



Decomposition methods for monotone two-time-scale stochastic optimization problems

Tristan Rigaut¹ · Pierre Carpentier² · Jean-Philippe Chancelier³ · Michel De Lara³

Received: 14 September 2023 / Accepted: 15 March 2024 / Published online: 21 April 2024
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

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.

Keywords Dynamic programming · Decomposition methods · Long-term battery management · Multi-horizon · Two-time-scale

✉ Jean-Philippe Chancelier
jean-philippe.chancelier@enpc.fr

Tristan Rigaut
tristan.rigaut@gmail.com

Pierre Carpentier
pierre.carpentier@ensta-paris.fr

Michel De Lara
michel.delara@enpc.fr

¹ Schneider Electric, Grenoble, France

² UMA, ENSTA Paris, Palaiseau, France

³ CERMICS, École des Ponts ParisTech, Champs-sur-Marne, France

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 15 or 30 min) 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 (Bellman 1957) is a standard method to solve a multistage stochastic optimization problem by time decomposition. This method suffers the so called *curse of dimensionality* as introduced in Bellman (1957), Bertsekas (2017), Powell (2007). In particular the complexity of the most classical implementation of SDP (that discretizes the state space) is exponential in the number of state variables.

A major contribution to handle a large number of state variables is the well-known Stochastic Dual Dynamic Programming (SDDP) algorithm Pereira and Pinto (1991). 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) Philpott et al. (2016) or Stochastic Dual Dynamic Integer Programming (SDDiP) Zou et al. (2018) for nonconvex problems, in particular those displaying binary variables. The performance of these algorithms is sensitive to the number of time steps (Leclère et al. 2020; Philpott et al. 2016).

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 Kaut et al. (2014) and further formalized in Maggioni et al. (2019). 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 Werner et al. (2013). Several authors have studied stochastic optimization problems with interdependent strategic/operational decisions or intrastage/interstage problems (Abgottspon and Andersson 2016; Abgottspon 2015; Abgottspon and Andersson 2014; Skar et al. 2016; Pritchard et al. 2005), but most of the time the developed methods to tackle the difficulties are problem-dependent. In Kaut et al. (2014) the authors present different particular cases where the two time-scales (called operational and strategic

decision problems) can be easily decomposed. In Maggioni et al. (2019) 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 Haessig et al. (2015)) and in a stochastic setting [which extends Heymann and Martinon (2018)].

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 Carpentier et al. (2023, 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 decomposition methods. The first one, akin to the so-called price decomposition, gives a lower bound of the slow time-scale Bellman functions, whereas the second one, 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. The appendices of the paper are available as supplementary material on the journal web site. In the supplementary material, we 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 $\llbracket r, s \rrbracket$ 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 $\Delta t = 30$ min, $M + 1 = 48$, 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 $m \in \llbracket 0, M \rrbracket$ of every slow time step $d \in \llbracket 0, D \rrbracket$,
2. another type of (say, strategic) decision every slow time step $d \in \llbracket 0, D \rrbracket \cup \{D+1\}$.

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

$$\mathbb{T} = \llbracket 0, D \rrbracket \times \llbracket 0, M \rrbracket \cup \{(D+1, 0)\}, \tag{1}$$

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

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

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 $(z_{d,m}, z_{d,m+1}, \dots, z_{d,m'-1}, z_{d,m'})$,
- the notation $\mathbb{Z}_{d,m:m'}$ refers to the Cartesian product $\prod_{k=m}^{m'} \mathbb{Z}_{d,k}$ of spaces $\{\mathbb{Z}_{d,k}\}_{k \in \llbracket m, m' \rrbracket}$

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 $\mathbf{W} = \{\mathbf{W}_d\}_{d \in \llbracket 0, D \rrbracket}$ at the slow time-scale, as detailed below. For any $d \in \llbracket 0, D \rrbracket$, the random variable \mathbf{W}_d consists of a sequence of random variables $\{\mathbf{W}_{d,m}\}_{m \in \llbracket 0, M \rrbracket}$ 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 $\mathbf{W}_{d,m} : \Omega \rightarrow \mathbb{W}_{d,m}$ takes values in a *Borel space*¹ $\mathbb{W}_{d,m}$ (“uncertainty” space), so that $\mathbf{W}_d : \Omega \rightarrow \mathbb{W}_d$ 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,

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

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

$$\mathcal{F}_{\llbracket 0,D \rrbracket} = \{\mathcal{F}_{d,M}\}_{d \in \llbracket 0,D \rrbracket} = (\mathcal{F}_{0,M}, \dots, \mathcal{F}_{D,M}) . \tag{4b}$$

In the same vein, we introduce a decision process $\mathbf{U} = \{\mathbf{U}_d\}_{d \in \llbracket 0,D \rrbracket}$ at the slow time-scale, where each \mathbf{U}_d consists of a sequence $\{\mathbf{U}_{d,m}\}_{m \in \llbracket 0,M \rrbracket}$ 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 $\mathbf{U}_{d,m} : \Omega \rightarrow \mathbb{U}_{d,m}$ 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 $\mathbf{X} = \{\mathbf{X}_d\}_{d \in \llbracket 0,D+1 \rrbracket}$ at the slow time-scale, where each random variable $\mathbf{X}_d : \Omega \rightarrow \mathbb{X}_d$ 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 $d \in \llbracket 0, D+1 \rrbracket$, 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 $d \in \llbracket 0, D+1 \rrbracket$, \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 $d \in \llbracket 0, D \rrbracket$, we introduce a nonnegative Borel-measurable instantaneous cost function $L_d : \mathbb{X}_d \times \mathbb{U}_d \times \mathbb{W}_d \rightarrow [0, +\infty]$ and a Borel-measurable dynamics $f_d : \mathbb{X}_d \times \mathbb{U}_d \times \mathbb{W}_d \rightarrow \mathbb{X}_{d+1}$. 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 $K : \mathbb{X}_{D+1} \rightarrow [0, +\infty]$.²

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

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

¹ We call Borel space $(\mathbb{X}, \mathcal{B}_{\mathbb{X}})$ a Borel set \mathbb{X} equipped with its Borel σ -field $\mathcal{B}_{\mathbb{X}}$ (Bertsekas and Shreve 1996, 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 .

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

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

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 $\mathbf{U}_{d,m}$ —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 $V^e(x)$ 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:

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

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

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

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

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):

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

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 functions³ 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 $\mathbf{U}_d = (\mathbf{U}_{d,0}, \dots, \mathbf{U}_{d,M})$ 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 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 $a \notin A$.

3 Time block decomposition, Bellman functions and monotonicity assumptions

In Sect. 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 Sect. 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 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 $\{V_d^e\}_{d \in \llbracket 0, D+1 \rrbracket}$ 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 $V_{D+1}^e = K$, and then, for $d \in \llbracket 0, D \rrbracket$ and for all $x \in \mathbb{X}_d$, we set

$$V_d^e(x) = \inf_{\mathbf{x}_{d+1}, \mathbf{U}_d} \mathbb{E} [L_d(x, \mathbf{U}_d, \mathbf{W}_d) + V_{d+1}^e(\mathbf{X}_{d+1})], \tag{9a}$$

$$\text{s.t. } \mathbf{X}_{d+1} = f_d(x, \mathbf{U}_d, \mathbf{W}_d), \tag{9b}$$

$$\sigma(\mathbf{U}_{d,m}) \subset \sigma(\mathbf{W}_{d,0:m}), \quad \forall m \in \llbracket 0, M \rrbracket, \tag{9c}$$

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* $\{V_d^i\}_{d \in \llbracket 0, D+1 \rrbracket}$ associated with Problem (7). At time $D+1$, we set $V_{D+1}^i = K$, and then, for $d \in \llbracket 0, D \rrbracket$ and for all $x \in \mathbb{X}_d$, we set

$$V_d^i(x) = \inf_{\mathbf{x}_{d+1}, \mathbf{U}_d} \mathbb{E} [L_d(x, \mathbf{U}_d, \mathbf{W}_d) + V_{d+1}^i(\mathbf{X}_{d+1})], \tag{10a}$$

$$\text{s.t. } f_d(x, \mathbf{U}_d, \mathbf{W}_d) \geq \mathbf{X}_{d+1}, \tag{10b}$$

$$\sigma(\mathbf{U}_{d,m}) \subset \sigma(\mathbf{W}_{d,0:m}), \quad \forall m \in \llbracket 0, M \rrbracket, \tag{10c}$$

$$\sigma(\mathbf{X}_{d+1}) \subset \sigma(\mathbf{W}_{d,0:M}). \tag{10d}$$

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 $V_{D+1}^e = V_{D+1}^i = K$, 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):

$$V_d^i \leq V_d^e, \quad \forall d \in \llbracket 0, D+1 \rrbracket. \tag{11}$$

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 $\{V_d^e\}_{d \in \llbracket 0, D+1 \rrbracket}$ associated with Problem (6) and the sequence of Bellman functions $\{V_d^i\}_{d \in \llbracket 0, D+1 \rrbracket}$ 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 $\{V_d^e\}_{d \in \llbracket 0, D+1 \rrbracket}$ Bellman functions and, for $d \in \llbracket 0, D+1 \rrbracket$, the inequality $V_d^i \geq V_d^e$ —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:

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

2. For all $d \in \llbracket 0, D \rrbracket$, the effective domain of the Bellman function V_d^e is induced by the effective domain of the instantaneous cost function L_d , namely

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

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

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

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

Let us comment the three items of Assumption 1. The first assumption concerning the final cost function K is rather natural in stock management, as it just expresses the fact that the more in the stock the more value (hence the less it costs).

Moreover, in long term optimization problem, as it is the case in the case study of the paper, the final cost is set to zero so that the assumption is indeed satisfied.

The second assumption concerning the effective domain of the Bellman functions is akin to the relatively complete recourse assumption (Rockafellar and Wets 1976). It is a usual assumption in multistage stochastic optimization, and it is satisfied in our case study.

The third assumption is more involved. However, it is rather natural as it expresses the fact that the higher the state, the higher the next state and the lower the cost. Nevertheless, to achieve higher future state and lower cost, one may have to change control, hence the existence of \tilde{U}_d in Eqs. (12c) and (12d). Once again, this assumption is satisfied in our case study and more generally in energy problems where having more stock at time t is better for the induced cost and for the stock at the next time step.

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

$$\forall d \in \llbracket 0, D+1 \rrbracket, \quad \forall (x, x') \in \text{dom}V_d^c \times \text{dom}V_d^c, \quad x \leq x' \implies V_d^c(x) \geq V_d^c(x').$$

Proof The proof is done by backward induction. At time $D+1$, the Bellman function $V_{D+1}^c = K$ is nonincreasing on its effective domain by Condition 1 of Assumption 1. For $d \in \llbracket 0, D \rrbracket$, assume that V_{d+1}^c is nonincreasing on its effective domain. Let $(x, x') \in \text{dom}V_d^c \times \text{dom}V_d^c$ 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

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

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

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

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 $d \in \llbracket 0, D+1 \rrbracket$, the (original) Bellman function V_d^e in (9) coincides with the (relaxed) Bellman function V_d^i in (10):*

$$V_d^i = V_d^e, \quad \forall d \in \llbracket 0, D+1 \rrbracket .$$

Proof By Eq. (11), we have that $V_d^i \leq V_d^e$ for all $d \in \llbracket 0, D+1 \rrbracket$. 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 $\llbracket 0, D \rrbracket$ and assume that $V_{d+1}^i = V_{d+1}^e$. For any $x \in \text{dom}V_d^i$ 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 Eq. (10b) and

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

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

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

Now, we obtain successively

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

We thus obtain the reverse inequality $V_d^i \geq V_d^e$, 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 $d \in \llbracket 0, D \rrbracket$ 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 $\{V_d^i\}_{d \in \llbracket 0, D+1 \rrbracket}$. 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 $d \in \llbracket 0, D \rrbracket$ and every $x \in \mathbb{X}_d$, and each numerical solving might be in itself hard.

To tackle this issue, we propose in Sects. 4.1 and 4.2 to compute respectively lower and upper bounds of the Bellman functions at the slow time-scale. Computing the lower (resp. upper) bounds of the Bellman functions is done by Algorithm 2. (resp. Algorithm 1) based on so-called price (resp. ressource) decomposition techniques [see Bertsekas (1999, Chap. 6) and Carpentier and Cohen (2017)] applied to Problem (10). These two algorithms are precisely described in Appendix A. These two Bellman functions bounds can then be used to design admissible two-time-scale optimization policies (see Sect. 5).

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 $\{V_d^i\}_{d \in \llbracket 0, D+1 \rrbracket}$ given by Eq. (10). These bounds derive from an algorithm which appears to be connected to the one developed in Heymann and Martinon (2018), called “adaptive weights algorithm”. We extend the results of Heymann and Martinon (2018) 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 $\{V_d^i\}_{d \in \llbracket 0, D+1 \rrbracket}$ 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 $\mathcal{F}_{\llbracket 0, D \rrbracket}$ -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 $d \in \llbracket 0, D \rrbracket$, we define the function $L_d^P : \mathbb{X}_d \times \mathbb{Y}_{d+1} \rightarrow \mathbb{R} \cup \{\pm\infty\}$ by⁴

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

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

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

⁴ Recall that $\mathbb{Y}_{d+1} = \mathbb{X}_{d+1} = \mathbb{R}^{n_{d+1}}$.

Proposition 4 Consider the sequence $\{V_{-d}^P\}_{d \in \llbracket 0, D+1 \rrbracket}$ of Bellman functions which is defined by $V_{-D+1}^P = K$ and for all $d \in \llbracket 0, D \rrbracket$, and for all $x \in \mathbb{X}_d$ by⁵

$$V_{-d}^P(x) = \sup_{p_{d+1} \leq 0} \left(L_d^P(x, p_{d+1}) - (V_{-d+1}^P)^*(p_{d+1}) \right), \tag{17}$$

where $(V_{-d+1}^P)^* : \mathbb{Y}_{d+1} \rightarrow \mathbb{R} \cup \{\pm\infty\}$ is the Fenchel conjugate of V_{-d+1}^P [see Rockafellar (2015)]. Then, the Bellman functions $\{V_{-d}^P\}_{\llbracket 0, D+1 \rrbracket}$ given by Eq. (17) are lower bounds of the corresponding Bellman functions $\{V_d^i\}_{d \in \llbracket 0, D+1 \rrbracket}$ given by Eq. (10), that is,

$$V_{-d}^P \leq V_d^i, \quad \forall d \in \llbracket 0, D+1 \rrbracket. \tag{18}$$

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 $\varphi : \mathbb{X} \rightarrow \mathbb{R} \cup \{\pm\infty\}$. We assume that \mathcal{X} contains all the constant random variables. We prove that

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

- The \leq inequality $\inf_{\mathbf{X} \in \mathcal{X}} \mathbb{E}[\varphi(\mathbf{X})] \leq \inf_{x \in \mathbb{X}} \varphi(x)$ 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 $\inf_{\mathbf{X} \in \mathcal{X}} \mathbb{E}[\varphi(\mathbf{X})] \leq \inf_{x \in \mathbb{X}} \varphi(x)$. Assume now that $\inf_{x \in \mathbb{X}} \varphi(x) = \underline{\varphi} > -\infty$. Then $\varphi(\mathbf{X}) \geq \underline{\varphi}$ \mathbb{P} -a.s. for all $\mathbf{X} \in \mathcal{X}$ and hence $\inf_{\mathbf{X} \in \mathcal{X}} \mathbb{E}[\varphi(\mathbf{X})] \geq \underline{\varphi}$. Consider an arbitrary $\epsilon > 0$ and \mathbf{X}_ϵ such that $\mathbb{E}[\varphi(\mathbf{X}_\epsilon)] \leq \inf_{x \in \mathbb{X}} \mathbb{E}[\varphi(\mathbf{X})] + \epsilon$. We successively obtain $\inf_{x \in \mathbb{X}} \varphi(x) = \mathbb{E}[\inf_{x \in \mathbb{X}} \varphi(x)] \leq \mathbb{E}[\varphi(\mathbf{X}_\epsilon)] \leq \inf_{x \in \mathbb{X}} \mathbb{E}[\varphi(\mathbf{X})] + \epsilon$. Thus, the reverse inequality $\inf_{\mathbf{X} \in \mathcal{X}} \mathbb{E}[\varphi(\mathbf{X})] \geq \inf_{x \in \mathbb{X}} \varphi(x)$ follows, hence the equality in (19).

We turn now to the proof of (18), that we do by backward induction. First, we have that $V_{-D+1}^P = K = V_{D+1}^i$. Second, consider $d \in \llbracket 0, D \rrbracket$ and assume that $V_{-d+1}^P \leq V_{d+1}^i$.

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 \dagger Moreau (1970), which extends the usual addition to extended reals by $(+\infty) + (-\infty) = (-\infty)$ and $(-\infty) + (+\infty) = -\infty$. So Eq. (17) has to be understood as $V_{-d}^P(x) = \sup_{p_{d+1} \leq 0} \left(L_d^P(x, p_{d+1}) \dagger (- (V_{-d+1}^P)^*(p_{d+1})) \right)$.

Explicitly using the Moreau lower addition \dagger (see Footnote 5), we successively have⁶

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

by substituting (16), and by using subadditivity of the infimum operation with respect to the Moreau lower addition \dagger

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

This ends the proof. □

Remark 5 In the presentation above, we could have defined the sequence $\{ \underline{V}_d^P \}_{d \in \llbracket 0, D+1 \rrbracket}$ as

$$\underline{V}_d^P(x) = \sup_{p_{d+1} \in \mathbb{V}_{d+1}} \left(L_d^P(x, p_{d+1}) - (V_{d+1}^P)^*(p_{d+1}) \right) ,$$

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

$$\underline{V}_d^P \leq V_d^c, \quad \forall d \in \llbracket 0, D+1 \rrbracket ,$$

the Bellman functions V_d^c replacing the V_d^i . 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_d^i is mandatory.

⁶ Here below, we use \dagger to stress that there might be an addition of two conflicting $\pm\infty$. When we leave the notation \dagger , 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 $\{V_d^i\}_{d \in \llbracket 0, D+1 \rrbracket}$ 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 $L_d^R : \mathbb{X}_d \times \mathbb{X}_{d+1} \rightarrow [0, +\infty]$ by

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

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

$$\sigma(\mathbf{U}_{d,m}) \subset \sigma(\mathbf{W}_{d,0:m}) , \quad \forall m \in \llbracket 0, M \rrbracket , \tag{20c}$$

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 than ensuring the equality between a random variable and a deterministic quantity.

Proposition 6 Consider the sequence $\{\bar{V}_d^R\}_{d \in \llbracket 0, D+1 \rrbracket}$ of Bellman functions defined inductively by $\bar{V}_{D+1}^R = K$ and for all $d \in \llbracket 0, D \rrbracket$ and for all $x \in \mathbb{X}_d$ by

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

Then, the Bellman functions $\{\bar{V}_d^R\}_{d \in \llbracket 0, D+1 \rrbracket}$ given by Eq. (21) are upper bounds of the Bellman functions $\{V_d^i\}_{d \in \llbracket 0, D+1 \rrbracket}$ given by Eq. (10), that is,

$$V_d^i \leq \bar{V}_d^R , \quad \forall d \in \llbracket 0, D+1 \rrbracket . \tag{22}$$

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

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

This ends the proof. □

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

In Sect. 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 Sect. 5.2, we discuss optimality.

5.1 Computation of policies

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

Then, for a given slow time step $d \in \llbracket 0, D \rrbracket$ and a given current state $x_d \in \mathbb{X}_d$, we can use \tilde{V}_{d+1} as an approximation of the Bellman function V_{d+1}^c 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 Bertsekas and Shreve (1996)]:

$$\mathbf{U}_d^* \in \arg \min_{\mathbf{U}_d} \mathbb{E} [L_d(x_d, \mathbf{U}_d, \mathbf{W}_d) + \tilde{V}_{d+1}(f_d(x_d, \mathbf{U}_d, \mathbf{W}_d))] , \quad (23a)$$

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

This fast-time-scale optimization problem can be solved by any method that provides an online policy as presented in Bertsekas (2005). The presence of a final cost \tilde{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 less time.

Obtaining a policy using the Bellman function lower bounds (Algorithm 2)

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 $d \in \llbracket 0, D \rrbracket$. This process produces, for each $d \in \llbracket 0, D \rrbracket$ and for each $(x_d, p_{d+1}) \in \mathbb{X}_d \times \mathbb{Y}_{d+1}$, an optimal policy $\pi_d^P[x_d, p_{d+1}] : \mathbb{W}_d \rightarrow \mathbb{U}_d$ and an optimal value $L_d^P(x_d, p_{d+1})$.

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

$$p_{d+1} \in \arg \max_{p \leq 0} \left(L_d^P(x_d, p) - (\underline{V}_{d+1}^P)^*(p) \right), \tag{24}$$

where the sequence of Bellman functions lower bounds sequence $\{\underline{V}_d^P\}_{d \in \llbracket 0, D+1 \rrbracket}$ is obtained by solving Eq. (17) using Algorithm 2.

Thanks to this deterministic price p_{d+1} , we apply the corresponding policy $\pi_d^P[x_d, p_{d+1}]$ 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 $x_{d+1} = f_d(x_d, \pi_d^P[x_d, p_{d+1}](w_d), w_d)$.

Obtaining a policy using the Bellman function upper bounds (Algorithm 1)

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 $d \in \llbracket 0, D \rrbracket$. This leads, for each $d \in \llbracket 0, D \rrbracket$ and for each $(x_d, x_{d+1}) \in \mathbb{X}_d \times \mathbb{X}_{d+1}$, to an optimal policy $\pi_d^R[x_d, x_{d+1}] : \mathbb{W}_d \rightarrow \mathbb{U}_d$ and to an optimal value $L_d^R(x_d, x_{d+1})$.

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

$$x_{d+1} \in \arg \min_{x \in \mathbb{X}_{d+1}} \left(L_d^R(x_d, x) + \overline{V}_{d+1}^R(x) \right), \tag{25}$$

where the sequence of Bellman functions lower bounds sequence $\{\overline{V}_d^R\}_{d \in \llbracket 0, D+1 \rrbracket}$ is obtained by solving Eq. (21) using Algorithm 1. We apply the corresponding policy $\pi_d^R[x_d, x_{d+1}]$ 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 $x_{d+1} = f_d(x_d, \pi_d^R[x_d, x_{d+1}](w_d), w_d)$.

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^P and the resource Bellman functions \overline{V}_d^R are respectively lower and upper bounds for the Bellman functions associated with (relaxed) Problem (7):

$$\underline{V}_d^P \leq V_d^i \leq \overline{V}_d^R, \quad \forall d \in \llbracket 0, D+1 \rrbracket. \tag{26}$$

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):

$$\underline{V}_d^P \leq V_d^e \leq \overline{V}_d^R, \quad \forall d \in \llbracket 0, D+1 \rrbracket. \tag{27}$$

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

Assumption 2 (White noise assumption) The sequence of random vectors $\{\mathbf{W}_d\}_{d \in \llbracket 0, D \rrbracket}$ is white, that is, $\{(\mathbf{W}_{d,0}, \dots, \mathbf{W}_{d,m}, \dots, \mathbf{W}_{d,M})\}_{d \in \llbracket 0, D \rrbracket}$ 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 $\mathbf{W}_d = (\mathbf{W}_{d,0}, \dots, \mathbf{W}_{d,M})$ 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 Eqs. (9) (resp. (10)). More explicitly, we have that

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

Proof The fact that the function V^e is equal to the function V_0^e is a consequence of Carpentier et al. (2023, 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 $\Delta = \{\Delta_{d+1}\}_{d \in \llbracket 0, D \rrbracket}$, each control variable Δ_{d+1} taking values in \mathbb{X}_{d+1} :

$$V^i(x) = \inf_{\mathbf{X}, \mathbf{U}, \mathbf{\Delta}} \mathbb{E} \left[\sum_{d=0}^D L_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d) + K(\mathbf{X}_{D+1}) \right], \tag{28a}$$

$$\text{s.t } \mathbf{X}_0 = x, \mathbf{X}_{d+1} = f_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d) - \mathbf{\Delta}_{d+1}, \forall d \in \llbracket 0, D \rrbracket, \tag{28b}$$

$$\mathbf{\Delta}_{d+1} \geq 0, \forall d \in \llbracket 0, D \rrbracket, \tag{28c}$$

$$\sigma(\mathbf{U}_{d,m}) \subset \mathcal{F}_{d,m}, \forall (d, m) \in \llbracket 0, D \rrbracket \times \llbracket 0, M \rrbracket, \tag{28d}$$

$$\sigma(\mathbf{\Delta}_{d+1}) \subset \mathcal{F}_{d,M}, \forall d \in \llbracket 0, D \rrbracket. \tag{28e}$$

Second, Problem (28) involves standard equality constraints in the dynamics, so that the machinery developed in Carpentier et al. (2023, 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 Eq. (10) when replacing the extra nonnegative decision variables by inequality constraints. \square

As an immediate consequence of Propositions 3 and 8, 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,*

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

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^P and the resource Bellman function \overline{V}_0^R at time $d = 0$ are respectively lower and upper bounds for the optimal value function of Problem (6):

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

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^P and the resource Bellman functions \overline{V}_d^R always allow to compute admissible policies for Problem (6), as explained in Sect. 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 Sect. 6.1, we formulate the problem. In Sect. 6.2, we simplify the intraday problems. In Sect. 6.3, we describe the data used for the numerical experiments. Finally, in Sect. 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 min. 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 Sect. 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 time⁷ $(d, m) \in \llbracket 0, D \rrbracket \times \llbracket 0, M+1 \rrbracket$, 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

⁷ There is here a slight difference with the notations presented in Sect. 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 Eq. (31) and comments above).

$$\mathbf{S}_{d,m+1} = \mathbf{S}_{d,m} + \rho^c \mathbf{U}_{d,m}^+ - \rho^d \mathbf{U}_{d,m}^-, \quad \forall m \in \llbracket 0, M \rrbracket, \tag{30a}$$

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

$$\mathbf{H}_{d,m+1} = \mathbf{H}_{d,m} - \mathbf{U}_{d,m}^+ - \mathbf{U}_{d,m}^-, \quad \forall m \in \llbracket 0, M \rrbracket, \tag{30b}$$

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)

$$\mathbf{C}_{d,m+1} = \mathbf{C}_{d,m}, \quad \forall m \in \llbracket 0, M \rrbracket. \tag{30c}$$

These equations at the fast time-scale are gathered as

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

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 $\mathfrak{N} : \mathbb{R}_+ \rightarrow \mathbb{N}$ 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:

$$(\mathbf{S}_{d+1,0}, \mathbf{H}_{d+1,0}, \mathbf{C}_{d+1,0}) = \psi(\mathbf{S}_{d,M+1}, \mathbf{H}_{d,M+1}, \mathbf{C}_{d,M+1}, \mathbf{B}_d), \quad \forall d \in \llbracket 0, D \rrbracket. \tag{31d}$$

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

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

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

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

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 $\pi_{d,m}^e$, assumed to be deterministic and known, whereas excess energy production is assumed to be wasted. The price \mathbf{P}_d^b of a new battery is supposed to be random, so that the operating cost L_d during the slow time step d is

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

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

$$\inf_{\{\mathbf{U}_{d,0:M}, \mathbf{B}_d\}_{d \in \llbracket 0, D \rrbracket}} \mathbb{E} \left[\sum_{d=0}^D \gamma^d \left(\sum_{m=0}^M \pi_{d,m}^e (\mathbf{D}_{d,m} + \mathbf{U}_{d,m}^+ - \mathbf{U}_{d,m}^-)^+ + \mathbf{P}_d^b \mathbf{B}_d \right) + K(\mathbf{S}_{D+1,0}, \mathbf{H}_{D+1,0}, \mathbf{C}_{D+1,0}) \right], \tag{34a}$$

subject, for all $(d, m) \in \llbracket 0, D \rrbracket \times \llbracket 0, M \rrbracket$, to state dynamics

$$(\mathbf{S}_{0,0}, \mathbf{H}_{0,0}, \mathbf{C}_{0,0}) = (s_0, h_0, c_0), \tag{34b}$$

$$(\mathbf{S}_{d,m+1}, \mathbf{H}_{d,m+1}, \mathbf{C}_{d,m+1}) = \varphi(\mathbf{S}_{d,m}, \mathbf{H}_{d,m}, \mathbf{C}_{d,m}, \mathbf{U}_{d,m}), \tag{34c}$$

$$(\mathbf{S}_{d+1,0}, \mathbf{H}_{d+1,0}, \mathbf{C}_{d+1,0}) = \psi(\mathbf{S}_{d,M+1}, \mathbf{H}_{d,M+1}, \mathbf{C}_{d,M+1}, \mathbf{B}_d), \tag{34d}$$

to bounds constraints

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

$$\mathbf{U}_{d,m} \in [\underline{\mathbf{U}}, \overline{\mathbf{U}}], \quad \mathbf{B}_d \in [0, \overline{\mathbf{B}}], \tag{34f}$$

and tononanticipativity constraints

$$\sigma(\mathbf{U}_{d,m}) \subset \sigma(\mathbf{D}_{0,0}, \dots, \mathbf{D}_{d,m}, \mathbf{P}_0^b, \dots, \mathbf{P}_{d-1}^b), \tag{34g}$$

$$\sigma(\mathbf{B}_d) \subset \sigma(\mathbf{d}_{0,0}, \dots, \mathbf{d}_{d,M}, \mathbf{P}_0^b, \dots, \mathbf{P}_d^b). \tag{34h}$$

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

$$\mathbf{U}_d = (\{\mathbf{U}_{d,m}\}_{m \in \llbracket 0, M \rrbracket}, \mathbf{B}_d). \tag{35a}$$

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

$$\mathbf{W}_d = (\{\mathbf{D}_{d,m}\}_{m \in \llbracket 0, M \rrbracket}, \mathbf{P}_d^b), \tag{35b}$$

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 Sect. 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):

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

- 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 Eq. (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 $m \in \llbracket 0, M \rrbracket$. 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}_d^b) 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 Sect. 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_d^P in (16) and L_d^R in (20), that we call *intraday functions* in this case study. As explained in Sect. 4.1, (resp. Sect. 4.2),

the intraday functions L_d^P (resp. L_d^R) depend on the couple (x_d, p_{d+1}) (resp. (x_d, r_{d+1})), namely the 6-uple $(s_d, h_d, c_d, p_{d+1}^s, p_{d+1}^h, p_{d+1}^c)$ (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 Sect. 4.2, the aim of the resource decomposition algorithm is to compute, for all slow time steps $d \in \llbracket 0, D+1 \rrbracket$, upper bounds \bar{V}_d^R 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 $d \in \llbracket 0, D \rrbracket$, 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 Eqs. (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 $\mathbf{P}_d^b \mathbf{B}_d$, 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 Eq. (30d) for $m = M$, instead of Eq. (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

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

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

with, for all $m \in \llbracket 0, M \rrbracket$,

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

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

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

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.

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

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

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

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

$$\begin{aligned} \overline{V}_d^R(s_d, h_d, c_d) = & \inf_{s_{d+1}, h_{d+1}, \mathbf{B}_d} \mathbb{E} \left[\gamma(L_d^R(s_d, h_d, c_d, s_{d+1}, h_{d+1}) + \mathbf{P}_d^b \mathbf{B}_d) \right. \\ & \left. + \overline{V}_{d+1}^R(s_{d+1,0}, \mathbf{H}_{d+1,0}, \mathbf{C}_{d+1,0}) \right] , \end{aligned} \tag{38a}$$

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

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

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

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

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_d^R by the function \tilde{L}_d^R with

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

where δ_A denotes the indicator function of the set A (see Footnote 3 on page 6). 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_d^R and the two other terms can be added as it is easily seen that $L_d^R(s_d, h_d, c_d, s_{d+1}, h_{d+1}) = +\infty$ 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

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

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 $(h_d, h_{d+1}) \in [0, \mathfrak{N}(c_d) c_d]^2$ 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 Eq. (39).

Resource intraday function approximation

As suggested in Kaut et al. (2014), 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 Eq. (39), we obtain a new function \widehat{L}_d^R approximating the original function L_d^R , that is

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

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

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

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

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

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

where the new dynamics $\psi^{\text{H.C}}$ 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, \dots, 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 Sect. 4.1, the aim of the price decomposition algorithm is to compute, for all slow time steps $d \in \llbracket 0, D+1 \rrbracket$, lower bounds \underline{V}_d^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 $d \in \llbracket 0, D \rrbracket$, 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 Eq. (31d), but on Eq. (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

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

Following the framework of Sect. 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 $m \in \llbracket 0, M \rrbracket$ 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_d^P does not depend anymore on the health h_d , and is defined as

$$L_d^P(s_d, c_d, p_{d+1}^s, p_{d+1}^h) = \inf_{\mathbf{U}_{d,0:M}} \mathbb{E} \left[\sum_{m=0}^M \left(\pi_{d,m}^e \max(0, \mathbf{D}_{d,m} + \mathbf{U}_{d,m}^+ - \mathbf{U}_{d,m}^-) - p_{d+1}^h (\mathbf{U}_{d,m}^+ + \mathbf{U}_{d,m}^-) \right) + p_{d+1}^s \mathbf{S}_{d,M+1} \right], \tag{43a}$$

subject to, for all $m \in \llbracket 0, M \rrbracket$,

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

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

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

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

$$\begin{aligned} \underline{V}_d^P(s_d, h_d, c_d) = & \sup_{(p_{d+1}^s, p_{d+1}^h) \leq 0} \left(L_d^P(s_d, c_d, p_{d+1}^s, p_{d+1}^h) + \inf_{s_{d,M+1}, h_{d,M+1}} \inf_{\mathbf{B}_d} \mathbb{E} \left[\gamma \mathbf{P}_d^b \mathbf{B}_d \right. \right. \\ & \left. \left. - p_{d+1}^s s_{d,M+1} + (p_{d+1}^h (h_d - h_{d,M+1}) + \underline{V}_{d+1}^P(\mathbf{S}_{d+1,0}, \mathbf{H}_{d+1,0}, \mathbf{C}_{d+1,0})) \right] \right), \end{aligned} \tag{44a}$$

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

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

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

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 \hat{L}_d^P approximating the original function L_d^P , that is,

$$\hat{L}_d^P(c_d, p_{d+1}^h) = L_d^P(0, c_d, 0, p_{d+1}^h) \approx L_d^P(s_d, c_d, p_{d+1}^s, p_{d+1}^h). \tag{45}$$

The approximated price intraday function \hat{L}_d^P only depends on the 2-tuple (c_d, p_{d+1}^h) , which significantly reduces the time needed to compute it. Then, the sequence $\{\underline{V}_d^P\}_{d \in \llbracket 0, D+1 \rrbracket}$ of Bellman functions in (44) is approximated by the sequence $\{\hat{V}_d^P\}_{d \in \llbracket 0, D+1 \rrbracket}$ given by the following recursion:

$$\begin{aligned} \hat{V}_d^P(h_d, c_d) = \sup_{p_{d+1}^h \leq 0} & \left(\hat{L}_d^P(c_d, p_{d+1}^h) + \inf_{h_{d,M+1}} \inf_{\mathbf{B}_d} \mathbb{E} \left[\gamma \mathbf{P}_d^b \mathbf{B}_d \right. \right. \\ & \left. \left. + p_{d+1}^h (h_d - h_{d,M+1}) + \hat{V}_{d+1}^P(\psi^{\text{H,C}}(h_{d,M+1}, c_d, \mathbf{B}_d)) \right] \right), \end{aligned} \tag{46a}$$

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

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

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

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, \dots, 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.

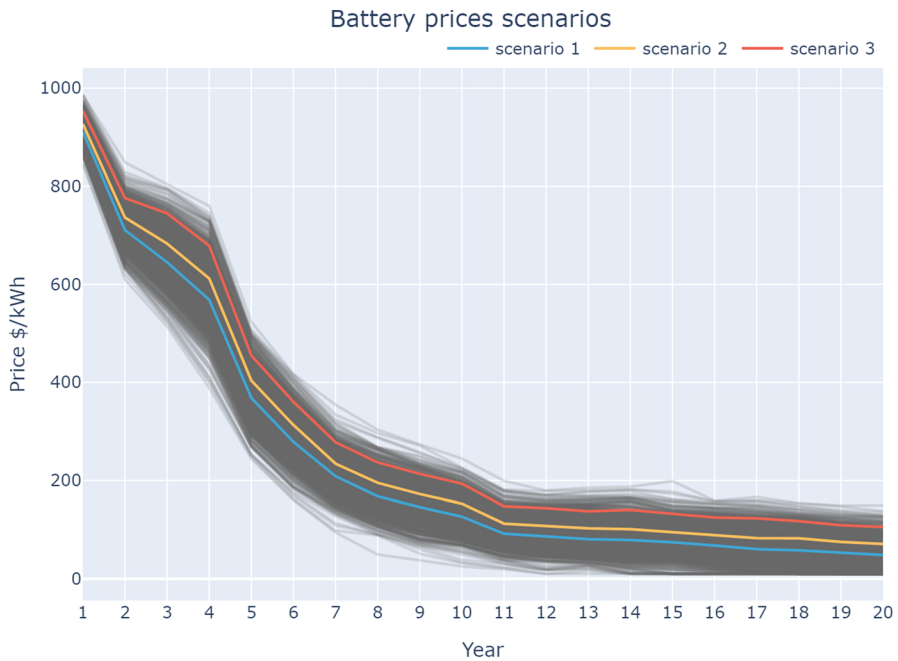


Fig. 1 Scenarios of battery prices over a twenty-year timespan

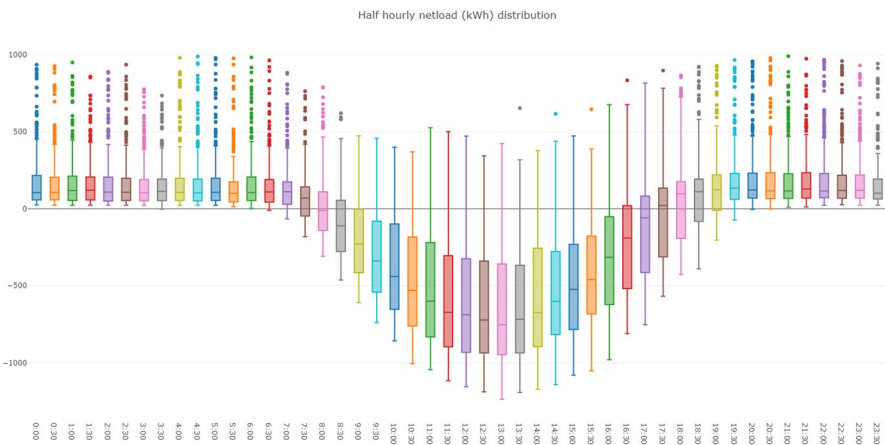


Fig. 2 Daily half hourly distribution of net demand (kWh)

- Horizon: 20 years.
- Fast time step: 30 min.
- Slow time step: 1 day.
- Number of time steps: 350,400 (= $(24 \times 2) \times (20 \times 365)$).
- Battery renewal capacity: between 0 and 1500 kWh with an 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 Fig. 1 the scenarios we generated.

Three scenarios are highlighted in Fig. 1; they correspond to the three scenarios we comment in the numerical results in Sect. 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.

Data to model demand and production

In order to have a realistic dataset in the model described in Sect. 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 Fig. 2 the half hourly distribution of the net demand (demand minus solar production) during one day.

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 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}_d^b 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 Sect. 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 \hat{L}_d^R in (40) and the associated resource Bellman functions \hat{V}_d^R in (41) (resp. price intraday functions \hat{L}_d^P in (45) and price Bellman functions \hat{V}_d^P in (46)) after all simplifications presented in Sect. 6.2.1 (resp. Sect. 6.2.2), and second evaluating the policies induced by the Bellman functions \hat{V}_d^R and \hat{V}_d^P .

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

¹⁰ <https://zenodo.org/record/5510400>.

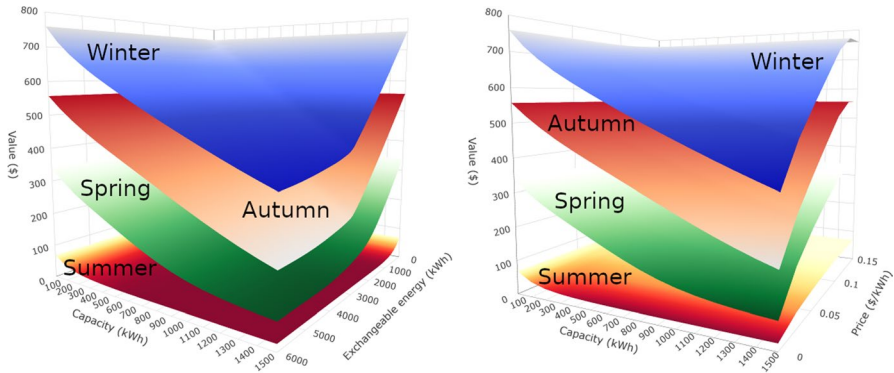


Fig. 3 Resource (left) and price (right) intraday functions for each trimester

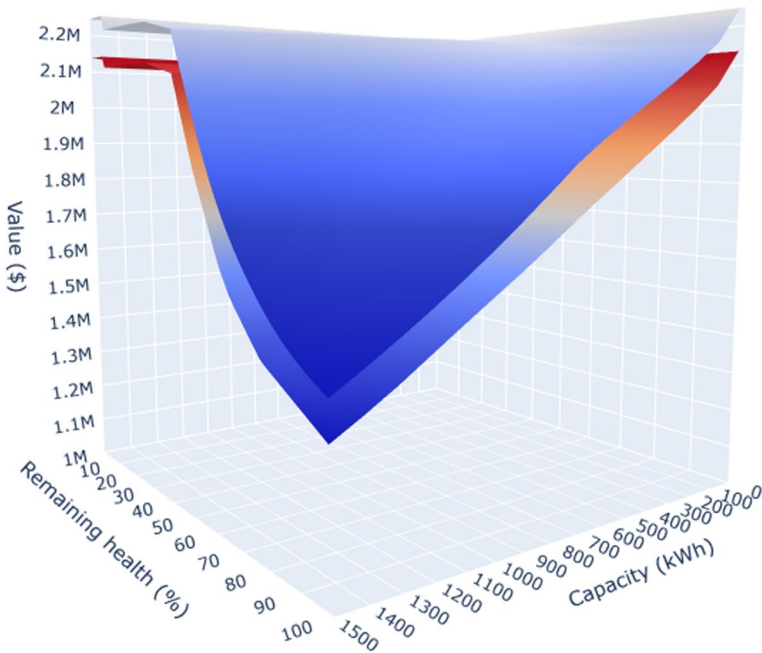


Fig. 4 Resource and price Bellman functions at day 1

6.4.1 Computation of the resource and price intraday functions

To compute the approximated resource intraday functions \hat{L}_d^R as given in Eq. (40) and the approximated price intraday functions \hat{L}_d^P as given in Eq. (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

Table 1 Computing times of the price and resource decomposition methods

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

Table 2 Bounds obtained by resource and price decomposition

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

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_{d+1}^h) 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 quantization, using k -means Kaut (2021), 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 \hat{L}_d^R (resp. \hat{L}_d^P) for every possible capacity c_d and every possible exchangeable energy $h_d - h_{d+1}$ (resp. every possible price p_{d+1}^h). In this study, the possible values of the capacity c_d are $\{0, 100 \dots, 1500\}$ kWh, whereas the possible values of the price p_{d+1}^h are $\{0, 0.025, 0.05, \dots, 0.2\}$.

We display in Fig. 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.

6.4.2 Computation of the resource and price Bellman functions

Once obtained all possible intraday functions \hat{L}_d^R and \hat{L}_d^P , the Bellman functions \hat{V}_d^R and \hat{V}_d^P are respectively computed by the Bellman recursions (41) and (46), for $d \in \llbracket 0, D \rrbracket$. We display in Fig. 4 the resource and price Bellman functions obtained for the first day of the time horizon.

Table 3 Computing times of simulation

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

Table 4 Average simulation costs using original scenarios

	Price	Resource
Average cost over 1000 true scenarios	2.83 M\$	2.86 M\$

We observe that the resource and price Bellman functions present approximately 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 price and resource decomposition algorithms, namely the total CPU times and the total wall times¹¹ when parallelization is on.

The computation is run on an Intel i7-8850H CPU @ 2.60 GHz 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 $\hat{V}_0^R(x_0)$ and $\hat{V}_0^P(x_0)$ 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 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 Sects. 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.

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

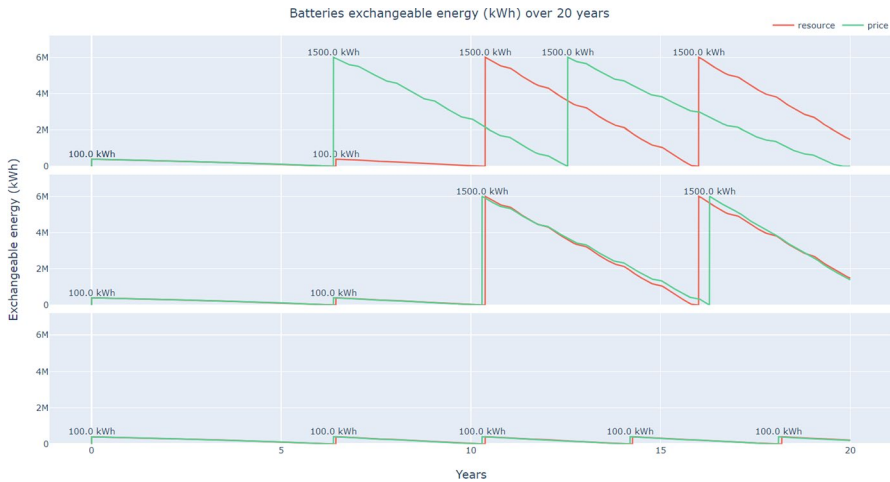


Fig. 5 Three simulations of the evolution of the battery health over 20 years

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.

Simulation using scenarios

We draw 1000 “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 Sect. 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.

Table 5 Simulation results along three scenarios

	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\$

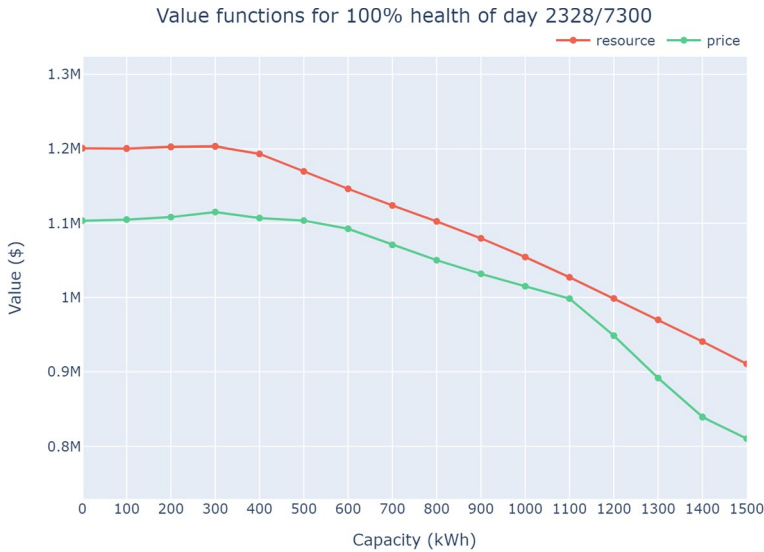


Fig. 6 Bellman function of first renewal day of Scenario 1 for health fixed at 100 %

We also note that the average costs are 20–25% 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 Sect. 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 Fig. 1) among the 1000 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 energy¹²) 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 (1500 kWh, that is, 6 MWh of exchangeable energy) is purchased in price decomposition. Then, over 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 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 Fig. 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 \bar{H} associated with a large battery renewal (1500 kWh), that is, $\hat{V}_{2328}^R(\bar{H}, \cdot)$ in (41) and $\hat{V}_{2328}^P(\bar{H}, \cdot)$ in (46). We observe that the resource Bellman function is significantly higher than the price Bellman function.

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 min (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.

¹² The integer function \mathfrak{R} in Eq. (31b) is such that $\mathfrak{R}(100) = \mathfrak{R}(1500) = 4$.

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 Carpentier et al. (2020)).

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.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s10287-024-00510-5>.

References

- Abgottspon H (2015) Hydro power planning: multi-horizon modeling and its applications. PhD thesis, ETH Zürich. <https://doi.org/10.3929/ethz-a-010548986>
- Abgottspon H, Andersson G (2014) Medium-term optimization of pumped hydro storage with stochastic intrastage subproblems. In: 2014 Power systems computation conference, pp 1–7
- Abgottspon H, Andersson G (2016) Multi-horizon modeling in hydro power planning. *Energy Procedia* 87:2–10
- Bellman R (1957) *Dynamic programming*. Princeton University Press, Princeton
- Bertsekas DP (1999) *Nonlinear programming*. Athena Scientific, Belmont
- Bertsekas DP (2005) Dynamic programming and suboptimal control: a survey from ADP to MPC. *Eur J Control* 11(4–5):310–334
- Bertsekas DP (2017) *Dynamic programming and optimal control*, vol I, 4th edn. Athena Scientific, Belmont, p 555
- Bertsekas DP, Shreve SE (1996) *Stochastic optimal control: the discrete-time case*. Athena Scientific, Belmont
- Carpentier P, Cohen G (2017) *Decomposition-coordination en Optimisation Deterministe et Stochastique*. Springer, Berlin
- Carpentier P, Chancelier J-P, De Lara M, Pacaud F (2020) Mixed spatial and temporal decompositions for large-scale multistage stochastic optimization problems. *J Optim Theory Appl* 186(3):985–1005
- Carpentier P, Chancelier J-P, De Lara M, Martin T, Rigaut T (2023) Time block decomposition of multistage stochastic optimization problems. *J Convex Anal* 30(2)
- Haessig P, Ahmed HB, Multon B (2015) Energy storage control with aging limitation. In: *PowerTech, 2015 IEEE Eindhoven*. IEEE, pp 1–6
- Heymann B, Martinon P (2018) Optimal battery aging: an adaptive weights dynamic programming algorithm. *J Optim Theory Appl* 179(3):1043–1053. <https://doi.org/10.1007/s10957-018-1371-9>
- Kaut M (2021) Scenario generation by selection from historical data. *Comput Manag Sci* 18(3):411–429
- Kaut M, Midhun KT, Werner AS, Tomasgard A, Hellemo L, Fodstad M (2014) Multi-horizon stochastic programming. *Comput Manag Sci* 11(1–2):179–193. <https://doi.org/10.1007/s10287-013-0182-6>. (**Special Issue: Computational Techniques in Management Science**)
- Leclère V, Carpentier P, Chancelier J-P, Lenoir A, Pacaud F (2020) Exact converging bounds for stochastic dual dynamic programming via Fenchel duality. *SIAM J Optim* 30(2):1223–1250
- Maggioli F, Allevi E, Tomasgard A (2019) Bounds in multi-horizon stochastic programs. *Ann Oper Res* 292:1–21

- Moreau JJ (1970) Inf-convolution, sous-additivité, convexité des fonctions numériques. *J Math Pures Appl* 9(49):109–154
- Pereira MV, Pinto LM (1991) Multi-stage stochastic optimization applied to energy planning. *Math Program* 52(1–3):359–375. <https://doi.org/10.1007/BF01582895>
- Philpott A, Wahid F, Bonnans F (2016). MIDAS: a mixed integer dynamic approximation scheme. Research report, Inria Saclay Ile de France. <https://hal.inria.fr/hal-01401950>
- Powell WB (2007) Approximate dynamic programming: solving the curses of dimensionality, vol 703. Wiley-Blackwell, Hoboken
- Pritchard G, Philpott AB, Neame PJ (2005) Hydroelectric reservoir optimization in a pool market. *Math. Program.* 103(3):445–461
- Rockafellar RT (2015) Convex analysis. Princeton University Press, Princeton
- Rockafellar RT, Wets RJ (1976) Stochastic convex programming: relatively complete recourse and induced feasibility. *SIAM J Control Optim* 14(3):574–589
- Skar C, Doorman G, Pérez-Valdés GA, Tomasgard A (2016) A multi-horizon stochastic programming model for the European power system. *Int Peer Rev J* (In review)
- Werner AS, Pichler A, Midthun KT, Hellemo L, Tomasgard A (2013) Risk measures in multi-horizon scenario trees. Springer, Boston, pp 177–201. https://doi.org/10.1007/978-1-4614-9035-7_8
- Zou J, Ahmed S, Sun XA (2018) Stochastic dual dynamic integer programming. *Math Program.* <https://doi.org/10.1007/s10107-018-1249-5>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.