{cases} \tag{31a}$$

meaning that, when renewed, a new battery is empty,

$$\mathbf{H}_{d+1,0} = \begin{cases} \mathfrak{N}(\mathbf{B}_d)\mathbf{B}_d & \text{if } \mathbf{B}_d > 0 \ , \\ \mathbf{H}_{d,M+1} & \text{otherwise} \ , \end{cases} \tag{31b}$$

meaning that, when renewed, the health of a battery is the product of the new battery capacity $\mathbf{B}_d$ by an integer-valued function $\mathfrak{N} : \mathbb{R}_+ \to \mathbb{N}$ estimated at $\mathbf{B}_d$,

$$\mathbf{C}_{d+1,0} = \begin{cases} \mathbf{B}_d & \text{if } \mathbf{B}_d > 0 \ , \\ \mathbf{C}_{d,M+1} & \text{otherwise} \ , \end{cases} \tag{31c}$$

corresponding to the renewal of the battery. These equations are gathered as:

$$(\mathbf{S}_{d+1,0}, \mathbf{H}_{d+1,0}, \mathbf{C}_{d+1,0}) = \psi\big(\mathbf{S}_{d,M+1}, \mathbf{H}_{d,M+1}, \mathbf{C}_{d,M+1}, \mathbf{B}_d\big) \ , \ \ \forall d \in [\![0, D]\!] \ . \tag{31d}$$

We assume that the initial state of the battery is known: $(\mathbf{S}_{0,0}, \mathbf{H}_{0,0}, \mathbf{C}_{0,0}) = (s_0, h_0, c_0)$.

All the control variables are subject to bound constraints

$$\mathbf{U}_{d,m} \in [\underline{U}, \overline{U}] \ , \ \ \mathbf{B}_d \in [0, \overline{B}] \ , \tag{32a}$$

(with $\underline{U} < 0$ and $\overline{U} > 0$), as well as the state variables:

$$\mathbf{S}_{d,m} \in [0, \xi\, \mathbf{C}_{d,m}] \ , \ \ \mathbf{H}_{d,m} \in \big[0, \mathfrak{N}(\mathbf{C}_{d,m})\, \mathbf{C}_{d,m}\big] \ , \ \ \mathbf{C}_{d,m} \in [0, \overline{B}] \ . \tag{32b}$$

The amount of remaining exchangeable energy $\mathbf{H}_{d,m}$ has to be nonnegative for the battery to operate, and the upper bound on the state of charge $\mathbf{S}_{d,m}$ is a fraction $\xi \in [0,1]$ of the capacity $\mathbf{C}_{d,m}$.

At each fast time step (d,m), a local renewable energy production unit produces energy and a local demand consumes energy: we denote by $\mathbf{D}_{d,m}$ the net demand (consumption minus production) and we suppose that it is an exogenous random variable. The excess energy consumption $(\mathbf{D}_{d,m} + \mathbf{U}^{+}_{d,m} - \mathbf{U}^{-}_{d,m})^{+}$ is paid at a given price $\pi^{e}_{d,m}$, assumed to be deterministic and known, whereas excess energy production is assumed to be wasted. The price $\mathbf{P}^{b}_{d}$ of a new battery is supposed to be random, so that the operating cost L_d during the slow time step d is

$$\sum_{m=0}^{M} \pi^{e}_{d,m}\big(\mathbf{D}_{d,m} + \mathbf{U}^{+}_{d,m} - \mathbf{U}^{-}_{d,m}\big)^{+} + \mathbf{P}^{b}_{d}\mathbf{B}_d \ . \tag{33}$$

The value of battery at the end of the optimization horizon is represented by a cost function K depending on the state of the battery. Then, the objective function to be minimized is the expected sum over the time span of the discounted daily costs (discount factor γ), plus the final cost K. We assume that the effective domain of the final cost K is $\mathbb{R}^3$ and that K is a nonincreasing function. In the numerical application the final cost K is taken identically equal to 0.

Finally, the optimization problem under consideration is

$$\inf_{\{\mathbf{U}_{d,0:M},\mathbf{B}_d\}_{d\in[\![0,D]\!]}} \mathbb{E}\Bigg[\sum_{d=0}^{D}\gamma^d\Big(\sum_{m=0}^{M}\pi^e_{d,m}\big(\mathbf{D}_{d,m}+\mathbf{U}^+_{d,m}-\mathbf{U}^-_{d,m}\big)^+ + \mathbf{P}^b_d\mathbf{B}_d\Big) \\ + K(\mathbf{S}_{D+1,0},\mathbf{H}_{D+1,0},\mathbf{C}_{D+1,0})\Bigg] , \tag{34a}$$

subject, for all $(d,m)\in[\![0,D]\!]\times[\![0,M]\!]$, to state dynamics

$$(\mathbf{S}_{0,0},\mathbf{H}_{0,0},\mathbf{C}_{0,0}) = (s_0,h_0,c_0) , \tag{34b}$$
$$(\mathbf{S}_{d,m+1},\mathbf{H}_{d,m+1},\mathbf{C}_{d,m+1}) = \varphi\big(\mathbf{S}_{d,m},\mathbf{H}_{d,m},\mathbf{C}_{d,m},\mathbf{U}_{d,m}\big) , \tag{34c}$$
$$(\mathbf{S}_{d+1,0},\mathbf{H}_{d+1,0},\mathbf{C}_{d+1,0}) = \psi\big(\mathbf{S}_{d,M+1},\mathbf{H}_{d,M+1},\mathbf{C}_{d,M+1},\mathbf{B}_d\big) , \tag{34d}$$

to bounds constraints

$$\mathbf{S}_{d,m}\in[0,\xi\,\mathbf{C}_{d,m}] ,\ \ \mathbf{H}_{d,m}\in\big[0,\mathfrak{N}(\mathbf{C}_{d,m})\,\mathbf{C}_{d,m}\big] ,\ \ \mathbf{C}_{d,m}\in[0,\overline{B}] , \tag{34e}$$
$$\mathbf{U}_{d,m}\in[\underline{U},\overline{U}] ,\ \ \mathbf{B}_d\in[0,\overline{B}] , \tag{34f}$$

and to nonanticipativity constraints

$$\sigma(\mathbf{U}_{d,m})\subset\sigma\big(\mathbf{D}_{0,0},\dots,\mathbf{D}_{d,m},\mathbf{P}^b_0,\dots,\mathbf{P}^b_{d-1}\big) , \tag{34g}$$
$$\sigma(\mathbf{B}_d)\subset\sigma\big(\mathbf{D}_{0,0},\dots,\mathbf{D}_{d,M},\mathbf{P}^b_0,\dots,\mathbf{P}^b_d\big) . \tag{34h}$$

We denote by $\mathbf{U}_d$ the vector of decision variables to be taken during the slow time step d

$$\mathbf{U}_d = \big(\{\mathbf{U}_{d,m}\}_{m\in[\![0,M]\!]},\mathbf{B}_d\big) . \tag{35a}$$

We also denote by $\mathbf{W}_d$ the vector of noise variables occurring during the slow time step d

$$\mathbf{W}_d = \big(\{\mathbf{D}_{d,m}\}_{m\in[\![0,M]\!]},\mathbf{P}^b_d\big) , \tag{35b}$$

and by $\mathbf{X}_d$ the vector of state variables at the beginning of the slow time step d

$$\mathbf{X}_d = (\mathbf{S}_{d,0},\mathbf{H}_{d,0},\mathbf{C}_{d,0}) . \tag{35c}$$

Problem (34) is amenable to the form (6) given in §2.2, as explained below.

- In the expression of $\mathbf{X}_{d+1}=(\mathbf{S}_{d+1,0},\mathbf{H}_{d+1,0},\mathbf{C}_{d+1,0})$ given by (34d), replacing the variable $\mathbf{S}_{d,m}$ recursively from $m=M+1$ to $m=1$ by using (34c), one obtains a slow-time-scale dynamics of the form (6b):

$$(\mathbf{S}_{d+1,0},\mathbf{H}_{d+1,0},\mathbf{C}_{d+1,0}) = f_d(\mathbf{S}_{d,0},\mathbf{H}_{d,0},\mathbf{C}_{d,0},\mathbf{U}_{d,0:M},\mathbf{B}_d) ,\ \ \forall d\in[\![0,D]\!] . \tag{36}$$

- The cost function of slow time step d in (33) is obviously a function depending on $\mathbf{U}_d$ and $\mathbf{W}_d$. The bound constraints on the control (34f) (resp. the bound constraints on the state (34e)) only depend on $\mathbf{U}_d$ (resp. on $(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d)$: indeed, in the same way we obtained Equation (36), replacing in the right-hand side of (34c) the state variable $\mathbf{S}_{d,m'}$ recursively from $m' = m$ to $m' = 0$ by using (34c), we obtain that the state $(\mathbf{S}_{d,m+1}, \mathbf{H}_{d,m+1}, \mathbf{C}_{d,m+1})$ is a function of $(\mathbf{S}_{d,0}, \mathbf{H}_{d,0}, \mathbf{C}_{d,0}, \mathbf{U}_{d,0:m})$ for all $m \in [\![0, M]\!]$. These constraints are incorporated in the cost of slow time step d (see Remark 1), which makes it an extended real-valued function of the form $L_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d)$ as in (6a). The final cost K is, by definition, a function of $\mathbf{X}_{D+1}$.

- Since $\mathbf{B}_d$ (resp. $\mathbf{P}^b_d$) represents a control (resp. a noise) at the fictitious time step between $(d, M+1)$ and $(d+1, 0)$, the nonanticipativity constraints (34g) – (34h) are of the form (6c).

Thus, Problem (34) fits the framework developed in §4 for two-time-scale optimization problems. Moreover, Assumption 1 is fulfilled for Problem (34) (see Appendix C), so that Proposition 3 applies: relaxing the dynamics (36) as inequality constraints allows to compute price and resource Bellman functions that are lower and upper bounds for the Bellman functions associated with Problem (34).

6.2 Simplifying the intraday problems

We turn now to the computation of the functions L^{P}_d in (16) and L^{R}_d in (20), that we call *intraday functions* in this case study. As explained in §4.1, (resp. §4.2), the intraday functions L^{P}_d (resp. L^{R}_d) depend on the couple (x_d, p_{d+1}) (resp. (x_d, r_{d+1})), namely the 6-uple $(s_d, h_d, c_d, p^s_{d+1}, p^h_{d+1}, p^c_{d+1})$ (resp. $(s_d, h_d, c_d, s_{d+1}, h_{d+1}, c_{d+1})$) in the case study under consideration. We use here some characteristics of the problem to make approximations to alleviate the computation of these intraday functions.

6.2.1 Intraday problem associated with resource decomposition

As explained in §4.2, the aim of the resource decomposition algorithm is to compute, for all slow time steps $d \in [\![0, D{+}1]\!]$, upper bounds $\overline{V}^{\mathrm{R}}_d$ of the Bellman functions associated with Problem (34), which can be put in the form of Problem (6). These upper bounds are obtained by solving a collection of intraday problems such as (20) for each slow time step $d \in [\![0, D]\!]$, and then by solving the Bellman recursion (21). The intraday problems have a priori to be solved for every 6-tuple $(s_d, h_d, c_d, s_{d+1}, h_{d+1}, c_{d+1})$, that is, the state (s_d, h_d, c_d) at the beginning of the slow time step and the resource target $(s_{d+1}, h_{d+1}, c_{d+1})$ at the end of the slow time step. This extremely computationally demanding task is greatly simplified thanks to the following considerations.

Resource intraday function reduction. Since the capacity component $\mathbf{C}_{d,m}$ of the state can only change at the end of a slow time step (see (30c) and (31c)), it is possible to take the capacity dynamics $\mathbf{C}_{d,m}$, the capacity control $\mathbf{B}_d$ and the associated bound constraint, and the cost term $\mathbf{P}^b_d\mathbf{B}_d$, out of the intraday problem and to take them into account in the Bellman recursion. To achieve that, resource decomposition is performed by dealing with Equation (30d) for $m = M$, instead of Equation (31d). We introduce the two resources s_{d+1}

and h_{d+1} for the state of charge and the health of the battery[8]. Then, the intraday problem (20) becomes

$$L^{\mathrm{R}}_d(s_d, h_d, c_d, s_{d+1}, h_{d+1}) = \inf_{\mathbf{U}_{d,0:M}} \mathbb{E}\left[\sum_{m=0}^{M} \pi^e_{d,m} \max(0, \mathbf{D}_{d,m} + \mathbf{U}^+_{d,m} - \mathbf{U}^-_{d,m})\right] , \tag{37a}$$

$$\text{s.t.} \quad (\mathbf{S}_{d,0}, \mathbf{H}_{d,0}) = (s_d, h_d) , \tag{37b}$$

with, for all $m \in [\![0, M]\!]$,

$$\mathbf{S}_{d,m+1} = \mathbf{S}_{d,m} + \rho^{\mathrm{c}}\mathbf{U}^+_{d,m} - \rho^{\mathrm{d}}\mathbf{U}^-_{d,m} , \tag{37c}$$

$$\mathbf{H}_{d,m+1} = \mathbf{H}_{d,m} - \mathbf{U}^+_{d,m} - \mathbf{U}^-_{d,m} , \tag{37d}$$

$$\mathbf{S}_{d,M+1} \geq s_{d+1} , \ \ \mathbf{H}_{d,M+1} \geq h_{d+1} , \tag{37e}$$

$$\mathbf{U}_{d,m} \in [\underline{U}, \overline{U}] , \tag{37f}$$

$$\mathbf{S}_{d,m} \in [0, \xi\, c_d] , \ \ \mathbf{H}_{d,m} \in \big[0, \mathfrak{N}(c_d)\, c_d\big] , \tag{37g}$$

$$\sigma(\mathbf{U}_{d,m}) \subset \sigma\big(\mathbf{D}_{d,0}, \ldots, \mathbf{D}_{d,m}\big) , \tag{37h}$$

and is parameterized by the 5-tuple $(s_d, h_d, c_d, s_{d+1}, h_{d+1})$. The sequence $\{\overline{V}^{\mathrm{R}}_d\}_{d \in [\![0, D+1]\!]}$ of Bellman functions is computed by the following recursion:

$$\overline{V}^{\mathrm{R}}_d(s_d, h_d, c_d) = \inf_{s_{d+1}, h_{d+1}, \mathbf{B}_d} \mathbb{E}\Big[\gamma\big(L^{\mathrm{R}}_d(s_d, h_d, c_d, s_{d+1}, h_{d+1}) + \mathbf{P}^b_d \mathbf{B}_d\big) + \overline{V}^{\mathrm{R}}_{d+1}(\mathbf{S}_{d+1,0}, \mathbf{H}_{d+1,0}, \mathbf{C}_{d+1,0})\Big] , \tag{38a}$$

$$\text{s.t.} \quad (\mathbf{S}_{d+1,0}, \mathbf{H}_{d+1,0}, \mathbf{C}_{d+1,0}) = \psi\big(s_{d+1}, h_{d+1}, c_d, \mathbf{B}_d\big) , \tag{38b}$$

$$s_{d+1} \in [0, \xi\, c_d] , \ \ h_{d+1} \in \big[0, \mathfrak{N}(c_d)\, c_d\big] , \tag{38c}$$

$$\mathbf{B}_d \in [0, \overline{B}] , \tag{38d}$$

$$\sigma(\mathbf{B}_d) \subset \sigma\big(\mathbf{P}^b_d\big) . \tag{38e}$$

In order to further simplify the computation of the intraday functions, we remark that, in the Bellman recursion (38), we can replace the function L^{R}_d by the function $\tilde{L}^{\mathrm{R}}_d$ with

$$\tilde{L}^{\mathrm{R}}_d(s_d, h_d, c_d, s_{d+1}, h_{d+1}) = L^{\mathrm{R}}_d(s_d, h_d, c_d, s_{d+1}, h_{d+1}) + \delta_{[0, \mathfrak{N}(c_d)\, c_d]}(h_d) + \delta_{\{h_{d+1} \leq h_d\}}(h_d, h_{d+1}) + \delta_{[0, \mathfrak{N}(c_d)\, c_d]}(h_{d+1}) ,$$

where δ_A denotes the indicator function of the set A (see Footnote 3 on page 5). Indeed, the last term $\delta_{[0, \mathfrak{N}(c_d)\, c_d]}(h_{d+1})$ is obtained by moving the right-hand side of Constraint (38c) to the minimized cost L^{R}_d and the two other terms can be added as it is easily seen that $L^{\mathrm{R}}_d(s_d, h_d, c_d, s_{d+1}, h_{d+1}) = +\infty$ when $h_{d+1} > h_d$ or when $h_d \notin [0, \mathfrak{N}(c_d)\, c_d]$. Then, it is straightforward to prove that

$$\tilde{L}^{\mathrm{R}}_d(s_d, h_d, c_d, s_{d+1}, h_{d+1}) = \tilde{L}^{\mathrm{R}}_d(s_d, h_d - h_{d+1}, c_d, s_{d+1}, 0) . \tag{39}$$

Indeed, as the health dynamics is linear nonincreasing, any admissible control for Problem (37) for the ordered pair (h_d, h_{d+1}), with $h_{d+1} \leq h_d$ and $(h_d, h_{d+1}) \in \big[0, \mathfrak{N}(c_d)\, c_d\big]^2$ is also admissible for the ordered pair $(h_d - h_{d+1}, 0)$ and conversely. Moreover, the resulting cost is the same since the cost does not depend on the health variable. We thus obtain Equation (39).

[8] We do not associate a resource variable c_{d+1} with the capacity of the battery since this component of the state is taken into account in the Bellman recursion.

Resource intraday function approximation. As suggested in [14], we decide to neglect the state of charge target at the slow time-scale. As a matter of fact, the operation of the battery is daily periodic and such that it is more or less empty at the beginning (and thus at the end) of a slow time step (day). It is thus reasonable to assume that the battery is empty at the beginning and at the end of every slow time step, which is a pessimistic but rather realistic assumption. Combined with Equation (39), we obtain a new function $\widehat{L}_d^{\mathrm{R}}$ approximating the original function L_d^{R}, that is

$$\widehat{L}_d^{\mathrm{R}}(h_d - h_{d+1}, c_d) = \tilde{L}_d^{\mathrm{R}}(0, h_d - h_{d+1}, c_d, 0, 0) \approx \tilde{L}_d^{\mathrm{R}}(s_d, h_d, c_d, s_{d+1}, h_{d+1}) \; . \tag{40}$$

The approximated intraday function $\widehat{L}_d^{\mathrm{R}}$ now only depends on two variables, which significantly reduces the time needed to compute it. Then, the sequence $\{\overline{V}_d^{\mathrm{R}}\}_{d \in [\![0, D+1]\!]}$ of Bellman functions in (38) is approximated by the sequence $\{\widehat{\overline{V}}_d^{\mathrm{R}}\}_{d \in [\![0, D+1]\!]}$ given by the following recursion

$$\widehat{\overline{V}}_d^{\mathrm{R}}(h_d, c_d) = \inf_{h_{d+1}, \mathbf{B}_d} \mathbb{E}\Big[\gamma\big(\widehat{L}_d^{\mathrm{R}}(h_d - h_{d+1}, c_d) + \mathbf{P}_d^b \mathbf{B}_d\big) + \widehat{\overline{V}}_{d+1}^{\mathrm{R}}\big(\psi^{\mathrm{H,C}}(h_{d+1}, c_d, \mathbf{B}_d)\big)\Big] \; , \tag{41a}$$

$$\text{s.t.} \quad h_{d+1} \in \big[0, \mathfrak{N}(c_d)\, c_d\big] \; , \tag{41b}$$

$$\mathbf{B}_d \in [0, \overline{B}] \; , \tag{41c}$$

$$\sigma(\mathbf{B}_d) \subset \sigma\big(\mathbf{P}_d^b\big) \; , \tag{41d}$$

where the new dynamics $\psi^{\mathrm{H,C}}$ is deduced from ψ in (31) by keeping only the last two dynamics (31b) and (31c), which do not depend on the state of charge.

As explained in Appendix A, we consider in this study I periodicity classes ($I = 4$, that is, one class for each season of the year), so that the computation of the resource intraday problem is done only for I different days denoted $d_1, \ldots, d_I$. The complexity of the associated resource decomposition algorithm is sketched in Appendix B.

6.2.2 Intraday problem associated with price decomposition

As detailed in §4.1, the aim of the price decomposition algorithm is to compute, for all slow time steps $d \in [\![0, D+1]\!]$, lower bounds $\underline{V}_d^{\mathrm{P}}$ of the Bellman functions associated with Problem (34). These lower bounds are obtained by solving a collection of intraday problems such as (16) for each slow time step $d \in [\![0, D]\!]$, and then by solving the Bellman recursion (17). The intraday problems have a priori to be solved for every 6-tuple $(s_d, h_d, c_d, p_{d+1}^s, p_{d+1}^h, p_{d+1}^c)$, that is, the state (s_d, h_d, c_d) at the beginning of the slow time step and the prices $(p_{d+1}^s, p_{d+1}^h, p_{d+1}^c)$ associated with the dualization of the equality dynamics equations.

Price intraday function reduction. As in the resource intraday function reduction, it is possible to take the capacity dynamics, its associated control and bound constraints as well as the cost term $\mathbf{P}_d^b \mathbf{B}_d$ out of the intraday problem and to take them into account in the Bellman recursion, so that the intraday problem does not depend on the price p_{d+1}^c associated with the capacity dynamics. To achieve that, price decomposition is not performed on Equation (31d), but on Equation (30d) for $m = M$, which leads to an intraday function whose arguments are $(s_d, h_d, c_d, p_{d+1}^s, p_{d+1}^h)$. But another possible reduction occurs here: from the health dynamics (30b) summed over the fast time steps of day d, we derive the inequality

$$h_d - \mathbf{H}_{d,M+1} - \sum_{m=0}^{M} \big(\mathbf{U}^+_{d,m} + \mathbf{U}^-_{d,m}\big) \geq 0 \; . \tag{42}$$

Following the framework of §4.1, we dualize this induced constraint by incorporating, on the one hand, the terms $\mathbf{U}^+_{d,m} + \mathbf{U}^-_{d,m}$ in the definition of the price intraday function for $m \in [\![0, M]\!]$ and, on the other hand, the term $h_d - \mathbf{H}_{d,M+1}$ in the computation of the Bellman functions. Doing so, the intraday function L^{P}_d does not depend anymore on the health h_d, and is defined as

$$L^{\mathrm{P}}_d(s_d, c_d, p^s_{d+1}, p^h_{d+1}) = \inf_{\mathbf{U}_{d,0:M}} \mathbb{E}\Bigg[\sum_{m=0}^{M} \Big(\pi^e_{d,m} \max(0, \mathbf{D}_{d,m} + \mathbf{U}^+_{d,m} - \mathbf{U}^-_{d,m}) \\ - p^h_{d+1}\big(\mathbf{U}^+_{d,m} + \mathbf{U}^-_{d,m}\big)\Big) + p^s_{d+1}\mathbf{S}_{d,M+1}\Bigg] \; , \tag{43a}$$

subject to, for all $m \in [\![0, M]\!]$,

$$\mathbf{S}_{d,0} = s_d \; , \;\; \mathbf{S}_{d,m+1} = \mathbf{S}_{d,m} + \rho^{\mathrm{c}}\mathbf{U}^+_{d,m} - \rho^{\mathrm{d}}\mathbf{U}^-_{d,m} \; , \tag{43b}$$

$$\mathbf{U}_{d,m} \in [\underline{U}, \overline{U}] \; , \;\; \mathbf{S}_{d,m} \in [0, \xi c_d] \; , \tag{43c}$$

$$\sigma(\mathbf{U}_{d,m}) \subset \sigma\big(\mathbf{D}_{d,0}, \dots, \mathbf{D}_{d,m}\big) \; . \tag{43d}$$

The associated sequence of Bellman functions $\{\underline{V}^{\mathrm{P}}_d\}_{d \in [\![0, D+1]\!]}$ is computed by the following recursion:

$$\underline{V}^{\mathrm{P}}_d(s_d, h_d, c_d) = \sup_{(p^s_{d+1}, p^h_{d+1}) \leq 0} \Bigg(L^{\mathrm{P}}_d(s_d, c_d, p^s_{d+1}, p^h_{d+1}) + \inf_{s_{d,M+1}, h_{d,M+1}} \inf_{\mathbf{B}_d} \mathbb{E}\Big[\gamma \mathbf{P}^b_d \mathbf{B}_d \\ - p^s_{d+1} s_{d,M+1} + \big(p^h_{d+1}\big(h_d - h_{d,M+1}\big) + \underline{V}^{\mathrm{P}}_{d+1}(\mathbf{S}_{d+1,0}, \mathbf{H}_{d+1,0}, \mathbf{C}_{d+1,0})\big)\Big]\Bigg) \; , \tag{44a}$$

$$\text{subject to} \quad (\mathbf{S}_{d+1,0}, \mathbf{H}_{d+1,0}, \mathbf{C}_{d+1,0}) = \psi\big(s_{d,M+1}, h_{d,M+1}, c_d, \mathbf{B}_d\big) \; , \tag{44b}$$

$$\mathbf{B}_d \in [0, \overline{B}] \; , \;\; s_{d,M+1} \in [0, \xi c_d] \; , \;\; h_{d,M+1} \in \big[0, \mathfrak{N}(c_d)\, c_d\big] \; , \tag{44c}$$

$$\sigma(\mathbf{B}_d) \subset \sigma\big(\mathbf{P}^b_d\big) \; . \tag{44d}$$

Price intraday function approximation. As in the resource decomposition algorithm, it is possible to consider that the state of charge of the battery has no influence at the slow time-scale. Doing so, we obtain a new function $\widehat{L}^{\mathrm{P}}_d$ approximating the original function L^{P}_d, that is,

$$\widehat{L}^{\mathrm{P}}_d(c_d, p^h_{d+1}) = L^{\mathrm{P}}_d(0, c_d, 0, p^h_{d+1}) \approx L^{\mathrm{P}}_d(s_d, c_d, p^s_{d+1}, p^h_{d+1}) \; . \tag{45}$$

The approximated price intraday function $\widehat{L}^{\mathrm{P}}_d$ only depends on the 2-tuple (c_d, p^h_{d+1}), which significantly reduces the time needed to compute it. Then, the sequence $\{\underline{V}^{\mathrm{P}}_d\}_{d \in [\![0, D+1]\!]}$ of

Bellman functions in (44) is approximated by the sequence $\{\underline{\widehat{V}}^{\mathrm{P}}_d\}_{d\in[\![0,D+1]\!]}$ given by the following recursion:

$$\underline{\widehat{V}}^{\mathrm{P}}_d(h_d, c_d) = \sup_{p^h_{d+1}\leq 0} \Big(\widehat{L}^{\mathrm{P}}_d(c_d, p^h_{d+1}) + \inf_{h_{d,M+1}} \inf_{\mathbf{B}_d} \mathbb{E}\Big[\gamma \mathbf{P}^b_d \mathbf{B}_d$$

$$+ p^h_{d+1}\big(h_d - h_{d,M+1}\big) + \underline{\widehat{V}}^{\mathrm{P}}_{d+1}\big(\psi^{\mathrm{H,C}}(h_{d,M+1}, c_d, \mathbf{B}_d)\big)\Big]\Big) \; , \tag{46a}$$

$$\text{subject to} \quad h_{d,M+1} \in \big[0, \mathfrak{N}(c_d)\, c_d\big] \; , \tag{46b}$$

$$\mathbf{B}_d \in [0, \overline{B}] \; , \tag{46c}$$

$$\sigma(\mathbf{B}_d) \subset \sigma\big(\mathbf{P}^b_d\big) \; . \tag{46d}$$

As explained in Appendix A, we consider in this study I periodicity classes ($I = 4$), that is, one class for each season of the year), so that the computation of the price intraday problem is done only for I different days denoted $d_1, \ldots, d_I$. The complexity of the associated price decomposition algorithm is sketched in Appendix B.

6.3 Experimental setup

The data used in the application come from case studies provided by a Schneider Electric industrial site, equipped with solar panels and a battery, and submitted to three sources of randomness — namely, solar panels production, electrical demand and prices of batteries per kWh. We present hereby the different parameters of the instance under consideration.

- Horizon: 20 years.
- Fast time step: 30 minutes.
- Slow time step: 1 day.
- Number of time steps: 350,400 (= (24 × 2) × (20 × 365)).
- Battery renewal capacity: between 0 and 1,500 kWh with a increment of 100 kWh.
- Periodicity class: 4 classes, one per trimester of the year.

Data to model the cost of batteries and electricity. For the prices of batteries, we obtained a yearly forecast over 20 years from Statista[9]. We added a Gaussian noise to generate synthetic random batteries prices scenarios. We display in Figure 1 the scenarios we generated. Three scenarios are highlighted in Figure 1; they correspond to the three scenarios we comment in the numerical results in §6.4.

For the price of electricity, we chose a "time of use" tariff defined by three rates:

- an off-peak rate at 0.0255$ between 22:00 and 7:00,
- a shoulder rate at 0.0644$ between 7:00 and 17:00,
- a peak rate at 0.2485$ between 17:00 and 22:00.

[9] https://www.statista.com/statistics/883118/global-lithium-ion-battery-pack-costs/

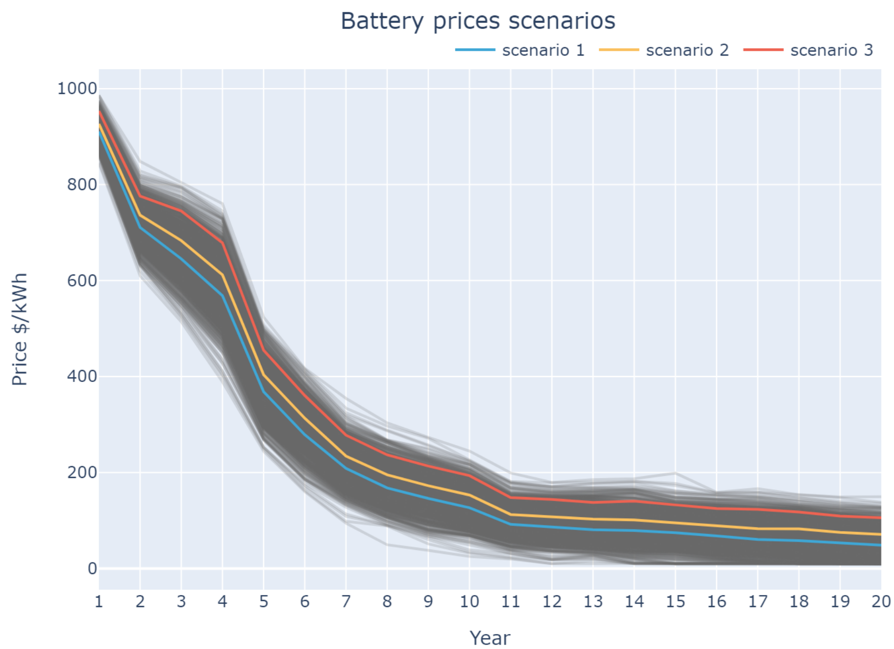

Figure 1: Scenarios of battery prices over a twenty-year timespan

Data to model demand and production. In order to have a realistic dataset in the model described in §6.1, we use the data collected on 70 anonymized industrial sites monitored by Schneider Electric. This data set is openly available.[10] We extracted the data of the site numbered 70. For this site, we display in Figure 2 the half hourly distribution of the net demand (demand minus solar production) during one day.

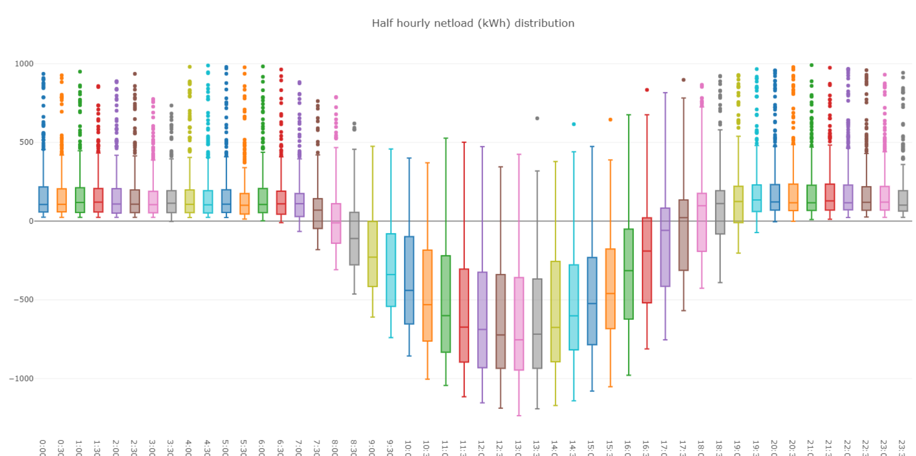

Figure 2: Daily half hourly distribution of net demand (kWh)

Remark 10 ***(About the probabilistic independence of the data).*** *Both batterie prices and net demands correspond to realistic data that are given as scenarios, and there is a*

[10]https://zenodo.org/record/5510400

priori no independence property for these data. Of course, it is possible to compute marginal probability distributions from these scenarios: at a given time step (d, m), collect all the values $\mathbf{D}_{d,m}$ (the value $\mathbf{P}^b_d$ if $m = M{+}1$) available from the scenarios and build a discrete probability distribution from these values. This procedure gives probability distributions at the half-hourly scale. This way of proceeding will be implemented to compute the resource and price intraday functions by Dynamic Programming (see §6.4.1).

6.4 Numerical experiments

The aim of the numerical experiments is to compute and evaluate policies induced by resource (resp. price) decomposition, that is, first solving an approximation of Problem (34) by computing the resource intraday functions $\widehat{L}^{\mathrm{R}}_d$ in (40) and the associated resource Bellman functions $\widehat{\overline{V}}^{\mathrm{R}}_d$ in (41) (resp. price intraday functions $\widehat{L}^{\mathrm{P}}_d$ in (45) and price Bellman functions $\widehat{\underline{V}}^{\mathrm{P}}_d$ in (46)) after all simplifications presented in §6.2.1 (resp. §6.2.2), and second evaluating the policies induced by the Bellman functions $\widehat{\overline{V}}^{\mathrm{R}}_d$ and $\widehat{\underline{V}}^{\mathrm{P}}_d$.

6.4.1 Computation of the resource and price intraday functions

To compute the approximated resource intraday functions $\widehat{L}^{\mathrm{R}}_d$ as given in Equation (40) and the approximated price intraday functions $\widehat{L}^{\mathrm{P}}_d$ as given in Equation (45), we compute the marginal probability distributions of the noises at the fast time-scale as explained in Remark 10 and we apply the Dynamic Programming algorithm. Indeed, computing the intraday functions using Stochastic Programming would be very costly due to the large number of fast time steps inside a slow time step: for example computing a price intraday function (45) would require forming a scenario tree over 48 time steps for every possible (discretized) value of the pair (c_d, p^h_{d+1}) and solving the associated optimization problem, i.e. a task too expensive in computation time.

We recall that the intraday functions are not computed for all possible days in the time horizon, but only for a day representing each periodicity class. Here we split the year of the industrial site data into the four traditional trimesters, each trimester corresponding to one periodicity class. For each trimester, we model the net demand at a given half hour of the day by a discrete random variable with a support of size 10. The probability distribution of each discrete random variable is obtained by clustering, using k-means algorithm, the net demand realizations in the dataset associated with the half hour under consideration.

In the case of resource (resp. price) decomposition, we compute the intraday functions $\widehat{L}^{\mathrm{R}}_d$ (resp. $\widehat{L}^{\mathrm{P}}_d$) for every possible capacity c_d and every possible exchangeable energy $h_d - h_{d+1}$ (resp. every possible price p^h_{d+1}). In this study, the possible values of the capacity c_d are $\{0, 100\ldots, 1500\}$ kWh, whereas the possible values of the price p^h_{d+1} are $\{0, 0.025, 0.05, \ldots, 0.2\}$.

We display in Figure 3 the resource and price intraday functions for each season (trimester) of a year. Resource intraday functions depend on daily exchangeable energy and capacity, whereas price intraday functions depend on the price associated with aging and capacity.

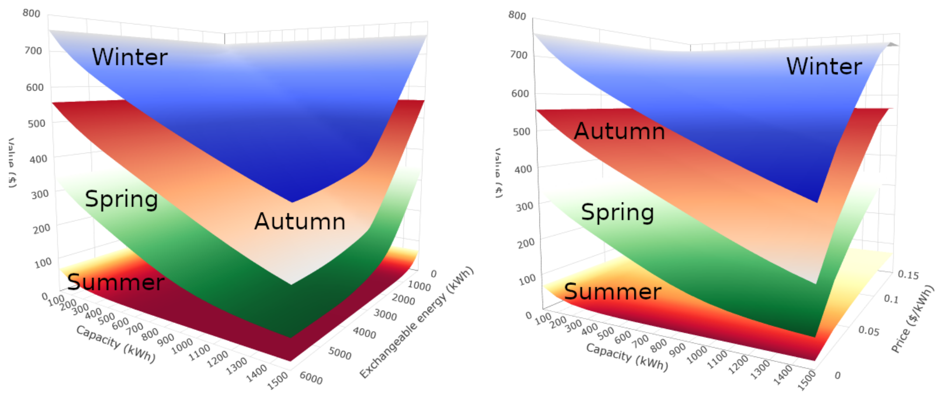

Figure 3: Resource (left) and price (right) intraday functions for each trimester

6.4.2 Computation of the resource and price Bellman functions

Once obtained all possible intraday functions $\widehat{L}_d^{\mathrm{R}}$ and $\widehat{L}_d^{\mathrm{P}}$, the Bellman functions $\widehat{\overline{V}}_d^{\mathrm{R}}$ and $\widehat{\underline{V}}_d^{\mathrm{P}}$ are respectively computed by the Bellman recursions (41) and (46), for $d \in [\![0, D]\!]$. We display in Figure 4 the resource and price Bellman functions obtained for the first day of the time horizon. We observe that the resource and price Bellman functions present approximately

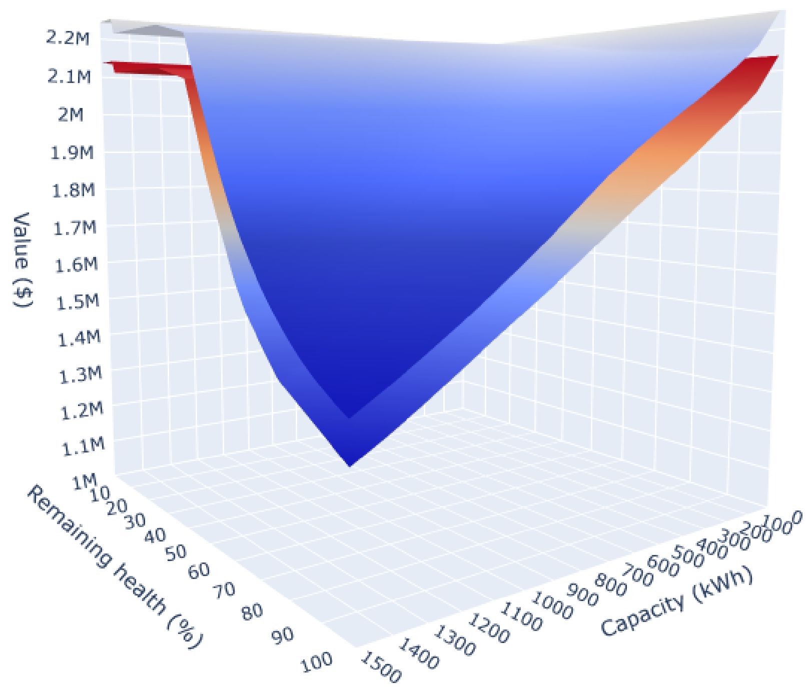

Figure 4: Resource and price Bellman functions at day 1

the same shape and are just separated by a relatively small gap. The same observation holds true for all days of the time span. The largest relative gap between these bounds is 7.90%. The relative gap at the initial state $(s_0, h_0, c_0) = (0, 0, 0)$, that is, the battery no longer works and has to be replaced, is around 4.84%.

We gather in Table 1 the computing times of the two decomposition algorithms, namely the total CPU times and the total wall times[11] when parallelization is on. The computation

	Price	**Resource**
Intraday functions CPU time	1053 s	2836 s
Value functions CPU time	6221 s	1515 s
Total CPU time	7274 s	4351 s
Intraday functions wall time	267 s	714 s
Value functions wall time	2227 s	1310 s
Total wall time	2494 s	2024 s

Table 1: Computing times of the decomposition methods

is run on an Intel i7-8850H CPU @ 2.60GHz 6 cores with 16 GB RAM. Table 1 displays the times needed to compute the intraday functions and the Bellman functions. We observe that, whereas the price decomposition algorithm requires a significantly longer CPU time than the resource decomposition algorithm, the two decomposition algorithms require a comparable wall time when parallelization is on. The main reason is that the parallelization of the computation of price Bellman functions decreases more significantly the computing time than the parallelization for resource Bellman functions. The explanation is that the computation done in parallel is longer in the price case, hence the CPU time saved is not compensated by too frequent memory sharings. The price intraday functions are also faster to compute because the price space is more coarsely discretized than the exchangeable energy space.

Finally, in Table 2, we give the values $\widehat{\overline{V}}_0^{\mathrm{R}}(x_0)$ and $\widehat{\underline{V}}_0^{\mathrm{P}}(x_0)$ of the resource and price Bellman functions at day $d = 0$ for the initial state $x_0 = (s_0, h_0, c_0) = (0, 0, 0)$. According to

	Price	**Resource**
Lower (price) and upper (resource) bounds	2.14 M$	2.24 M$

Table 2: Bounds obtained by resource and price decomposition

Sect. 4, these values are respectively an upper bound and a lower bound of the optimal value of Problem (34). Note however that the numerically computed values given in Table 2 may fail to be upper and lower bounds of the optimal cost of Problem (34) since the resource and price intraday functions are obtained (i) using approximations as explained in §6.2.1 and §6.2.2, and (ii) using the marginal probability distributions of the noises at the fast time-scale (see Remark 10), thus these intraday functions are optimal only if the noises are independent random variables at the fast time-scale.

6.4.3 Simulation of the resource and price policies

We present now several simulation results. Table 3 displays the times needed to perform a 20 years simulation over one scenario of battery prices and net demands, from which we deduce the average time needed to compute a decision at each time step.

[11] Wall time measures how much time has passed for executing the code, as if you were looking at the clock on your wall.

	Price	**Resource**
Average time to simulate a scenario	6.19 s	5.22 s
Average time to compute a decision	17.7 μs	14.9 μs

Table 3: Computing times of simulation

Simulation using scenarios. We draw $1,000$ "true" scenarios of battery prices and net demands over 20 years, that is, scenarios extracted from the realistic data of the problem. There is thus no more independence assumption available for these scenarios. Then, as explained in §5.1, we simulate the charge and renewal decisions that are made when using the intraday functions and the Bellman functions obtained by resource and price decomposition, in order to compare the performances of both methods. All simulations start from the initial state $(s_0, h_0, c_0) = (0, 0, 0)$. The average costs of these scenarios are given in Table 4. The comparison of the average costs shows that both decomposition methods provide comparable performances. However, the price decomposition outperforms the resource decomposition by achieving on average 1.05% of additional economic savings. This slightly superior performance of the price decomposition is observed on every simulation scenario.

	Price	**Resource**
Average cost over $1,000$ true scenarios	2.83 M$	2.86 M$

Table 4: Average simulation costs using original scenarios

We also note that the average costs are 20% to 25% higher than the corresponding values of the Bellman functions at the initial day for the initial state given in Table 2. This is a somewhat surprising result since the values in Table 2 are lower (price) and upper (resource) bounds of the optimal value of the problem, provided that the white noise Assumption 2 is fulfilled. But, as explained at the end of §6.4.2, the intraday functions have been computed by dynamic programming, and thus the values obtained are optimal only if the noises are independent at the fast time-scale. The computation of the price and resource Bellman functions makes use of these intraday functions and moreover are optimal only if the battery prices are day by day independent. However, the simulations are made with scenarios where the net demands and the prices are likely to be strongly correlated, hence the discrepancies.

Analysis of some scenarios. We select three scenarios (the colored scenarios in Figure 1) among the $1,000$ scenarios of battery prices and net demands over 20 years that we used in the previous paragraph, and we analyse the behavior of the policies induced by resource and price decompositions. We recall that all simulations start from the initial state $(s_0, h_0, c_0) = (0, 0, 0)$, that is, the battery no longer works and has to be replaced. Figure 5 displays the health (or exchangeable energy in kWh) of the batteries at the end of each day for the three scenarios, and Table 5 gives the associated simulation costs. In the first simulation, price and resource decompositions lead to significantly different renewal decisions and different costs. A small battery, (100 kWh, that is, 400 kWh of exchangeable energy[12]) is purchased at day $d = 0$ for both price and resource decomposition. But at day $d = 2328$, another small battery is purchased in resource decomposition, whereas a large battery ($1,500$ kWh, that is, 6 MWh of exchangeable energy) is purchased in price decomposition. Then, over

[12]The integer function $\mathfrak{N}$ in Equation (31b) is such that $\mathfrak{N}(100) = \mathfrak{N}(1500) = 4$.

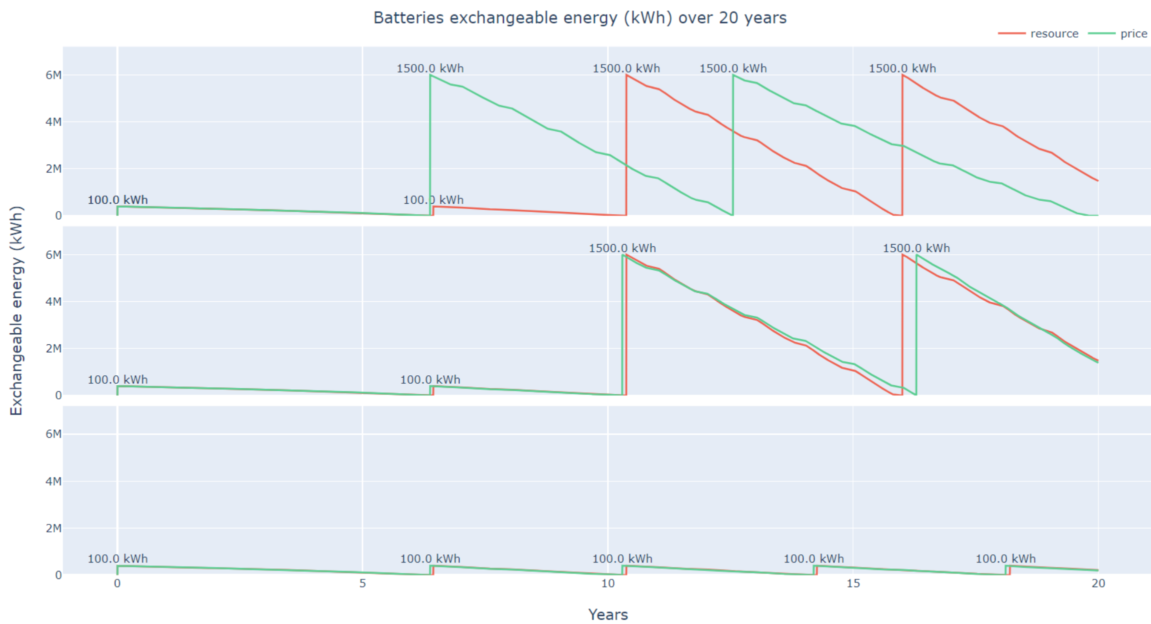

Figure 5: Three simulations of the evolution of the battery health over 20 years

the remaining time horizon, there is one battery renewal in price decomposition and two renewals in resource decomposition, hence a lower cost for price decomposition. In the second simulation, resource and price decompositions produce very similar health trajectories and costs. This is even clearer for the third simulation for which the health trajectories and the costs are almost identical. The third simulation shows a case where battery prices are high, hence only small batteries, that is, 100 kWh, are purchased. Price decomposition outperforms

	Scenario 1	**Scenario 2**	**Scenario 3**
Total cost (resource)	2.757 M$	2.825 M$	3.200 M$
Total cost (price)	2.722 M$	2.820 M$	3.199 M$

Table 5: Simulation results along three scenarios

resource decomposition on the three scenarios, but only by 1.27 % on Scenario 1 while the renewal decisions are significantly different. Our interpretation is that, in Scenario 1, it is almost as rewarding to buy either a big battery or a small battery taking into account the investment. Moreover, it seems that resource decomposition slightly underestimates the benefits of using a large battery compared to a small one. Indeed, we present in Figure 6 the resource and price Bellman functions of the day $d = 2328$ (first battery renewal in Scenario 1), when the health of the battery is fixed to the value $\overline{H}$ associated with a large battery renewal (1,500 kWh), that is, $\widehat{\overline{V}}^{\mathrm{R}}_{2328}(\overline{H}, \cdot)$ in (41) and $\widehat{\underline{V}}^{\mathrm{P}}_{2328}(\overline{H}, \cdot)$ in (46). We observe that the resource Bellman function is significantly higher than the price Bellman function.

Figure 6: Bellman function of first renewal day of Scenario 1 for health fixed at 100 %

7 Conclusion

We have introduced the formal definition of a two-time-scale stochastic optimization problem. The motivation for two-time-scale modeling originated from a battery management problem over a long term horizon (20 years) with decisions being made every 30 minutes (charge/discharge). We have presented two algorithmic methods to compute daily Bellman functions to solve these generic problems — with an important number of time steps and decisions on different time-scales — when they display monotonicity properties. Both methods rely on a Bellman equation applied at the slow time-scale, producing Bellman functions at this scale.

Our first method, called resource decomposition algorithm, is a primal decomposition of the daily Bellman equation that requires to compute the value of a multistage stochastic optimization problem parameterized by a stochastic resource. Some monotonicity properties here make it possible to relax the coupling constraint and to replace the stochastic resource by a deterministic one, yielding an upper bound for the slow time-scale Bellman functions. Instead of this simplification, we could have turned the almost sure coupling constraint into a constraint in expectation. It would be interesting to compare this with our approach.

We address a similar and related difficulty in our price decomposition algorithm. It requires the computation of the value of a stochastic optimization problem parameterized by a stochastic price. Once again we replace it by a deterministic price, which is equivalent to dualize an expectation target constraint. This makes the previous enhancement proposal even more relevant. Our algorithm produces a lower bound for the slow time-scale Bellman functions that reveals to produce better results in simulation than the ones obtained by using the resource Bellman functions (we already have observed this numerical favorable phenomenon in [10]).

Finally, we have shown with a realistic numerical application that these methods make it possible to compute design and control policies for problems with a very large number of time steps. But they could also be used for single-time-scale problems that exhibit monotonicity, periodicity and a large number of time steps.

A Periodicity classes

To compute the upper bound Bellman functions $\{\overline{V}_d^{\mathrm{R}}\}_{[\![0,D+1]\!]}$ given by Equation (21) (respectively the lower bound Bellman functions $\{\underline{V}_d^{\mathrm{P}}\}_{d\in[\![0,D+1]\!]}$ given by Equation (17)), we need to compute the value of the fast-scale optimization problems $L_d^{\mathrm{R}}(x_d, r_{d+1})$ defined by (20) (respectively $L_d^{\mathrm{P}}(x_d, p_{d+1})$ defined by (16)), for all $d \in [\![0, D]\!]$ and for all couples $(x_d, r_{d+1}) \in \mathbb{X}_d \times \mathbb{X}_{d+1}$ (respectively for all couples $(x_d, p_{d+1}) \in \mathbb{X}_d \times \mathbb{Y}_{d+1}$ with $p_{d+1} \leq 0$). The computational cost can be significant as we need to solve a stochastic optimization problem for every couple $(x_d, r_{d+1}) \in \mathbb{X}_d \times \mathbb{X}_{d+1}$, for every couple $(x_d, p_{d+1}) \in \mathbb{X}_d \times \mathbb{Y}_{d+1}$ and for every d in $[\![0, D]\!]$. We present a simplification exploiting a possible periodicity at the fast time-scale.

Proposition 11 *Let $\mathbb{I} \subset [\![0, D]\!]$ be a nonempty subset. Assume that there exists two sets $\mathbb{X}_{\mathbb{I}}$ and $\mathbb{U}_{\mathbb{I}}$ such that, for any $d \in \mathbb{I}$, we have $\mathbb{X}_d = \mathbb{X}_{\mathbb{I}}$ and $\mathbb{U}_d = \mathbb{U}_{\mathbb{I}}$. Moreover, assume that there exists two mappings $L_{\mathbb{I}}$ and $f_{\mathbb{I}}$ such that, for any $d \in \mathbb{I}$, $L_d = L_{\mathbb{I}}$ and $f_d = f_{\mathbb{I}}$. Finally assume that the random vectors $\{\mathbf{W}_d\}_{d\in\mathbb{I}}$ are independent and identically distributed. Then, there exists two functions $L_{\mathbb{I}}^{\mathrm{R}}$ and $L_{\mathbb{I}}^{\mathrm{P}}$ such that the functions L_d^{R} defined by (20) and L_d^{P} defined by (16) satisfy*

$$L_d^{\mathrm{R}} = L_{\mathbb{I}}^{\mathrm{R}} \quad \textit{and} \quad L_d^{\mathrm{P}} = L_{\mathbb{I}}^{\mathrm{P}} \ , \ \ \forall d \in \mathbb{I} \ . \tag{47}$$

Proof. The proof is an immediate consequence of the assumptions made on the functions L_d and f_d and on $\mathbf{W}_d$ for $d \in \mathbb{I}$. □

The nonempty set $\mathbb{I}$ in Proposition 11 is called a *periodicity class*. We denote by I the number of periodicity classes of Problem (6) and by $(\mathbb{I}_1, \ldots, \mathbb{I}_I)$ the periodicity classes (possibly reduced to singletons), that is, the sets of slow time indices that satisfy (47).

Remark 12 *A periodicity property often appears in long term energy management problems with renewable energies, due to yearly seasonality of natural processes such as solar production. In this case, $I < D+1$ and it is enough to solve only I problems at the fast time-scale. When there is no periodicity, $I = D+1$ and the periodicity classes are singletons. In this case, all the fast time-scale problems have to be computed.*

The ways to obtain the upper bound and lower bound Bellman functions are presented in Algorithm 1 and Algorithm 2. They allow to efficiently compute fast-time-scale problems and upper and lower bounds for the Bellman functions using resource and price decompositions.

Algorithm 1 Two-time-scale Dynamic Programming with deterministic resources and periodicity classes

Data: Periodicity classes $(\mathbb{I}_1, \dots, \mathbb{I}_I)$
Result: Upper bound Bellman functions $\left\{\overline{V}_d^{\mathrm{R}}\right\}_{d \in [\![0, D+1]\!]}$
Initialization: $\overline{V}_{D+1}^{\mathrm{R}} = K$
for $i = 1, \dots, I$ **do**
  Choose a $d_i \in \mathbb{I}_i$
  for $(x, r) \in \mathbb{X}_{d_i} \times \mathbb{X}_{d_i+1}$ **do**
    Compute $L_{\mathbb{I}_i}^{\mathrm{R}}(x, r) = L_{d_i}^{\mathrm{R}}(x, r)$ and set $L_d^{\mathrm{R}}(x, r) = L_{\mathbb{I}_i}^{\mathrm{R}}(x, r)$ for all $d \in \mathbb{I}_i$
  end
end
for $d = D, D-1, \dots, 0$ **do**
  for $x_d \in \mathbb{X}_d$ **do**
    Solve $\overline{V}_d^{\mathrm{R}}(x_d) = \inf_{r_{d+1}} L_d^{\mathrm{R}}(x_d, r_{d+1}) + \overline{V}_{d+1}^{\mathrm{R}}(r_{d+1})$
  end
end

Algorithm 2 Two-time-scale Dynamic Programming with deterministic prices and periodicity classes

Data: Periodicity classes $(\mathbb{I}_1, \dots, \mathbb{I}_I)$
Result: Lover bound Bellman functions $\left\{\underline{V}_d^{\mathrm{P}}\right\}_{d \in [\![0, D+1]\!]}$
Initialization: $\underline{V}_{D+1}^{\mathrm{P}} = K$
for $i = 1, \dots, I$ **do**
  Choose a $d_i \in \mathbb{I}_i$
  for $(x, p) \in \mathbb{X}_{d_i} \times \mathbb{Y}_{d_i+1},\ p \leq 0$ **do**
    Compute $L_{\mathbb{I}_i}^{\mathrm{P}}(x, p) = L_{d_i}^{\mathrm{P}}(x, p)$ and set $L_d^{\mathrm{P}}(x, p) = L_{\mathbb{I}_i}^{\mathrm{P}}(x, p)$ for all $d \in \mathbb{I}_i$
  end
end
for $d = D, D-1, \dots, 0$ **do**
  for $x_d \in \mathbb{X}_d$ **do**
    Solve $\underline{V}_d^{\mathrm{P}}(x) = \sup_{p_{d+1} \leq 0} L_d^{\mathrm{P}}(x, p_{d+1}) - \left(\underline{V}_{d+1}^{\mathrm{P}}\right)^{\star}(p_{d+1})$
  end
end

Remark 13 *The interest of Algorithm 1 (resp. 2) is that we can solve the I fast-time-scale problems* (20) *(resp.* (16)*) in parallel and then distribute the numerical solving of these problems across slow time steps. Moreover, we can theoretically apply any stochastic optimization method to solve the fast-time-scale problems. Without stagewise independence assumption at the fast time-scale, we may use Stochastic Programming techniques (for example scenario trees and Progressive Hedging [23]) to solve the fast-time-scale problems. With a stagewise independence assumption at the fast time-scale, we may apply Stochastic Dynamic Programming or Stochastic Dual Dynamic Programming.*

B Complexity of the decomposition algorithms

We compute the complexity of the resource and price decomposition Algorithms 1 and 2 in Appendix A, in terms of number of operations required to implement them, and we compare them to a brute force use of Dynamic Programming.

We denote by N_x^{s} (resp. N_u^{s} and N_w^{s}) the dimension of the space of state (resp. control and noise) variables that change only at the slow time-scale. We denote by N_x^{sf} (resp. N_x^{ff}) the dimension of the space of state variables that change at the fast time-scale with an influence at the slow time-scale (resp. without influence at the slow time-scale), and by N_u^{f} (resp. N_w^{f}) the dimension of the space of control (resp. noise) noise variables that change at the fast time-scale. In numerical applications, we stick to the battery problem under consideration in the paper:

- $N_x^{\mathrm{s}} = 1$ (capacity $\mathbf{C}_d$), $N_x^{\mathrm{sf}} = 1$ (health $\mathbf{H}_{d,m}$), $N_x^{\mathrm{ff}} = 1$ (state of charge $\mathbf{S}_{d,m}$),
- $N_u^{\mathrm{s}} = 1$ (renewal $\mathbf{B}_d$), $N_u^{\mathrm{f}} = 1$ (charge/discharge $\mathbf{U}_{d,m}$)
- $N_w^{\mathrm{s}} = 1$ (price $\mathbf{P}_d^b$), $N_w^{\mathrm{f}} = 1$ (net demand $\mathbf{D}_{d,m}$).

We compute the intraday functions $\widehat{L}_d^{\mathrm{R}}$ and $\widehat{L}_d^{\mathrm{P}}$ by Dynamic Programming. To make things simple, we assume that each one-dimensional variable is discretized in 10 values, and that each elementary minimization is conducted by exhaustive search in the control space. We finally assume that the number $D+1$ of slow time steps and the number $M+1$ of fast time steps in a slow time step are rather large, whereas the number I of periodicity classes is rather small.

We first compute the complexity of solving Problem (34) by Dynamic Programming. Taking into account that the dimension of the state at each time step is $N_x^{\mathrm{ff}} + N_x^{\mathrm{sf}} + N_x^{\mathrm{s}}$ and that the last fictitious time step at the fast time-scale only involves the control $\mathbf{B}_d$ and the noise $\mathbf{P}_d^b$, whereas the $M+1$ previous steps involve the control $(\mathbf{U}_{d,m}^{+}, \mathbf{U}_{d,m}^{+})$ and the noise $\mathbf{D}_{d,m}$, the number of elementary operations required to solve the problem is:

$$
\begin{aligned}
& (D+1)\Big(10^{N_x^{\mathrm{ff}}+N_x^{\mathrm{sf}}+N_x^{\mathrm{s}}+N_u^{\mathrm{s}}+N_w^{\mathrm{s}}} + (M+1)\big(10^{N_x^{\mathrm{ff}}+N_x^{\mathrm{sf}}+N_x^{\mathrm{s}}+N_u^{\mathrm{f}}+N_w^{\mathrm{f}}}\big)\Big) \qquad (48)\\
\approx\ & D\big(10^5 + M10^5\big)\\
\approx\ & DM10^5 \ .
\end{aligned}
$$

B.1 Resource decomposition algorithm

We now compute the complexity of the resource decomposition Algorithm 1, which depends on the complexity of the intraday problem numerical solving, and on the way the Bellman recursion is solved.

- We first have to solve the intraday problem, that is, we compute the optimal value of the intraday problem for each possible value of the initial state (s_d, h_d, c_d) and each possible value of the final resource target $(s_{d+1}, h_{d+1}, c_{d+1})$. We apply the simplifications introduced in §6.2.1.
  - A first simplification arises when separating the slow and fast time-scales, that is, the slow components of the problem (state, control and noise) are no more taken into account in the computation of the optimal value of the intraday problem, which leads to Problem (37).

– A second simplification is based on the assumption that the fast component of the state does not influence the slow dynamics and thus can be set to zero at the beginning and at the end of each slow time step.

– The last simplification is that the resource intraday functions only depends on the health difference during a slow time step.

The resulting intraday problem $\widehat{L}^{\mathrm{R}}_d(\Delta h_d, c_d) = L^{\mathrm{R}}_d(0, \Delta h_d, c_d, 0, 0)$ in Equation (40) is solved by Dynamic Programming involving all fast components of the problem, and has to be computed for all possible values of the capacity c_d (the health difference Δh_d is part of the initial state of the problem and thus the associated complexity is taken into account by Dynamic Programming). The complexity of computing the value $\widehat{L}^{\mathrm{R}}_d$ for a given value of c_d is $(M+1)10^{N^{\mathrm{ff}}_x + N^{\mathrm{sf}}_x + N^{\mathrm{f}}_u + N^{\mathrm{f}}_w}$, and the whole complexity of computing all values of all intraday functions is

$$I(M+1)10^{N^{\mathrm{s}}_x}10^{N^{\mathrm{ff}}_x + N^{\mathrm{sf}}_x + N^{\mathrm{f}}_u + N^{\mathrm{f}}_w} \approx IM10^5 \ , \tag{49}$$

where we recall that I is the number of classes of periodicity of the problem.

- Having at disposal all possible functions $\widehat{L}^{\mathrm{R}}_d$ in (39), we compute the Bellman value functions $\widehat{\overline{V}}^{\mathrm{R}}_d$ given by Equation (41). The complexity associated with that recursion is

$$(D+1)10^{N^{\mathrm{sf}}_x + N^{\mathrm{s}}_x + N^{\mathrm{sf}}_x + N^{\mathrm{s}}_u + N^{\mathrm{s}}_w} \approx D10^5 \ . \tag{50}$$

We are now able to compare the complexity (49)–(50) with the complexity (48) of brute force Dynamic Programming. The resource decomposition algorithm is relevant if the following ratio is small

$$\mathfrak{R}^{\mathrm{R}} = \frac{IM10^5 + D10^5}{DM10^5} \ll 1 \ ,$$

or equivalently

$$\mathfrak{R}^{\mathrm{R}} = \frac{I}{D} + \frac{1}{M} \ll 1 \ . \tag{51}$$

Let us consider the relevance condition (51) in three different situations, with a number I of periodicity classes equal to 4.

1. *Horizon:* 20 *years, fast time step:* 1/2 *hour*, slow time step: 1 day.
   Then, we have $(D; M) = (7{,}300; 48)$ and the ratio in (51) is $\mathfrak{R}^{\mathrm{R}} \approx 1/50$.

2. *Horizon:* 20 *years, fast time step:* 1/2 *hour*, slow time step: 1 week.
   Then, we have $(D; M) = (1{,}040; 336)$ and the ratio in (51) is $\mathfrak{R}^{\mathrm{R}} \approx 1/150$.

3. *Horizon:* 20 *years, fast time step:* 1/2 *hour*, slow time step: 1 month.
   Then, we have $(D; M) = (240; 1{,}440)$ and the ratio in (51) is $\mathfrak{R}^{\mathrm{R}} \approx 1/60$.

This illustrates that the ratio $\mathfrak{R}^{\mathrm{R}}$ is minimal when the two quantities D and IM are of the same order of magnitude,[13] in which case the condition $I \ll D$ ensures large computer time savings.

[13]The solution (D^*, M^*) of the optimization problem $\min_{D,M} \frac{I}{D} + \frac{1}{M}$ subject to $DM = \alpha$ is such that $D^* = IM^*$.

In conclusion, the fact that the resource decomposition algorithm is such that the intraday functions can be computed offline and that the number of these functions is much less than the number of slow time steps makes the method very appealing from the computer time point of view.

Remark 14 *It is interesting to compare this complexity result with the one obtained by only separating the slow and fast time-scales, that is, when implementing the resource decomposition algorithm by computing all possible values of the intraday problem* (37) *and then by computing the Bellman functions using* (38)*. The computation by Dynamic Programming of* $L_d^{\mathrm{R}}(s_d, h_d, c_d, s_{d+1}, h_{d+1})$ *for a given* 3*-tuple* (c_d, s_{d+1}, h_{d+1}) *requires* $(M+1)10^{N_x^{\mathrm{ff}}+N_x^{\mathrm{sf}}+N_u^{\mathrm{f}}+N_w^{\mathrm{f}}}$, *so that the whole complexity of computing the values of all intraday functions is*

$$I(M+1)10^{N_x^{\mathrm{s}}+N_x^{\mathrm{ff}}+N_x^{\mathrm{sf}}}10^{N_x^{\mathrm{ff}}+N_x^{\mathrm{sf}}+N_u^{\mathrm{f}}+N_w^{\mathrm{f}}} \approx IM10^7 .$$

The complexity of the Bellman recursion (38) *is*

$$(D+1)10^{N_x^{\mathrm{ff}}+N_x^{\mathrm{sf}}+N_x^{\mathrm{s}}+N_x^{\mathrm{ff}}+N_x^{\mathrm{sf}}+N_u^{\mathrm{s}}+N_w^{\mathrm{s}}} \approx D10^7 ,$$

so that relevance Condition (51) *for the method to be interesting becomes*

$$10^2\Big(\frac{I}{D}+\frac{1}{M}\Big) \ll 1 ,$$

a less favorable condition than the one obtained where simplifications are taken into account.

B.2 Price decomposition algorithm

We consider now the computation of the intraday problem arising from the price decomposition Algorithm 2 and the associated Bellman recursion. Taking into account all simplifications introduced in §6.2.2, the resulting intraday function $\widehat{L}_d^{\mathrm{P}}(c_d, p_{d+1}^h) = L_d^{\mathrm{P}}(0, c_d, 0, p_{d+1}^h)$ given by (43) is solved by Dynamic Programming involving all fast components of the problem, and has to be computed for all possible values of the capacity c_d and of the multiplier p_{d+1}^h. The complexity of computing the value $\widehat{L}_d^{\mathrm{P}}(c_d, p_{d+1}^h)$ for a given couple (c_d, p_{d+1}^h) is $(M+1)10^{N_x^{\mathrm{ff}}+N_u^{\mathrm{f}}+N_w^{\mathrm{f}}}$, and the whole complexity of computing all values of all intraday functions is

$$I(M+1)10^{N_x^{\mathrm{s}}+N_x^{\mathrm{sf}}}10^{N_x^{\mathrm{ff}}+N_u^{\mathrm{f}}+N_w^{\mathrm{f}}} \approx IM10^5 . \tag{52}$$

Then we compute the Bellman function $\underline{\widehat{V}}_d^{\mathrm{P}}(h_d, c_d)$ given by Equation (46). The complexity associated with the Bellman recursion is

$$(D+1)10^{N_x^{\mathrm{sf}}+N_x^{\mathrm{s}}+N_x^{\mathrm{sf}}+N_x^{\mathrm{sf}}+N_u^{\mathrm{s}}+N_w^{\mathrm{s}}} \approx D10^6 . \tag{53}$$

Comparing the whole complexity of the price decomposition algorithm with the complexity (48) of Dynamic Programming, we conclude that the resource decomposition algorithm is relevant if the following ratio is small

$$\mathfrak{R}^{\mathrm{P}} = \frac{IM10^5 + D10^6}{DM10^5} \ll 1 ,$$

or, equivalently, if

$$\mathfrak{R}^{\mathrm{P}} = \frac{I}{D} + \frac{10}{M} \ll 1 . \tag{54}$$

Compared with the relevance condition (51) obtained for the resource decomposition algorithm, we conclude that the price decomposition algorithm is more demanding than the resource decomposition algorithm.

C Monotonicity-inducing assumption holds true in the battery management problem

Following the notations introduced in Sect. 6, we specialize in §C.1 the Bellman equation (9) to the case of the battery management. In §C.2, we prove that the monotonicity-inducing assumption 1 is fulfilled for the battery management problem.

C.1 Reformulation of Bellman Equation for the battery management problem

We consider the subset of $\mathbb{R}^3$ defined by

$$X = \big\{(s,h,c) \in \mathbb{R}^3 \,\big|\, s \in [0, \xi c] \ , \ \ h \in \big[0, \mathfrak{N}(c)\, c\big] \ , \ \ c \in [0, \overline{B}]\big\} \ . \tag{55}$$

For all $d \in [\![0, D]\!]$, we consider the composed state dynamics defined by

$$f_d\big((s,h,c), u_{d,0:M}, b_d\big) = \psi\Big(\varphi^{\circ}_{d,M}\big((s,h,c), u_{d,0:M}\big), b_d\Big) \ , \tag{56}$$

where $\varphi^{\circ}_{d,0} = \varphi$ and for $m \in [\![0, M{-}1]\!]$

$$\varphi^{\circ}_{d,m+1} : \big((s,h,c), u_{d,0:m+1}\big) \mapsto \varphi\Big(\varphi^{\circ}_{d,m}\big((s,h,c), u_{d,0:m}\big), u_{d,m+1}\Big) \ , \tag{57}$$

with φ and ψ being defined by (30d) and (31d) in §6.1, namely

$$\varphi : \big((s,h,c), u_{d,m}\big) \mapsto \begin{pmatrix} s + \rho^{\mathrm{c}} u^{+}_{d,m} - \rho^{\mathrm{d}} u^{-}_{d,m} \\ h - u^{+}_{d,m} - u^{-}_{d,m} \\ c \end{pmatrix} \ , \tag{58}$$

and

$$\psi : ((s,h,c), b_d) \mapsto \begin{pmatrix} s\mathbf{1}_{\{0\}}(b_d) \\ h\mathbf{1}_{\{0\}}(b_d) + \mathfrak{N}(b_d) b_d \mathbf{1}_{]0,+\infty[}(b_d) \\ c\mathbf{1}_{\{0\}}(b_d) + b_d \mathbf{1}_{]0,+\infty[}(b_d) \end{pmatrix} \ . \tag{59}$$

Finally, for all $d \in [\![0, D]\!]$, the instantaneous cost functions L_d in (33) is[14]

$$\begin{aligned} L_d : \big(x_{d,0}, u_{d,0:M}, b_d, w^1_{d,0:M}, w^2_d\big) \mapsto & \sum_{m=0}^{M} \ell_{d,m}\big(\varphi^{\circ}_{d,m-1}\big(x_{d,0}, u_{d,0:m-1}\big), u_{d,m}, w^1_{d,m}\big) \\ & + w^2_d b_d + \delta_{[0,\overline{B}]}(b_d) \ , \end{aligned} \tag{60}$$

with

$$\ell_{d,m} : \big(x_{d,m}, u_{d,m}, w^1_{d,m}\big) \mapsto \gamma^d \pi_{d,m}(w^1_{d,m} + u^{+}_{d,m} - u^{-}_{d,m}) + \delta_X(x_{d,m}) + \delta_{[\underline{U},\overline{U}]}(u_{d,m}) \ , \tag{61}$$

where $w^1_{d,m}$ is the net demand at time (d,m) and where w^2_d is the battery price.

[14] by convention, $\varphi^{\circ}_{d,-1}$ is such that $\varphi^{\circ}_{d,-1}(x_{d,0}, u_{d,0:m-1}) = x_{d,0}$.

For each $d \in [\![0, D]\!]$, we introduce the *Bellman operator* $\mathcal{B}_d$ defined by

$$\mathcal{B}_d V(x) = \min_{(\lambda_{d,0:M}, \theta_d) \in \Lambda_d \times \Theta_d} \int \Big(L_d(x, \lambda_{d,0:M}, \mu_d, w^1_{d,0:M}, w^2_d) \tag{62}$$

$$+ V\big(f_d(x, \lambda_{d,0:M}, \theta_d)\big)\Big) \mathbb{P}(\,\mathrm{d}(w^1_{d,0:M}, w^2_d))\ , \tag{63}$$

where, in the minimization, we incorporate the nonanticipative contraints, that is, for each $m \in [\![0, M]\!]$, $\lambda_{d,m}$ is a measurable function of $w^1_{d,0:m}$ (encoded in Λ_d) and θ_d is a measurable function of $(w^1_{d,0:m}, w^2_d)$ (encoded in Θ_d).

C.2 A proof that monotonicity-inducing Assumption 1 is satisfied

The main result of this appendix is Proposition 15.

Proposition 15 *Consider the optimization problem* (34) *described in Sect. 6. The data of the associated Bellman functions* (9) *satisfy monotonicity-inducing Assumption 1.*

Proof. As a consequence of the reformulation in §C.1, the Bellman functions $(V^e_d)_{d\in[\![0,D+1]\!]}$ which are solution of the Bellman equation (9) for the case study of Sect. 6 are also solution of the Bellman backward induction defined by $V_{D+1} = K$ and for all $d \in [\![0, D]\!]$ by $V_d = \mathcal{B}_d V_{d+1}$, where the operator $\mathcal{B}_d$ is defined by (63).

As a first result, we prove by backward induction that, for all $d \in [\![0, D]\!]$ we have that $\mathrm{dom} V_d = X$, where the set X is defined in Equation (55). Let $d \in [\![0, D]\!]$ and assume that $X \subset \mathrm{dom} V_{d+1}$. From the special form of the cost function $\ell_{d,m}$ in Equation (61), the effective domain of V_d is included in the set X. Let $x \in X$ be given, and consider the null control, that is, $u_{d,m} = 0$, for all $m \in [\![0, M]\!]$ and $b_d = 0$. The associated cost L_d in Equation (60) is finite and $f_d(x, u_{d,0:M}, b_d) = x \in X$. We thus have that $V_{d+1}(f_d(x, u_{d,0:M}, b_d) = V_{d+1}(x) < +\infty$ by induction assumption. We have obtained that $x \in \mathrm{dom} V_d$. By assumption, we have $X \subset \mathrm{dom} V_{D+1} = \mathbb{R}^3$ (see Section 6), hence $\mathrm{dom} V_d = X$ for all $d \in [\![0, D]\!]$. This proves that Condition 2 of Assumption 1 is fulfilled. Condition 1 of Assumption 1 is satisfied since we have assumed that the final cost function K is identically equal to 0. Finally, Condition 3 of Assumption 1 is satisfied thanks to the two lemmas 18 and 19 given below. □

To prove the two postponed lemmas 18 and 19, we start by a technical result involving the instantaneous dynamics φ in (58) defined at the fast time scale.

Lemma 16 *Consider two states x and x' both in the subset X as defined in Equation* (55) *and such that $x' \geq x$ and consider a control $u \in [\underline{U}, \overline{U}]$ (with $\underline{U} < 0$) such that $\varphi\big(x, u\big) \in X$, where φ is given by Equation* (58)*. Then, there exists a control $u' \in [\underline{U}, \overline{U}]$ satisfying the two following conditions.*

1. *The control u' is in $[\underline{U}, \overline{U}]$, and u' is either equal to u, or u' satisfy $0 \leq u' < u$ and is given as a feedback on the state variable x';*[15]
2. *The control u' is such that $\varphi\big(x', u'\big) \in X$ and $\varphi\big(x', u'\big) \geq \varphi\big(x, u\big)$.*

Proof. As $x' = (s', h', c') \geq x = (s, h, c)$ we immediately have that $\varphi((s', h', c'), u) \geq \varphi((s, h, c), u)$. Now, we consider two cases. First, if $\varphi((s', h', c'), u) \in X$, then the two conditions of the Lemma are immediately satisfied with $u' = u$. Second, we assume that $\varphi((s', h', c'), u) \notin X$, that is $s' + \rho^c u^+ -$

[15] implying that u' can only differ from u when u is positive.

$\rho^{\mathrm{d}} u^- \notin [0, \xi\, c']$ or $h' - u^+ - u^- \notin \big[0, \mathfrak{N}(c')\, c'\big]$ or $c' \notin [0, \overline{B}]$. We notice that $c' \notin [0, \overline{B}]$ is not possible since we have assumed that $(s', h', c') \in X_t$. Moreover the assertion $h' - u^+ - u^- \notin \big[0, \mathfrak{N}(c')\, c'\big]$ is also not possible as we have

$$
\begin{aligned}
0 &\leq h - u^+ - u^- && \text{(by assumption } \varphi(x,u) \in X\text{)} \\
&\leq h' - u^+ - u^- && \text{(as } h \leq h'\text{)} \\
&\leq \mathfrak{N}(c')\, c' \;. && \text{(as } h' \in \big[0, \mathfrak{N}(c')\, c'\big] \text{ and } u^+ + u^- \geq 0\text{)}
\end{aligned}
$$

Thus we have that $s' + \rho^{\mathrm{c}} u^+ - \rho^{\mathrm{d}} u^- \notin [0, \xi\, c']$. As the lower bound is satisfied since we have that $s' + \rho^{\mathrm{c}} u^+ - \rho^{\mathrm{d}} u^- \geq s + \rho^{\mathrm{c}} u^+ - \rho^{\mathrm{d}} u^- \geq 0$, we must have the following inequality $s' + \rho^{\mathrm{c}} u^+ - \rho^{\mathrm{d}} u^- > \xi\, c'$. which combined with the fact that $s' \in [0, \xi\, c']$ implies that the control u must be positive ($u^- = 0$). We therefore end with $s' + \rho^{\mathrm{c}} u > \xi\, c'$. We define a new control u' defined by

$$
u' = \frac{\xi c' - s'}{\rho^{\mathrm{c}}} < u \;.
$$

This control u' is nonnegative and less than u and therefore satisfies $u' \in [\underline{U}, \overline{U}]$ (as $\underline{U} < 0$). This gives Condition 1 of the Lemma.

Now we have that $\varphi((s', h'), u', w) \in X$ as we have $s' + \rho^{\mathrm{c}} u'^+ - \rho^{\mathrm{d}} u'^- = \xi\, c' \in [0, \xi\, c']$ and $0 \leq h' - u^+ - u^- \leq h' - u'^+ - u'^- \leq \mathfrak{N}(c')\, c'$ and we easily obtain that

$$
\varphi((s', h', c'), u') = \begin{pmatrix} \xi\, c' \\ h' - u'^+ - u'^- \\ c' \end{pmatrix} \geq \begin{pmatrix} s + \rho^{\mathrm{c}} u^+ - \rho^{\mathrm{d}} u^- \\ h - u^+ - u^- \\ c \end{pmatrix} = \varphi((s, h, c), u) \;, \tag{64}
$$

which gives Condition 2 of the Lemma, and ends the proof. □

We now give a lemma involving the full dynamics f_d defined at the slow time scale.

Definition 17 *Consider the set X defined by Equation (55). We say that a given ordered pair of controls $(u_{d,0:M}, b_d)$ is* X-preserving *for the state x if, for all $m \in [\![0, M]\!]$, we have $\varphi^{\circ}_{d,m}\big(x, u_{d,0:m}\big) \in X$ and $f_d\big(x, u_{d,0:M}, b_d\big) \in X$.*

Lemma 18 *Consider two initial states $x'_{d,0} \geq x_{d,0}$ both in X and assume that the controls $(u_{d,0:M}, b_d)$ are X-preserving for $x_{d,0}$ and such that for all $m \in [\![0, M]\!]$, $u_{d,m} \in [\underline{U}, \overline{U}]$. Then, there exists controls $u'_{d,0:M}$ satisfying the two following conditions.*

1. *For all $m \in [\![0, M]\!]$, the control $u'_{d,m}$ is in $[\underline{U}, \overline{U}]$ and $u'_{d,m}$ is either equal to $u_{d,m}$ or is given as a feedback on the state variable $x'_{d,m} = \varphi^{\circ}_{d,m-1}(x'_{d,0}, u'_{d,0:m-1})$ and satisfy $0 \leq u'_{d,m} < u_{d,m}$;*[16]

2. *The controls $(u'_{d,0:M}, b_d)$ are X-preserving for $x'_{d,0}$ and such that*

$$
f_d\big(x'_{d,0}, u'_{d,0:M}, b_d\big) \geq f_d\big(x_{d,0}, u_{d,0:M}, b_d\big) \;.
$$

Proof. We consider two states $x_{d,0}$ and $x'_{d,0}$ both in X and such that $x'_{d,0} \geq x_{d,0}$ and we assume that the controls $(u_{d,0:M}, b_d)$ are X-preserving for $x_{d,0}$.

We start by proving an auxiliary result. We prove by induction that for all $m \in [\![0, M]\!]$ there exists controls $u'_{d,0:m}$ such that

$$
\varphi^{\circ}_{d,m}(x'_{d,0}, u'_{d,0:m}) \geq \varphi^{\circ}_{d,m}(x_{d,0}, u_{d,0:m}) \text{ and } \varphi^{\circ}_{d,m}(x'_{d,0}, u'_{d,0:m}) \in X \;. \tag{65}
$$

[16]implying that $u'_{d,m}$ can only differ from $u_{d,m}$ when $u_{d,m}$ is positive.

First, for $m=0$, we have that $\varphi^{\circ}_{d,0}=\varphi$ and the existence of $u'_{d,0}$ satisfying Equation (65) follows from Lemma 16.

Second, we assume that Equation (65) is satisfied for m and we prove that it is satisfied for $m{+}1$ as follows. Consider $x_{d,m+1}=\varphi^{\circ}_{d,m}(x_{d,0},u_{d,0:m})$ and $x'_{d,m+1}=\varphi^{\circ}_{d,m}(x'_{d,0},u'_{d,0:m})$. By induction assumption we have that both $x_{d,m+1}$ and $x'_{d,m+1}$ are in X and they are such that $x'_{d,m+1}\geq x_{d,m+1}$. Now using Equation (57), we have that

$$\varphi^{\circ}_{d,m+1}(x_{d,0},u_{d,0:m+1})=\varphi\big(\varphi^{\circ}_{d,m}\big(x_{d,0},u_{d,0:m}\big),u_{d,m+1}\big)=\varphi(x_{d,m+1},u_{d,m+1}),$$

and similarly $\varphi^{\circ}_{d,m+1}\big(x'_{d,0},(u'_{d,0:m},u_{d,m+1})\big)=\varphi(x'_{d,m+1},u_{d,m+1})$. Using again Lemma 16, we obtain a new control $u'_{d,m+1}$ such that $\varphi(x'_{d,m+1},u'_{d,m+1})\geq\varphi(x_{d,m+1},u_{d,m+1})$ and $\varphi(x'_{d,m+1},u'_{d,m+1})\in X$. Now, the sequence $u'_{d,0:m+1}=(u'_{d,0:m},u'_{d,m+1})$ gives the induction assumption for $m{+}1$. This ends the proof of the auxiliary result.

We now turn to the proof of the Lemma. We consider the controls $u'_{d,0:M}$ given by the auxiliary result and then use the control b_d. Using the auxiliary result, we obtain that, $\varphi^{\circ}_{d,m}(\ x'_{d,0},\ u'_{d,0:m})\in X$ for all $m\in[\![0,M]\!]$. Thus it remains to prove that

$$f_d(x'_{d,0},u'_{d,0:M},b_d)\geq f_d(x_{d,0},u_{d,0:M},b_d)\text{ and }f_d(x'_{d,0},u'_{d,0:M},b_d)\in X\ . \tag{66}$$

For that purpose, we consider two cases. First, if $b_d=0$, we have that $f_d(\cdot,\cdot,b_d)=\varphi^{\circ}_{d,M}(\cdot,\cdot)$ and thus Equation (66) also follows from the auxiliary result. Second, if $b_d>0$ we obtain immediately that have that $f_d(x',u_{d,0:M},b_d)=f_t(x,u_{d,0:M},b_d)=(0,\mathfrak{N}(b_d)b_d,b_d)\in X$ which also gives Equation (66).

The last assertions of the lemma follow easily as the controls $u'_{d,0:M}$ are build through successive use of Lemma 16 which provide an explicit value for each $u'_{d,m}$ controls satisfying the requested properties listed in the lemma. □

In the following lemma 19 we show that the controls $u'_{d,0:M}$ given by Lemma 18 induce a monotonicity property on the cost L_d.

Lemma 19 *Let $d\in[\![0,D]\!]$ and consider the cost function L_d given by Equation (60). Consider two states $x'_{0,d}\geq x_{0,d}$ both in X and controls $(u_{d,0:M},b_d)$ which are X-preserving for $x_{d,0}$ and such that, for all $m\in[\![0,M]\!]$, $u_{d,m}\in[\underline{U},\overline{U}]$ and $b_d\in[0,\overline{B}]$.*

Then, the controls $(u'_{d,0:M},b_d)$ given by Lemma 18 are such that

$$\forall(w^1_{d,0:M},w^2_d)\ ,\ \ L_d\big(x_{d,0},u_{d,0:M},b_d,w^1_{d,0:M},w^2_d\big)\geq L_d\big(x'_{d,0},u'_{d,0:M},b_d,w^1_{d,0:M},w^2_d\big)\ . \tag{67}$$

Proof. We successively have

$$\begin{aligned}
&L_d\big(x_{d,0},u_{d,0:M},b_d,w^1_{d,0:M},w^2_d\big)\\
&\quad=\sum_{m=0}^{M}\gamma^d\pi_{d,m}(w^1_{d,m}+u^+_{d,m}-u^-_{d,m})+\delta_X\big(\varphi^{\circ}_{d,m}\big(x_{d,0},u_{d,0:m-1}\big)\big)+\delta_{[\underline{U},\overline{U}]}(u_{d,m})+w^2_db_d+\delta_{[0,\overline{B}]}(b_d)\\
&\quad=\sum_{m=0}^{M}\gamma^d\pi_{d,m}(w^1_{d,m}+u^+_{d,m}-u^-_{d,m})+w^2_db_d\\
&\qquad\qquad\qquad\text{(as $u_{d,0:M}$ is X-preserving for $x_{d,0}$ and $u_{d,m}\in[\underline{U},\overline{U}]$ and $b_d\in[0,\overline{B}]$)}\\
&\quad\leq\sum_{m=0}^{M}\gamma^d\pi_{d,m}(w^1_{d,m}+(u'_{d,m})^+-(u'_{d,m})^-)+w^2_db_d\\
&\qquad\qquad\qquad\text{(as when $u'_{d,m}\neq u_{d,m}$ we have $0\leq u'_{d,m}\leq u_{d,m}$ by Lemma 16)}\\
&\quad=L_d\big(x'_{d,0},u'_{d,0:M},b_d,w^1_{d,0:M},w^2_d\big)\ .\\
&\qquad\qquad\qquad\text{(as $u'_{d,0:M}$ is X-preserving for $x'_{d,0}$ and $u'_{d,m}\in[\underline{U},\overline{U}]$ and $b_d\in[0,\overline{B}]$)}
\end{aligned}$$

This ends the proof. □

References

[1] H. Abgottspon. *Hydro Power planning: Multi-horizon modeling and its applications.* PhD thesis, ETH Zürich, 08 2015.

[2] H. Abgottspon and G. Andersson. Medium-term optimization of pumped hydro storage with stochastic intrastage subproblems. In *2014 Power Systems Computation Conference*, pages 1–7, 2014.

[3] H. Abgottspon and G. Andersson. Multi-horizon modeling in hydro power planning. *Energy Procedia*, 87:2–10, 2016.

[4] R. Bellman. *Dynamic Programming.* Princeton University Press, New Jersey, 1957.

[5] D. P. Bertsekas. *Nonlinear programming.* Athena Scientific Belmont, 1999.

[6] D. P. Bertsekas. Dynamic programming and suboptimal control: A survey from ADP to MPC. *European Journal of Control*, 11(4-5):310–334, 2005.

[7] D. P. Bertsekas. *Dynamic programming and optimal control. Vol. I.* Athena Scientific, Belmont, MA, fourth edition, 2017.

[8] D. P. Bertsekas and S. E. Shreve. *Stochastic Optimal Control: The Discrete-Time Case.* Athena Scientific, Belmont, Massachusetts, 1996.

[9] P. Carpentier, J.-P. Chancelier, M. De Lara, T. Martin, and T. Rigaut. Time Block Decomposition of Multistage Stochastic Optimization Problems. *Journal of Convex Analysis*, 30(2), 2023.

[10] P. Carpentier, J.-P. Chancelier, M. De Lara, and F. Pacaud. Mixed spatial and temporal decompositions for large-scale multistage stochastic optimization problems. *Journal of Optimization Theory and Applications*, 186(3):985–1005, 2020.

[11] P. Carpentier and G. Cohen. *Décomposition-coordination en optimisation déterministe et stochastique.* Springer-Verlag, Berlin, 2017.

[12] P. Haessig, H. B. Ahmed, and B. Multon. Energy storage control with aging limitation. In *PowerTech, 2015 IEEE Eindhoven*, pages 1–6. IEEE, 2015.

[13] B. Heymann and P. Martinon. Optimal battery aging : an adaptive weights dynamic programming algorithm. *Journal of Optimization Theory and Applications*, 179(3):1043–1053, 2018.

[14] M. Kaut, K. T. Midthun, A. S. Werner, A. Tomasgard, L. Hellemo, and M. Fodstad. Multi-horizon stochastic programming. *Computational Management Science*, 11(1–2):179–193, 2014. Special Issue: Computational Techniques in Management Science.

[15] V. Leclère, P. Carpentier, J.-P. Chancelier, A. Lenoir, and F. Pacaud. Exact converging bounds for stochastic dual dynamic programming via Fenchel duality. *SIAM Journal on Optimization*, 30(2):1223–1250, 2020.

[16] F. Maggioni, E. Allevi, and A. Tomasgard. Bounds in multi-horizon stochastic programs. *Annals of Operations Research*, pages 1–21, 2019.

[17] J. J. Moreau. Inf-convolution, sous-additivité, convexité des fonctions numériques. *J. Math. Pures Appl. (9)*, 49:109–154, 1970.

[18] M. V. Pereira and L. M. Pinto. Multi-stage stochastic optimization applied to energy planning. *Math. Program.*, 52(1-3):359–375, May 1991.

[19] A. Philpott, F. Wahid, and F. Bonnans. MIDAS: A Mixed Integer Dynamic Approximation Scheme. Research report, Inria Saclay Ile de France, June 2016.

[20] W. B. Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. John Wiley & Sons, 2007.

[21] G. Pritchard, A. B. Philpott, and P. J. Neame. Hydroelectric reservoir optimization in a pool market. *Mathematical programming*, 103(3):445–461, 2005.

[22] R. T. Rockafellar. *Convex analysis.* Princeton university press, 2015.

[23] R. T. Rockafellar and R. J.-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of operations research*, 16(1):119–147, 1991.

[24] C. Skar, G. Doorman, G. A. Pérez-Valdés, and A. Tomasgard. A multi-horizon stochastic programming model for the European power system. *Subimtted to an international peer reviewed journal, In review*, 2016.

[25] A. S. Werner, A. Pichler, K. T. Midthun, L. Hellemo, and A. Tomasgard. *Risk Measures in Multi-Horizon Scenario Trees*, pages 177–201. Springer US, Boston, MA, 2013.

[26] J. Zou, S. Ahmed, and X. A. Sun. Stochastic dual dynamic integer programming. *Mathematical Programming*, Mar 2018.