# Algorithms for two-time scales stochastic optimization with applications to long term management of energy storage

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

HAL Id: hal-02013969

https://hal.science/hal-02013969v1

Preprint submitted on 11 Feb 2019 (v1), last revised 29 Aug 2023 (v2)

# Algorithms for two-time scales stochastic optimization with applications to long term management of energy storage

Pierre Carpentier[1], Jean-Philippe Chancelier[2], Michel De Lara[2],
Tristan Rigaut[2,3]

[1] UMA, ENSTA ParisTech    [2] Université Paris Est, CERMICS (ENPC)    [3] Efficacity
February 11, 2019

## Abstract

We design algorithms for two time scales stochastic optimization problems arising from long term storage management. Energy storage devices are of major importance to integrate more renewable energies and demand-side management in a new energy mix. However batteries remain costly even if recent market developments in the field of electrical vehicles and stationary storage tend to decrease their cost. We present a stochastic optimization model aiming at minimizing the investment and maintenance costs of batteries for a house with solar panels. For any given capacity of battery it is necessary to compute a charge/discharge strategy as well as maintenance to maximize revenues provided by intraday energy arbitrage while ensuring a long term aging of the storage devices. Long term aging is a slow process while charge/discharge control of a storage handles fast dynamics. For this purpose, we have designed algorithms that take into account this two time scales aspect in the decision making process. These algorithms are applied to three numerical experiments. First, one of them is used to control charge/discharge, aging and renewal of batteries for a house. Results show that it is economically significant to control aging. Second, we apply and compare our algorithms one a simple charge/discharge and aging problem, that is a multistage stochastic optimization problem with many time steps. We compare our algorithms to SDP and Stochastic Dual Dynamic Programming and we observe that they are less computationally costly while displaying similar performances on the control of a storage. Finally we show that how one of our algorithm can be

1

used for the optimal sizing of a storage taking into account charge/discharge strategy as well as aging.

# Contents

# 1 Introduction

We introduce hereby the reasons to study long term management of energy storage problems and why we use a two time scales stochastic optimization framework to tackle them. Then we present existing literature on these issues.

## 1.1   Context

The integration of renewable energies is of upmost importance to ensure a clean energy production mix that can face the perpetually rising electrical demand. These energies and demand are inherently uncertain as they respectively depend on our environment and on consumers behavior. Electrical storage is used as a buffer to mitigate uncertainties in new electricity grids. Every battery requires a proper management strategy able to make charge/discharge decisions in an uncertain setting to minimize an economical, environmental or energy criterion. These systems are costly, have a fast dynamical behavior (noticeably changing every minute) but can last multiple years. The revenues they provide and their lifetime are deeply related as a battery that is never used lasts many years but does not allow to save energy while a intensively used battery will last a much smaller amount of years but save more energy every day.

In this paper we present a two time scales stochastic optimal control formalism to control systems with fast dynamics that affect long term behavior, as it is the case for batteries. Using well known results from discrete time stochastic optimal control and convex analysis theory, we develop two general methods to decompose that kind of problems by time blocks. Based on these theoretical methods, we develop associated numerical algorithms. We apply these algorithms to a battery charge/discharge and renewal management problem, namely two-time scales stochastic dynamic programming and its dual variant. We also combine one of our method with Stochastic Dual Dynamic Programming and Linear Programming to solve an aging aware sizing-control problem of a battery.

## 1.2   Literature review

The management of micro-grids involves different time scales as the dynamic of currents is faster than the dynamics of voltage/power which is faster than the dynamics of energy flows. For this reason, micro-grids control architecture is often divided into hierarchical levels exchanging information at different paces [19]. In this paper we focus on the energy management level (time step 1 minute) and the long term aging level (time step 1 day). We survey literature on energy storage operation and long term aging management using optimization methods.

### 1.2.1   Stochastic optimization for energy management problems

Stochastic dynamic optimization methods based on the Bellman equation [1] have often been applied to energy storage management. In [24, 13] or [11] the authors

apply Stochasic Dynamic Programming with discretized state and control spaces to solve an energy management problem. This method suffers the so called *curses of dimensionality* as introduced in [1, 3, 23] or [6]. Moreover it is demonstrated in [2] that the convergence of the discretization procedure is of course dependent on the number of time stages.

A major contribution to handle a large number of energy storage for an electricity system is the well known Stochastic Dual Dynamic Programming (SDDP) algorithm [21]. This method is adapted to problems with linear dynamics and convex costs. It has been applied to energy management with battery in [20]. Other similar methods have been developed such that Mixed Integer Dynamic Approximation Scheme (MIDAS) [22] or Stochastic Dual Dynamic Integer Programming [30] for non convex problems, in particular those displaying binary variables. These algorithms performance is sensitive to the number of time steps as stated in [18] and [22].

Other classical Stochastic Programming methods are sensible to the number of time stages. Solving a multistage stochastic optimization problem on a scenario tree displays a complexity exponential in the number of time steps [29].

We present algorithms to decompose, in time, problems displaying many time stages. The algorithms are based on a time block application of the Bellman equation [7]. The motivation is a problem displaying two decisions time scales, but time decomposition also helps to enhance classical methods and algorithms whose performance is sensitive to the number of time steps.

### 1.2.2   Energy storage aging management

Batteries are expensive equipment whose long term management strategy significantly impacts their economic profitability. The authors of [10] use an energy counting model to model and manage the aging of the battery. It corresponds to measure the equivalent number of full cycles $N_{cycles}$ that a battery makes when it charges and discharges a given amount of energy, for instance when a 10 kWh battery charges or discharges 3 kWh, it performs a number of $\frac{3}{2 \times 10}$ cycles as a full cycle exchanges the amount of energy of two times the capacity. The health of the battery is managed in a stochastic infinite horizon setting using Average-Cost Value Iteration [3, 12] that requires a stationary assumption. A more detailed model for NaS batteries is developped in [11] that takes into account depth of discharge (DoD) and temperature in addition to $N_{cycles}$. This model is too detailed to be embedded in an stochastic optimization energy management system. In [15] the authors first propose an abstract model of battery aging to develop continuous time deterministic optimal control methods. An overview of heuristic methods to handle storage aging in a

control framework is provided in [14]. In [17] the authors compare different battery aging models in a stochastic optimal control framework.

In this paper we use the simplest aging model provided in [10]. The main goal is to design algorithms for discrete time finite horizon optimization problems with many time steps and two decision time scales, hence without a stationary assumption (contrary to [10]) and in a stochastic setting (contrary to [14]).

# 2 Stochastic optimization of an energy storage system in a microgrid over the long term

We introduce different methods to solve a "novel class" of stochastic optimization problems, namely two-time scales stochastic optimization problems. Those are optimization problems displaying stochasticity and decisions that have to be made at different paces.

## 2.1 Energy system description and notations

We consider the system sketched in Figure 1. This is a micro-grid with the following features:

1. an electrical load, or demand, that is uncertain (right),

2. solar panels producing uncertain renewable electricity (left),

3. a connection to the national grid if self production is not enough to provide electricity to the load (top),

4. an electrical storage to ensure supply demand balance (bottom).

All the equipment exchange electricity though a DC grid. The arrows in Figure 1 represent the flow of energy: it is bidirectional in the case of the storage as it can charge and discharge. The central node can be seen as a very small storage on a really fast time scale (milliseconds).

The scope of the paper is to propose an Energy Management System (EMS) that controls both the charge/discharge and health of the battery so as to minimize the electricity consumption on the national grid while ensuring a good aging for the battery. We present here a model of this stochastic dynamical system used to design algorithms to implement the EMS.

Figure 1: The schematic representation of the DC micro-grid to be managed

### 2.1.1 Notations for two-time scales

For a given constant time interval $\Delta t$, let $M \in \mathbb{N}^*$ such that $M+1$ is the number of time steps in a day, e.g. for $\Delta t = 60$ seconds, $M+1 = 1440$. The EMS has to make decisions on two-time scales over a given number of days $D \in \mathbb{N}^*$:

1. one battery charge/discharge decision every minute $m \in \{0, \ldots, M\}$ of every day $d \in \{0, \ldots, D\}$,

2. one potential renewal of the battery every day $d \in \{0, \ldots, D+1\}$.

In order to take into account the two-time scales, we adopt in the sequel the following notation. A variable $z$ will have two time indexes $z_{d,m}$ if it changes every minute $m$ of every day $d$. An index $(d, m)$ belongs to the following set

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

which is a totally ordered set when equipped with the *lexicographical order*

$$(d, m) < (d', m') \iff (d < d') \vee (d = d' \wedge m < m') . \tag{2}$$

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

- the notation $z_{d,m:m'}$ is used to refer to the sequence $(z_{d,m}, \ldots, z_{d,m'-1}, z_{d,m'})$,

- the notation $\mathbb{Z}_{d,m:m'}$ is used to refer to the cartesian $\prod_{k=m}^{m'} \mathbb{Z}_{d,k}$.

The following time-line illustrates how time flows between two days in our model:



Figure 2: Time-line

### 2.1.2 Uncertainties are modelled as random variables

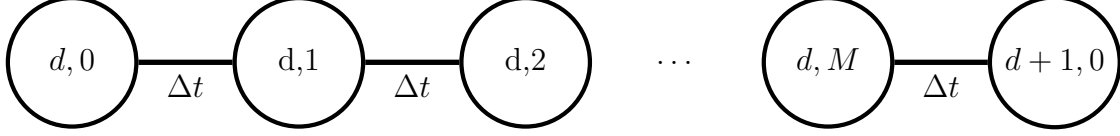We write random variables in capital bold letters, like $\boldsymbol{Z}$, to distinguish them from deterministic variables $z$.

Let $d \in \{0, \ldots, D\}$ be a given day of the whole time span. Every minute $m \in \{0, \ldots, M\}$, two uncertain outcomes materialize at the end of the time interval $[(d, m-1), (d, m))$ (when $m = 0$, the time interval is $[(d-1, M), (d, 0))$, namely,

- $\boldsymbol{E}_{d,m}^{S}$: the solar production in kWh,

- $\boldsymbol{E}_{d,m}^{L}$: the electrical demand (load) in kWh.

Another uncertain outcome realizes once a day at the beginning of the time interval $[(d, 0), (d, M))$ namely,

- $\boldsymbol{P}_{d}^{b}$: the price of a battery replacement in €/kWh.

We gather all uncertainties in vectors and build a sequence of random variables $\{\boldsymbol{W}_{d,m}\}_{(d,m)\in\mathbb{T}}$ as follows. For all $d \in \{0, \ldots, D\}$ we define:

$$\boldsymbol{W}_{d,m} = \begin{pmatrix} \boldsymbol{E}_{d,m}^{S} \\ \boldsymbol{E}_{d,m}^{L} \end{pmatrix}, \text{ for } m \in \{0, \ldots, M-1\}, \quad \text{and} \quad \boldsymbol{W}_{d,M} = \begin{pmatrix} \boldsymbol{E}_{d,M}^{S} \\ \boldsymbol{E}_{d,M}^{L} \\ \boldsymbol{P}_{d}^{b} \end{pmatrix}. \quad (3)$$

We assume in this model that, at the end of the last minute of the day, solar energy production and demand materialize as well as the price of batteries. We call $\mathbb{W}_{d,m}$ the uncertainty space where this uncertainty takes its values.

**Remark 2.1.** *We assume in this model that the "slow" randomness $\boldsymbol{P}_{d}^{b}$ materializes during a minute at the same time as one solar and load "fast" randomness $(\boldsymbol{E}_{d,M}^{S}, \boldsymbol{E}_{d,M}^{L})$. We could add a virtual minute to avoid this.* ◇

7

The decision maker take decisions based on observation of uncertainties but he cannot anticipate on future uncertainties. To describe this non-anticipativity constraint, for each $(d, m) \in \mathbb{T}$, we introduce the $\sigma$-algebra $\mathcal{F}_{d,m}$ generated by all the past noises up to time stage $(d, m)$

$$\mathcal{F}_{d,m} = \sigma\left(\boldsymbol{W}_{d',m'}; (d', m') \leq (d, m)\right). \tag{4}$$

Throughout the paper a random variable indexed by $(d, m)$ refers to a $\mathcal{F}_{d,m}$-measurable random variable (the measurability being imposed by constraints or derived through dynamics equations). The filtration $\{\mathcal{F}_t\}_{t \in \mathbb{T}}$ models the information flow of the problem and the non-anticipativity constraints

$$\forall (d, m) \in \mathbb{T} \quad \sigma(\boldsymbol{U}_{d,m}) \subset \mathcal{F}_{d,m} \tag{5}$$

express the fact that random variables $\boldsymbol{U}_{d,m}$ are adapted to the natural filtration $\{\mathcal{F}_{d,m}\}_{(d,m) \in \mathbb{T}}$, i.e $\boldsymbol{U}_{d,m}$ only depends on uncertainties up to time $(d, m)$. We define precisely the decision variables $\boldsymbol{U}_{d,m}$ in the next paragraph.

### 2.1.3 Decisions are modelled as random variables

As already mentioned, as time goes on, the noise variables are progressively unfolded and made available to the decision maker. This is why, as decisions depend on observations in a stochastic optimal control problem, decision variables are random variables. On a day $d \in \{0, \ldots, D\}$, the EMS has to make decisions every minute $m \in \{0, \ldots, M\}$ regarding the charge/discharge of the battery as well as the electricity consumption on the national grid. These decisions depend on all the randomness unfolded previously, that is, all the prices of batteries of the previous days and all the solar production and load of the previous days and minutes of the day as described in Equation (3)). Finally, these decisions are made at the beginning of the time interval $[(d, m), (d, m+1))$ (when $m = M$ the time interval is $[(d, M), (d+1, 0))$):

- $\boldsymbol{E}_{d,m}^E$: the import from the national grid in kWh;

- $\boldsymbol{E}_{d,m}^B$: the battery charge ($\geq 0$) or discharge ($\leq 0$) in kWh.

At the end of the time interval $[(d, 0), (d, M))$, the decision maker can replace the battery by a new one after observing the current price of batteries $\boldsymbol{P}_d^b$. The decision variable is again a random variable as it depends on the randomness that materialized the previous minutes of previous days:

- $\boldsymbol{R}_d$: the size of the new battery in kWh.

8

For all $d \in \{0, \dots, D\}$, we group all controls in vectors as follows:

$$\text{for } 0 \le m < M , \quad \boldsymbol{U}_{d,m} = \begin{pmatrix} \boldsymbol{E}_{d,m}^E \\ \boldsymbol{E}_{d,m}^B \end{pmatrix} \text{ and } \boldsymbol{U}_{d,M} = \begin{pmatrix} \boldsymbol{E}_{d,M}^E \\ \boldsymbol{R}_d \end{pmatrix} . \tag{6}$$

We assume in this model that at the last minute of the day, the national grid consumption $\boldsymbol{E}_{d,M}^E$ is chosen as well as the potential renewal of the battery $\boldsymbol{R}_d$. In order to take intro account the renewal of the battery in a simplified way, we assume in the model that the battery charge is not chosen at the end of the day. We call $\mathbb{U}_{d,m}$ the control space where the control takes its values.

On Figure 1, we observe that all flows converge to a central node named "DC". At a very small time scale (milliseconds), it could be described as a small energy storage to model the voltage stability problem of the DC micro-grid. In practice, it could be implemented by a controlled DC/DC converter and super-capacitors (see [16]). We do not model this part and assume that the balance constraint (7) is ensured in this problem. It states that at a minute time scale we consider that voltage stability is handled and therefore that we have to ensure energy supply/demand balance at the central node. This materializes as the following constraint:

$$\boldsymbol{E}_{d,m+1}^E + \boldsymbol{E}_{d,m+1}^S = \boldsymbol{E}_{d,m}^B + \boldsymbol{E}_{d,m+1}^L . \tag{7}$$

We observe a difference of indexes between $\boldsymbol{E}_{d,m}^B$ and the other variables. This is due to the fact that battery charge/discharge is to be implementable on a real system. We need to be able to provide a charge/discharge target to the battery controller at the beginning of the minute. The control variable $\boldsymbol{E}_{d,m+1}^E$ is virtual, it is deduced when voltage stability is ensured in the grid at a lower control level. Therefore, we can remove this variable from the optimization problem and replace Equation (6) by the following equation:

$$\text{for } 0 \le m < M , \quad \boldsymbol{U}_{d,m} = \begin{pmatrix} \boldsymbol{E}_{d,m}^B \end{pmatrix} \text{ and } \boldsymbol{U}_{d,M} = \begin{pmatrix} \boldsymbol{R}_d \end{pmatrix} . \tag{8}$$

### 2.1.4  Charge/discharge impacts state of charge and age dynamics

We use a very simple model to describe charging and aging of the battery. We call $\rho_c \in [0,1]$ and $\rho_d \in [0,1]$ respectively the charge and discharge efficiency of the battery. On day $d \in \{0, \dots, D\}$ at minute $m \in \{0, \dots, M\}$, we call $\boldsymbol{B}_{d,m}$ the state of charge of the battery in kWh and $\boldsymbol{H}_{d,m}$ the remaining amount of exchangeable energy in the battery. As we can change a battery only once a day, we call $\boldsymbol{C}_d$ the capacity of the battery. For a given capacity a battery can make up to $N_c(\boldsymbol{C}_d)$ cycles before being considered unusable. At the beginning of the life of the battery with

9

capacity $\boldsymbol{C}_d$, the formula $2 \times N_c(\boldsymbol{C}_d) \times \boldsymbol{C}_d$ gives the maximum health of the battery in kWh. This is the maximum amount of exchangeable energy for the battery. A cycle represents a full charge of the battery plus a full discharge, hence two times the capacity. Every-time we charge or discharge the battery we change its state of charge according to the following dynamical equation.

$$\boldsymbol{B}_{d,m+1} = \boldsymbol{B}_{d,m} - \frac{1}{\rho_d} \boldsymbol{E}_{d,m}^{B-} + \rho_c \boldsymbol{E}_{d,m}^{B+} \, , \tag{9a}$$

where $(x)^+ = 0 \wedge x$ and $(x)^- = 0 \wedge (-x)$. Moreover, its amount of exchangeable energy (or health) decreases according to the following dynamical equation:

$$\boldsymbol{H}_{d,m+1} = \boldsymbol{H}_{d,m} - \boldsymbol{E}_{d,m}^{B-} - \boldsymbol{E}_{d,m}^{B+} \, . \tag{9b}$$

When the battery health reaches zero, it cannot be used anymore. Hence we have the following health constraint

$$0 \le \boldsymbol{H}_{d,m} \, . \tag{10}$$

We constrain the state of charge to remain between two prescribed bounds that are a percentage of the capacity $\boldsymbol{C}_d$:

$$\underline{B} \times \boldsymbol{C}_d \le \boldsymbol{B}_{d,m} \le \overline{B} \times \boldsymbol{C}_d \, . \tag{11}$$

Using Equations (9a) and (9b) repeatedly, we obtain that $\boldsymbol{B}_{d,M}$ (resp. $\boldsymbol{H}_{d,M}$) is a function of $(\boldsymbol{B}_{d,0}, \boldsymbol{U}_{d,0:M-1})$ (resp. $(\boldsymbol{H}_{d,0}, \boldsymbol{U}_{d,0:M-1})$) that we call $f_d^B$ (resp. $f_d^H$):

$$\boldsymbol{B}_{d,M} = f_d^B \left( \boldsymbol{B}_{d,0}, \boldsymbol{U}_{d,0:M-1} \right) \, , \tag{12a}$$

$$\boldsymbol{H}_{d,M} = f_d^H \left( \boldsymbol{H}_{d,0}, \boldsymbol{U}_{d,0:M-1} \right) \, . \tag{12b}$$

### 2.1.5 Battery renewal impacts state dynamics

In this paragraph, we model how the decision to renew the battery using the control $\boldsymbol{U}_{d,M} (= \boldsymbol{R}_d)$ affects the slow state dynamics. If, at the end of day $d$, we replace the battery with capacity $\boldsymbol{C}_d$ by a new battery of capacity $\boldsymbol{R}_d$, then the capacity $\boldsymbol{C}_{d+1}$ becomes equal to $\boldsymbol{R}_d$. Otherwise the capacity remains unchanged. This gives:

$$\boldsymbol{C}_{d+1} = \begin{cases} \boldsymbol{R}_d \, , & \text{if } \boldsymbol{R}_d > 0 \, , \\ \boldsymbol{C}_d \, , & \text{otherwise} \, . \end{cases} \tag{13}$$

10

The renewal decision affects as well the fast variables $\boldsymbol{B}_{d,M}$ as a new battery is assumed empty,

$$\boldsymbol{B}_{d+1,0} = \begin{cases} \underline{B} \times \boldsymbol{R}_d \,, & \text{if } \boldsymbol{R}_d > 0 \,, \\ \boldsymbol{B}_{d,M} \,, & \text{otherwise} \,, \end{cases} \tag{14}$$

and $\boldsymbol{H}_{d,M}$ as a new battery has a renewed health,

$$\boldsymbol{H}_{d+1,0} = \begin{cases} 2 \times N_c(\boldsymbol{R}_d) \times \boldsymbol{R}_d \,, & \text{if } \boldsymbol{R}_d > 0 \,, \\ \boldsymbol{H}_{d,M} \,, & \text{otherwise} \,. \end{cases} \tag{15}$$

We group these state variables at the begining of a day $d$ under the name $\boldsymbol{X}_d$:

$$\boldsymbol{X}_d = \begin{pmatrix} \boldsymbol{C}_d \\ \boldsymbol{B}_{d,0} \\ \boldsymbol{H}_{d,0} \end{pmatrix} \,. \tag{16}$$

We call $\mathbb{X}_d$ the state space where this state takes its values. We build a mapping

$$f_d^S : \mathbb{C}_d \times \mathbb{B}_{d,M} \times \mathbb{H}_{d,M} \times \mathbb{U}_{d,M} \to \mathbb{X}_{d+1}$$

$$(c, b, h, u) \mapsto \begin{cases} (u, \underline{B}u, 2N_c(u)u) & \text{if } u > 0 \,, \\ (c, b, h) & \text{otherwise} \,. \end{cases} \tag{17}$$

We thus obtain a state dynamics equation given by

$$\boldsymbol{X}_{d+1} = f_d^S\big(\boldsymbol{C}_d, \boldsymbol{B}_{d,M}, \boldsymbol{H}_{d,M}, \boldsymbol{U}_{d,M}\big) \text{ using (13-15) and (17)} \tag{18a}$$

$$= f_d^S\big(\boldsymbol{C}_d, f_d^B\big(\boldsymbol{B}_{d,0}, \boldsymbol{U}_{d,0:M-1}\big), f_d^H\big(\boldsymbol{H}_{d,0}, \boldsymbol{U}_{d,0:M-1}\big), \boldsymbol{U}_{d,M}\big) \text{ using (12)} \tag{18b}$$

$$= f_d\big(\boldsymbol{X}_d, \boldsymbol{U}_{d,0:M}\big) \tag{18c}$$

with

$$f_d\big((c_d, b_{d,0}, h_{d,0}), u_{d,0:M}\big) = f_d^S\big(c_d, f_d^B\big(b_{d,0}, u_{d,0:M-1}\big), f_d^H\big(h_{d,0}, u_{d,0:M-1}\big), u_{d,M}\big) \,. \tag{18d}$$

**Remark 2.2.** *We note that, in our model, the state dynamics does not depend directly on uncertainties $\boldsymbol{W}_{t+1}$.* $\diamond$

11

## 2.2  Stochastic optimization problem statement

We have introduced all the requested features to state a two-time scale stochastic optimal control problem dynamics. It remains to define the objective function that the EMS seeks to minimize.

The objective is a discounted expected sum over a finite horizon. We consider the following objective to be minimized:

$$\mathbb{E}\Big[ \sum_{d=0}^{D} \gamma_d \Big( \boldsymbol{P}_d^b \times \boldsymbol{R}_d + \sum_{m=0}^{M-1} p_{d,m}^e \times (\boldsymbol{E}_{d,m}^B + \boldsymbol{E}_{d,m+1}^L - \boldsymbol{E}_{d,m+1}^S)^+ \Big) \Big] . \tag{19}$$

We now comment each term. Over the whole daily horizon $D$, the decision maker wants to minimize a discounted sum of all his expenses, that is, the battery renewals and the national grid energy consumption. The first term of the sum over days $\boldsymbol{P}_d^b \times \boldsymbol{R}_d$ is the cost incurred by a battery renewal during day $d$. The second term $\sum_{m=0}^{M-1} p_{d,m}^e \times \boldsymbol{E}_{d,m+1}^{E}{}^+$ is a sum of the national grid energy consumption every minute of the day, where $E_{d,m+1}^E$ is eliminated using Equation (7). We take the positive part, denoted by $^+$, assuming that an excessive production of solar energy is wasted. The sum is discounted by an arbitrary discount factor $\gamma_d$. In the sequel, the discount factor $\gamma_d$ changes once a year to model a discount rate of $\tau = 4.5\%$ every year

$$\gamma_d = \Big( \frac{1}{1+\tau} \Big)^{\lfloor d/365 \rfloor - 1} . \tag{20}$$

Using Equations (3) and (8), we obtain that the expectation cost given by Equation (19) can be rewritten as

$$\mathbb{E}\Big[ \sum_{d=0}^{D} L_d(\boldsymbol{X}_d, \boldsymbol{U}_d, \boldsymbol{W}_d) + K(\boldsymbol{X}_{D+1}) \Big] \tag{21}$$

$$= \mathbb{E}\Big[ \sum_{d=0}^{D} \gamma_d \Big( \boldsymbol{W}_{d,M}^3 \boldsymbol{U}_{d,M} + \sum_{m=0}^{M-1} p_{d,m}^e \times (\boldsymbol{U}_{d,m} + \boldsymbol{W}_{d,m+1}^2 - \boldsymbol{W}_{d,m+1}^1) \Big) \Big] ,$$

where the final cost $K$ is null and the intraday cost $L_d$ is given by

$$L_d : \mathbb{X}_d \times \mathbb{U}_{d,0:M} \times \mathbb{W}_{d,0:M} \to (-\infty, +\infty] , \tag{22}$$

$$(x_d, u_d, w_d) \mapsto \gamma_d \Big( w_{d,M}^3 u_{d,M} + \sum_{m=0}^{M-1} p_{d,m}^e \big( u_{d,m} + w_{d,m+1}^2 - w_{d,m+1}^1 \big) \Big) .$$

12

# 3 Two algorithms for two-time scales stochastic optimal control problems

We introduce hereby a generic two-time scales stochastic optimization problem. For the sake of simplicity we assume that the constraints on states and controls, that are not a dynamic or a non anticipativity constraint, are placed in the instantaneous costs $L_d$ using characteristic functions taking the value $+\infty$. Gathering all the above equations, we can state the optimization problem to be solved:

$$V(x) = \min_{\boldsymbol{X}_{0:D+1},\ \boldsymbol{U}_{0:D}} \mathbb{E}\Big[\sum_{d=0}^{D} L_d(\boldsymbol{X}_d, \boldsymbol{U}_d, \boldsymbol{W}_d) + K(\boldsymbol{X}_{D+1})\Big]\,, \tag{23a}$$

$$\text{s.t } \boldsymbol{X}_{d+1} = f_d(\boldsymbol{X}_d, \boldsymbol{U}_d, \boldsymbol{W}_d)\,, \tag{23b}$$

$$\boldsymbol{U}_d = (\boldsymbol{U}_{d,0}, \ldots, \boldsymbol{U}_{d,m}, \ldots, \boldsymbol{U}_{d,M})\,, \tag{23c}$$

$$\boldsymbol{W}_d = (\boldsymbol{W}_{d,0}, \ldots, \boldsymbol{W}_{d,m}, \ldots, \boldsymbol{W}_{d,M})\,, \tag{23d}$$

$$\sigma(\boldsymbol{U}_{d,m}) \subset \sigma\big(\boldsymbol{W}_{d',m'}; (d', m') \leq (d, m)\big) \tag{23e}$$

$$\boldsymbol{X}_0 = x\,. \tag{23f}$$

The daily cost $L_d$ is given by Equation (22), the final cost is equal to zero and the state dynamics between days $f_d$ is given by Equation (18c). Note that the notation $\boldsymbol{X}_d$ refers to the state random variable at the minute $(d, 0)$ while the notation $\boldsymbol{U}_d$ and $\boldsymbol{W}_d$ refer respectively to random decision and uncertainty vectors containing all decisions and uncertainties of the day $d$.

As stated in Problem (23), the optimization problem is very similar to a classical discrete time stochastic optimal control problem, except for the non anticipativity constraint (5) that expresses the fact that the decision vector $\boldsymbol{U}_d = (\boldsymbol{U}_{d,0}, \ldots, \boldsymbol{U}_{d,M})$ at every time step $d$ does not display the same measurability for each component (information grows every minute).

We present in this part two methods to decompose the two-time scales stochastic optimal control problem (23). We apply the decomposition schemes to design tractable algorithms to compute suboptimal policies and values for that kind of problems.

## 3.1 Time blocks decomposition

We introduce a daily independence assumption in order to obtain a day by day decomposition of the optimization problem (23), that is, a dynamic programming

equation between days. We assume that the sequence of random vectors $\left\{\boldsymbol{W}_d\right\}_{d=0,\dots,D}$ is constituted of independent random variables. However, note that we do not assume that each random vector $\boldsymbol{W}_d = (\boldsymbol{W}_{d,0}, \dots, \boldsymbol{W}_{d,M})$ is itself composed of independent random variables.

**Assumption 3.1.** *The sequence $\left\{\boldsymbol{W}_d\right\}_{d=0,\dots,D}$ is a sequence of independent random vectors.*

We introduce a sequence of *slow time scale value functions*, $\{V_d\}_{d\in\{0,\dots,D+1\}}$, defined by backward induction as follows. At time $D+1$, we set

$$V_{D+1} = K \, , \tag{24a}$$

and then for $d \in \{0, \dots, D\}$ we define by backward induction

$$V_d(x) = \min_{\boldsymbol{X}_{d+1}, \boldsymbol{X}_d, \boldsymbol{U}_d} \mathbb{E}\left[L_d(\boldsymbol{X}_d, \boldsymbol{U}_d, \boldsymbol{W}_d) + V_{d+1}(\boldsymbol{X}_{d+1})\right] \, , \tag{24b}$$

$$\text{s.t } \boldsymbol{X}_{d+1} = f_d(\boldsymbol{X}_d, \boldsymbol{U}_d, \boldsymbol{W}_d) \, , \tag{24c}$$

$$\boldsymbol{U}_d = (\boldsymbol{U}_{d,0}, \dots, \boldsymbol{U}_{d,m}, \dots, \boldsymbol{U}_{d,M}) \, , \tag{24d}$$

$$\boldsymbol{W}_d = (\boldsymbol{W}_{d,0}, \dots, \boldsymbol{W}_{d,m}, \dots, \boldsymbol{W}_{d,M}) \, , \tag{24e}$$

$$\sigma(\boldsymbol{U}_{d,m}) \subset \sigma(\boldsymbol{X}_d, \boldsymbol{W}_{d,0:m}) \, , \tag{24f}$$

$$\boldsymbol{X}_d = x \, . \tag{24g}$$

Let $d \in \{0, \dots, D\}$ be fixed. To each given pair $(x_d, \boldsymbol{X}_{d+1}) \in \mathbb{X}_d \times L^0(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{X}_{d+1})$, we associate an optimization problem denoted by $\mathcal{P}_{(d,=)}\big[x_d, \boldsymbol{X}_{d+1}\big]$ and given by:

$$\mathcal{P}_{(d,=)}\big[x, \boldsymbol{X}\big] \begin{cases} \min_{\boldsymbol{X}_d, \boldsymbol{U}_d} \mathbb{E}\Big[L_d(\boldsymbol{X}_d, \boldsymbol{U}_d, \boldsymbol{W}_d)\Big] \, , & (25a) \\[2mm] \text{s.t } f_d(\boldsymbol{X}_d, \boldsymbol{U}_d, \boldsymbol{W}_d) = \boldsymbol{X} \, , & (25b) \\[1mm] \boldsymbol{U}_d = (\boldsymbol{U}_{d,0}, \dots, \boldsymbol{U}_{d,m}, \dots, \boldsymbol{U}_{d,M}) \, , & (25c) \\[1mm] \boldsymbol{W}_d = (\boldsymbol{W}_{d,0}, \dots, \boldsymbol{W}_{d,m}, \dots, \boldsymbol{W}_{d,M}) \, , & (25d) \\[1mm] \sigma(\boldsymbol{U}_{d,m}) \subset \sigma(\boldsymbol{X}_d, \boldsymbol{W}_{d,0:m}) \, , & (25e) \\[1mm] \boldsymbol{X}_d = x \, . & (25f) \end{cases}$$

The value of the optimization Problem (25) is denoted by $\phi_{(d,=)}(x, \boldsymbol{X})$ and we call this optimization problem *the intraday optimization problem* with equality target. Adopting standard conventions, the value function $\phi_{(d,=)}$ will take the value $+\infty$, when Problem (25) does not have an admissible solution for a given pair $(x, \boldsymbol{X})$.

14

**Proposition 3.2.** *Under Assumption 3.1, the value function $V$ solution of optimization problem* (23) *coincides with the value function $V_0$ given by the Bellman equation* (24). *Moreover, the sequence of value functions given by Equation* (24) *coincides with the sequence of mappings given by the following backward induction:*

$$V_{D+1} = K \tag{26a}$$

$$\forall x \in \mathbb{X}_d, \quad V_d(x) = \min_{\boldsymbol{X} \in L^0(\Omega,\mathcal{F},\mathbb{P};\mathbb{X}_{d+1})} \left( \phi_{(d,=)}(x, \boldsymbol{X}) + \mathbb{E}\left[ V_{d+1}(\boldsymbol{X}) \right] \right), \tag{26b}$$

$$s.t \; \sigma(\boldsymbol{X}) \subset \sigma(\boldsymbol{W}_d) . \tag{26c}$$

*Proof.* Under Assumption 3.1, the optimal value of Problem (23) remains unchanged when the non anticipativity constraint (23e) is replaced by the constraint:

$$\sigma(\boldsymbol{U}_{d,m}) \subset \sigma(\boldsymbol{X}_d, \boldsymbol{W}_{d,0:m}) . \tag{27}$$

Then, the fact that the backward induction (24) is the Bellman equation which gives the solution of Problem (23) is detailed in [7]. Exploiting the linearity of the expectation and the fact that minimization can be done sequentially, we rewrite Equation (24) as

$$V_d(x) = \min_{\boldsymbol{X}_{d+1} \in L^0(\Omega,\mathcal{F},\mathbb{P};\mathbb{X}_{d+1})} \min_{\boldsymbol{U}_d, \boldsymbol{X}_d} \mathbb{E}\left[ L_d(\boldsymbol{X}_d, \boldsymbol{U}_d, \boldsymbol{W}_d) \right] + \mathbb{E}\left[ V_{d+1}(\boldsymbol{X}_{d+1}) \right],$$

$$s.t \; (24c)\text{-}(24g) ,$$

$$= \min_{\boldsymbol{X}_{d+1} \in L^0(\Omega,\mathcal{F},\mathbb{P};\mathbb{X}_{d+1})} \left( \phi_{(d,=)}(x, \boldsymbol{X}_{d+1}) + \mathbb{E}\left[ V_{d+1}(\boldsymbol{X}_{d+1}) \right] \right), \tag{28}$$

Moreover, if $\boldsymbol{X}_{d+1} \in \text{dom}\left( \phi_{(d,=)}(x, \cdot) \right)$, then $\boldsymbol{X}_{d+1}$ is given by Equation (24c) and thus it is a $\sigma(\boldsymbol{W}_d)$-measurable random variable. Therefore, adding Constraint (26c) in the optimization problem (28) yields the same optimization problem. This ends the proof. $\qquad\square$

## 3.2 Stochastic targets decomposition algorithm

The numerical resolution of intraday problem (25) is most of the time out of reach due to the target constraint (25b). In order to compute approximations of the daily value functions (24), we present simplified versions of Problem (25). We introduce a relaxation of the target constraint (25b), turning the equality into an inequality. Furthermore that makes possible to look for deterministic targets instead of stochastic ones which simplifies the information constraint (26c). We apply the general results in § 5.1 to the slow scale Bellman equation (26).

15

### 3.2.1 Relaxed intraday optimization problem

For each $d \in \{0, \ldots, D\}$, we introduce a relaxed intraday optimization problem, $\mathcal{P}_{(d,\geq)}[x, \boldsymbol{X}]$, which is obtained by considering the optimization problem (25) whith the equality target (25b) replaced by the following inequality target (29b):

$$
\mathcal{P}_{(d,\geq)}[x, \boldsymbol{X}]
\begin{cases}
\min_{\boldsymbol{X}_d, \boldsymbol{U}_d} \mathbb{E}\Big[L_d(\boldsymbol{X}_d, \boldsymbol{U}_d, \boldsymbol{W}_d)\Big] \, , & \text{(29a)} \\[1mm]
\text{s.t } f_d(\boldsymbol{X}_d, \boldsymbol{U}_d, \boldsymbol{W}_d) \geq \boldsymbol{X} \, , & \text{(29b)} \\[1mm]
\boldsymbol{U}_d = (\boldsymbol{U}_{d,0}, \ldots, \boldsymbol{U}_{d,m}, \ldots, \boldsymbol{U}_{d,M}) \, , & \text{(29c)} \\[1mm]
\boldsymbol{W}_d = (\boldsymbol{W}_{d,0}, \ldots, \boldsymbol{W}_{d,m}, \ldots, \boldsymbol{W}_{d,M}) \, , & \text{(29d)} \\[1mm]
\sigma(\boldsymbol{U}_{d,m}) \subset \sigma(\boldsymbol{X}_d, \boldsymbol{W}_{d,0:m}) \, , & \text{(29e)} \\[1mm]
\boldsymbol{X}_d = x \, . & \text{(29f)}
\end{cases}
$$

We denote by $\phi_{(d,\geq)}(x, \boldsymbol{X})$ the value of the relaxed optimization problem $\mathcal{P}_{(d,\geq)}[x, \boldsymbol{X}]$. We associate to the relaxed value function $\phi_{(d,\geq)}(x, \boldsymbol{X})$ a sequence of relaxed Bellman value functions $\{V_{(d,\geq)}\}_{d \in \{0, \ldots, D+1\}}$ defined as follows:

$$
V_{(D+1,\geq)} = K \, , \tag{30a}
$$

and for all $d \in \{0, \ldots, D\}$, and for all $x \in \mathbb{X}_d$

$$
V_{(d,\geq)}(x) = \min_{\boldsymbol{X} \in L^0(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{X}_{d+1})} \Big( \phi_{(d,\geq)}(x, \boldsymbol{X}) + \mathbb{E}\big[V_{(d+1,\geq)}(\boldsymbol{X})\big] \Big) \, , \tag{30b}
$$

$$
\text{s.t } \sigma(\boldsymbol{X}) \subset \sigma(\boldsymbol{W}_d) \, . \tag{30c}
$$

We then consider the case where, in Equation (30), the minimization over the space $L^0(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{X}_{d+1})$ is replaced by minimization over the constants $x \in \mathbb{X}_{d+1}$; we denote by $\{V_{(d,\geq,\mathbb{X}_{d+1})}\}_{d \in \{0, \ldots, D+1\}}$ the associated sequence of Bellman functions.

$$
V_{(D+1,\geq,\mathbb{X}_{D+2})} = K \, , \tag{31a}
$$

and for all $d \in \{0, \ldots, D\}$, and for all $x \in \mathbb{X}_d$

$$
V_{(d,\geq,\mathbb{X}_{d+1})}(x) = \min_{X \in \mathbb{X}_{d+1}} \Big( \phi_{(d,\geq)}(x, X) + V_{(d+1,\geq,\mathbb{X}_{d+2})}(X) \Big) \, . \tag{31b}
$$

The undefined state space $\mathbb{X}_{D+2}$ in (31a) is introduced for consistency with recursive equation (31b). It can be any space as it is not used in Equation (31a).

**Assumption 3.3.** *The value functions $\{V_d\}_{d=0,\ldots,D}$ are non-increasing.*

We show in Proposition 3.4, that under Assumption 3.3, the value functions $V_{(d,\geq,\mathbb{X}_{d+1})}$ give upper bounds to the original value functions $V_d$ in (28).

**Proposition 3.4.** *The sequence of relaxed Bellman value functions $\{V_{(d,\geq)}\}_{d\in\{0,\ldots,D+1\}}$ given by Equation (30) gives lower bound to the sequence of value functions $\{V_d\}_{d\in\{0,\ldots,D+1\}}$ given by Equation (24). That is, for all $d \in \{0,\ldots,D+1\}$, we have*

$$V_{(d,\geq)} \leq V_d . \tag{32}$$

*Moreover, under Assumption 3.3 we have for all $d \in \{0,\ldots,D\}$ that*

$$V_d = V_{(d,\geq)} \leq V_{(d,\geq,\mathbb{X}_{d+1})} . \tag{33}$$

*Proof.* Let $d \in \{0,\ldots,D-1\}$ and a pair $(x_d, \boldsymbol{X}_{d+1}) \in \mathbb{X}_d \times L^0(\Omega,\mathcal{F},\mathbb{P};\mathbb{X}_{d+1})$ given. We have that

$$\phi_{(d,\geq)}(x_d, \boldsymbol{X}_{d+1}) \leq \phi_{(d,=)}(x_d, \boldsymbol{X}_{d+1}) . \tag{34}$$

From Equations (28) and (30), we obtain by backward induction that for all $d \in \{0,\ldots,D+1\}$

$$V_{(d,\geq)} \leq V_d . \tag{35}$$

Now, given $d \in \{0,\ldots,D-1\}$, since the set of constant random variables taking values in $\mathbb{X}_{d+1}$ is a subset of $L^0(\Omega,\mathcal{F},\mathbb{P};\mathbb{X}_{d+1})$ we obtain that $V_{(d,\geq)} \leq V_{(d,\geq,\mathbb{X}_{d+1})}$. Thus, the only point to prove is that under Assumption 3.3 we have the equality $V_d = V_{(d,\geq)}$. We proceed by backward induction. At time $D+1$, the two mappings $V_{(D+1,\geq)}$ and $V_{D+1}$ are both equal to $K$ which is non-increasing. Then, let $d$ be fixed in $\{0,\ldots,D\}$ and assume that $V_{(d+1,\geq)} = V_{d+1}$ and that these two value functions are non-increasing. We prove that $V_{(d,\geq)}$ and $V_d$ coincides using Lemma 5.3 which applies since $\mathbb{X}_{d+1}$ is a subset of some finite dimensional space $\mathbb{R}^{n_x}$. $\square$

**Remark 3.5.** *Looking for deterministic targets instead of stochastic targets is made possible by the fact that we relaxed the almost sure target equality constraint (29b) into an inequality using the value functions monotonicity. An almost sure equality constraint requires both sides to have the same measurability, we would have to ensure that a random variable is always equal to a deterministic one which is most of the time impossible.* $\diamond$

### 3.2.2 Statement of the algorithm with deterministic targets and periodicity classes

In order to compute the daily value functions upper bounds $\{V_{(d,\geq,\mathbb{X}_{d+1})}\}_{d=0,\dots,D+1}$ via Equation (25), we need the value of the relaxed intraday problems $\phi_{(d,\geq)}(x_d, x_{d+1})$ for all $d \in \{0,\dots,D+1\}$ and for all pairs $((x_d, x_{d+1}) \in \mathbb{X}_d \times \mathbb{X}_{d+1}$, where we recall that $\phi_{(d,\geq)}(x_d, x_{d+1})$ is given by

$$\phi_{(d,\geq)}(x_d, x_{d+1}) = \min_{\boldsymbol{U}_d} \mathbb{E}\Big[L_d(x_d, \boldsymbol{U}_d, \boldsymbol{W}_d)\Big] , \tag{36a}$$

$$\text{s.t } f_d(x_d, \boldsymbol{U}_d, \boldsymbol{W}_d) \geq x_{d+1} , \tag{36b}$$

$$\sigma(\boldsymbol{U}_{d,m}) \subset \sigma(\boldsymbol{W}_{d,0:m}) . \tag{36c}$$

The computational cost can be significant as we need to solve a stochastic optimization problem for every pair $(x_d, x_{d+1}) \in \mathbb{X}_d \times \mathbb{X}_{d+1}$ and for every $d$ in $\{0,\dots,D\}$. We present a simplification exploiting periodicity of the intraproblems.

**Lemma 3.6.** *Let* $\mathbb{I} \subset \{0,\dots,D\}$. *Assume that there exists two sets* $\mathbb{X}_{\mathbb{I}}$ *and* $\mathbb{U}_{\mathbb{I}}$ *such that for all* $d \in \mathbb{I}$, $\mathbb{X}_d = \mathbb{X}_{\mathbb{I}}$ *and* $\mathbb{U}_d = \mathbb{U}_{\mathbb{I}}$. *Assume moreover than there exists two mappings* $L_{\mathbb{I}}$ *and* $f_{\mathbb{I}}$ *such that for all* $d \in \mathbb{I}$, $L_d = L_{\mathbb{I}}$ *and* $f_d = f_{\mathbb{I}}$. *Finally assume that the random variables* $\{\boldsymbol{W}_d\}_{d\in\mathbb{I}}$ *are independent and identically distributed. Then, there exists a function* $\phi_{\mathbb{I}}$ *such that for all* $d \in \mathbb{I}$

$$\phi_d = \phi_{\mathbb{I}} . \tag{37}$$

*Proof.* The proof is immediate. $\qquad\square$

The set $\mathbb{I}$ introduced in Lemma 3.6 is called a periodicity class. We call $N_p$ the number of periodicity classes of Problem (23) and $(\mathbb{I}_1,\dots,\mathbb{I}_{N_p})$ the periodicity classes, that is, the sets of day indices that satisfy (37).

**Remark 3.7.** *When there is no periodicity,* $N_p = D+1$ *and the periodicity classes are singletons. In this case all the intraday problems have to be computed.* $\qquad\Diamond$

**Remark 3.8.** *A periodicity property often appears in long term energy management problems with renewable energies, due to seasonality of natural processes such as solar production. In these cases* $N_p < D+1$ *and it is enough to solve only* $N_p$ *intraproblems.* $\qquad\Diamond$

The algorithm to compute daily value functions approximations, with relaxed intraday problems, deterministic targets and periodicity classes, is presented in Algorithm 1.

---

**Algorithm 1:** Two-time scales dynamic programming with deterministic targets and periodicity classes

---

**Data:** Periodicity classes $(\mathbb{I}_1, \ldots, \mathbb{I}_{N_p})$
**Result:** Daily value functions approximations $(V_{(d,\geq,\mathbb{X}_{d+1})})_{d=0,\ldots,D+1}$
Initialization: $V_{(D+1,\geq,\mathbb{X}_{D+2})} = K$;
**for** $i = 1, \ldots, N_p$ **do**
    Let $d \in \mathbb{I}_i$;
    **for** $(x_d, x_{d+1}) \in \mathbb{X}_d \times \mathbb{X}_{d+1}$ **do**
        Compute $\phi_{(d,\geq)}(x_d, x_{d+1})$;
    **end**
**end**
**for** $d = D, D-1 \ldots, 0$ **do**
    **for** $x_d \in \mathbb{X}_d$ **do**
        Solve $V_{(d,\geq,\mathbb{X}_{d+1})}(x_d) = \min_{x_{d+1} \in \mathbb{X}_{d+1}} \phi_{(d,\geq)}(x_d, x_{d+1}) + V_{(d+1,\geq,\mathbb{X}_{d+2})}(x_{d+1})$;
    **end**
**end**

---

### 3.2.3 Two further simplifications for the intraday problems computation

Two particular properties can be exploited to lower further the computational burden of the sequence of intraday problems $\phi_{(d,\geq)}$.

**Initial state, final target pair dimension reduction.** We introduce a first particular property of some problems allowing to lower the computational burden of the intraproblems by reducing the dimensionality of the initial state/target pair.

**Assumption 3.9.**

    *1.* $\mathbb{X}_d = \mathbb{X}_{d+1}$,

    *2.* $L_d(x_d, \boldsymbol{U}_d, \boldsymbol{W}_d) = l_d(\boldsymbol{U}_d, \boldsymbol{W}_d)$,

    *3.* $f_d(x_d, \boldsymbol{U}_d, \boldsymbol{W}_d) = x_d + g_d(\boldsymbol{U}_d, \boldsymbol{W}_d)$.

Under Assumption 3.9, it is enough to solve, the optimization problem for all

$x \in \mathbb{X}_d - \mathbb{X}_{d+1}$ (instead of each $(x_d, x_{d+1}) \in \mathbb{X}_d \times \mathbb{X}_{d+1}$)

$$\overline{\phi}_{(d,\geq)}(x) = \min_{\boldsymbol{U}_d} \mathbb{E}\Big[L_d(x, \boldsymbol{U}_d, \boldsymbol{W}_d)\Big] \, , \tag{38a}$$

$$\text{s.t } f_d(x, \boldsymbol{U}_d, \boldsymbol{W}_d) \geq 0 \, , \tag{38b}$$

$$\sigma(\boldsymbol{U}_{d,m}) \subset \sigma(\boldsymbol{W}_{d,0:m}) \, . \tag{38c}$$

**Convexity and stagewise independence assumption.** When the state dynamics $f_d$ are linear and costs $L_d$ and $K$ are convex in $(x, u)$, we can use Stochastic Dual Dynamic Programming (SDDP) to solve the intraday problems, assuming stagewise independence of the intraday noises $(\boldsymbol{W}_{d,0}, \ldots, \boldsymbol{W}_{d,M})$. We obtain a convex polyhedral lower approximation of $\phi_{(d,\geq)}$. This convex polyhedral lower approximation can be represented by a linear program hence it makes it possible to compute a piecewise linear lower approximation $\underline{V}_{(d,\geq,\mathbb{X}_{d+1})}$ of the daily value functions upper bounds $V_{(d,\geq,\mathbb{X}_{d+1})}$ using Linear Programming (LP).

**Remark 3.10.** *We are not guaranteed that* $V_{(d,\geq)} \leq \underline{V}_{(d,\geq,\mathbb{X}_{d+1})}$. $\diamond$

Now we can compute efficiently intraday problems and lower bounds for the daily value functions using a deterministic target decomposition. We present another method to compute value functions for a two-time scales stochastic optimization problem relying on a deterministic weights decomposition.

## 3.3 Stochastic adaptative weights algorithm

In this part we investigate an algorithm based on applying Fenchel-Rockafellar duality [28, 27] to the dynamic programming equation with targets (26), in particular to the target constraint (25b). This method is connected to the one developed in [15] called "adaptative weights", hence the name " Stochastic Adaptative Weights" (SAWA). We extend their results in a stochastic setting and a more general framework as we are not tied to a battery management problem. Furthermore we use well known duality results to reach similar conclusions.

We introduce the dualized intraday problems, whose value is called $\psi_d$ for $d \in \{0, \ldots, D-1\}$, such that for all $(x_d, \boldsymbol{\lambda}_{d+1}) \in \mathbb{X}_d \times L^q(\Omega, \mathcal{F}, \mathbb{P}; \Lambda_{d+1})$, where $\Lambda_{d+1}$ is

the dual space of $\mathbb{X}_{d+1}$ ($\Lambda_{d+1} = \mathbb{R}^{n_x}$ if $\mathbb{X}_{d+1} = \mathbb{R}^{n_x}$):

$$\psi_d(x_d, \boldsymbol{\lambda}_{d+1}) = \min_{\boldsymbol{X}_d, \boldsymbol{U}_d} \mathbb{E}\left[L_d(\boldsymbol{X}_d, \boldsymbol{U}_d, \boldsymbol{W}_d) + \langle \boldsymbol{\lambda}_{d+1}, f_d(\boldsymbol{X}_d, \boldsymbol{U}_d, \boldsymbol{W}_d) \rangle\right], \tag{39a}$$

$$\text{s.t } \boldsymbol{U}_d = (\boldsymbol{U}_{d,0}, \dots, \boldsymbol{U}_{d,m}, \dots, \boldsymbol{U}_{d,M}), \tag{39b}$$

$$\boldsymbol{W}_d = (\boldsymbol{W}_{d,0}, \dots, \boldsymbol{W}_{d,m}, \dots, \boldsymbol{W}_{d,M}), \tag{39c}$$

$$\sigma(\boldsymbol{U}_{d,m}) \subset \sigma(\boldsymbol{X}_d, \boldsymbol{W}_{d,0:m}), \tag{39d}$$

$$\boldsymbol{X}_d = x_d. \tag{39e}$$

We assume that for all $d \in \{0, \dots, D\}$, for any state $x_d \in \mathbb{X}_d$, control $\boldsymbol{U}_d$ and uncertainty $\boldsymbol{W}_d$, the random variable $f_d(x_d, \boldsymbol{U}_d, \boldsymbol{W}_d)$ belongs to $L^p(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{X}_{d+1})$ with $1 < p < +\infty$ and $\frac{1}{p} + \frac{1}{q} = 1$.

For any state $x_d$, admissible control $\boldsymbol{U}_d$ and uncertainty $\boldsymbol{W}_d$, $f(x_d, \boldsymbol{U}_d, \boldsymbol{W}_d)$ is measurable with respect to $\sigma(\boldsymbol{W}_d)$ due to the non anticipativity constraint (39d). Hence for any random variable $\boldsymbol{\lambda}_{d+1} \in L^q(\Omega, \mathcal{F}, \mathbb{P}; \Lambda_{d+1})$ we have the following equality that make it possible to restrict the measurability of dual variables [5, Chap. 5.5]:

$$\psi_d(x_d, \boldsymbol{\lambda}_{d+1}) = \psi_d\left(x_d, \mathbb{E}\left[\boldsymbol{\lambda}_{d+1} | \sigma(\boldsymbol{W}_d)\right]\right). \tag{40}$$

Then, we introduce the following daily value functions,

$$\underline{V}_{D+1} = K, \tag{41a}$$

and, for all $d \in \{0, \dots, D\}$, and for all $x_d \in \mathbb{X}_d$,

$$\underline{V}_d(x_d) = \sup_{\boldsymbol{\lambda}_{d+1} \in \Lambda_{d+1}} \psi_d(x_d, \boldsymbol{\lambda}_{d+1}) - \mathbb{E}\left[\underline{V}_{d+1}^{\star}(\boldsymbol{\lambda}_{d+1})\right], \tag{41b}$$

$$\text{s.t } \sigma(\boldsymbol{\lambda}_{d+1}) \subset \sigma(\boldsymbol{W}_d), \tag{41c}$$

where $\underline{V}_{d+1}^{\star}$ is the Fenchel transform of the function $\underline{V}_{d+1}$. We prove, in the next proposition, that the value function $\underline{V}_d$ gives a lower bound to the value function $V_d$.

**Lemma 3.11.** *For every $d \in \{0, \dots, D\}$,*

$$\underline{V}_d \leq V_d. \tag{42}$$

*Proof.* We apply Lemma 5.6 to $\phi_{(d,=)}(x_d, \cdot)$ and $\mathbb{E}\left[V_{d+1}\right]$. $\square$

**Proposition 3.12.** *Assume that $K$ is convex and that for $d \in \{0, \ldots, D\}$ the instantaneous costs $L_d$ are jointly convex in $x$ and $u$ and that the dynamics $f_d$ are jointly linear in $x$ and $u$. If moreover $\mathrm{ri}\Big(\mathrm{dom}\big(\phi_{(d,=)}(x_d, \cdot)\big) - \mathrm{dom}\big(\mathbb{E}[V_{d+1}]\big)\Big) \neq \emptyset$, then we have the equality*

$$V_d = \underline{V}_d \ . \tag{43}$$

*Proof.* Under the convexity assumptions we are ensured that for all $d$, $\phi_{(d,=)}$ and $V_d$ are convex. We apply Proposition 5.7 to $\phi_{d,=}(x_d, \cdot)$ and $\mathbb{E}\Big[V_{d+1}\Big]$ to obtain that, for all $d \in \{0, \ldots, D\}$, and for all $x_d \in \mathbb{X}_d$:

$$\underline{V}_d(x_d) = \sup_{\boldsymbol{\lambda}_{d+1}} \psi_d(x_d, \boldsymbol{\lambda}_{d+1}) - \Big(\mathbb{E}\Big[\underline{V}_{d+1}\Big]\Big)^{\star}(\boldsymbol{\lambda}_{d+1}) \ , \tag{44a}$$

where

$$\Big(\mathbb{E}\Big[\underline{V}_{d+1}\Big]\Big)^{\star}(\boldsymbol{\lambda}_{d+1}) = \sup_{\boldsymbol{X} \in L^p(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{X}_{d+1})} \langle \boldsymbol{\lambda}_{d+1}, \boldsymbol{X} \rangle - \mathbb{E}\Big[\underline{V}_{d+1}\Big](\boldsymbol{X}) \ , \tag{45}$$

$$\text{s.t } \sigma(\boldsymbol{X}) \subset \sigma(\boldsymbol{W}_d) \ . \tag{46}$$

Due to the property (40), this is equivalent to

$$\underline{V}_d(x_d) = \sup_{\boldsymbol{\lambda}_{d+1}} \psi_d(x_d, \boldsymbol{\lambda}_{d+1}) - \Big(\mathbb{E}\Big[\underline{V}_{d+1}\Big]\Big)^{\star}(\boldsymbol{\lambda}_{d+1}) \ , \tag{47a}$$

$$\text{s.t } \sigma(\boldsymbol{\lambda}_{d+1}) \subset \sigma(\boldsymbol{W}_d) \ . \tag{47b}$$

Finally we need to invert Fenchel transform and expectation in (47a) to obtain (41). The proof of [9, Prop. 12], using [26] and [25], can be applied straightforwardly. $\square$

### 3.3.1 Deterministic weights simplification

It is computationally costly to compute the function $\psi_d$ in (39) for every $d \in \{0, \ldots, D\}$, initial state $x_d \in \mathbb{X}_d$ and stochastic weights $\boldsymbol{\lambda} \in L^q(\Omega, \mathcal{F}, \mathbb{P}; \Lambda_{d+1})$. As in §3.2.1 we relax the equality target constraint (25b) and restrict the computation to deterministic weights in $\Lambda_{d+1}$ which corresponds to dualize an expectation target constraint as detailed in the sequel.

We build by backward induction deterministic weights value functions:

$$V_{(D+1, \geq, \mathbb{E})} = K \ , \tag{48a}$$

and for all $d \in \{0, \ldots, D\}$, and for all $x_d \in \mathbb{X}_d$,

$$V_{(d,\geq,\mathbb{E})}(x_d) = \sup_{\lambda_{d+1} \in \Lambda_{d+1}} \psi_d(x_d, \lambda_{d+1}) - V^{\star}_{(d+1,\geq,\mathbb{E})}(\lambda_{d+1}) , \tag{48b}$$

$$\lambda_{d+1} \leq 0 . \tag{48c}$$

Relaxing the target equality constraint (25b) into an inequality makes it possible to constrain to non positive weights in (48c).

**Remark 3.13.** *The notation $V_{(d,\geq,\mathbb{E})}$ with the expectation in index comes from the connection with the dualization of the target constraint in the optimization problem $\mathcal{P}_{(d,\geq)}\big[x_d, \boldsymbol{X}_{d+1}\big]$ where the almost sure inequality target constraint is replaced by a constraint in expectation, see (49b). We denote this new optimization problem by $\mathcal{P}_{(d,\geq,\mathbb{E})}\big[x_d, \boldsymbol{X}_{d+1}\big]$:*

$$\mathcal{P}_{(d,\geq,\mathbb{E})}\big[x_d, \boldsymbol{X}_{d+1}\big] \begin{cases} \min_{\boldsymbol{U}_d} & \mathbb{E}\Big[L_d(x_d, \boldsymbol{U}_d, \boldsymbol{W}_d)\Big] , & (49a) \\[2mm] s.t & \mathbb{E}\Big[f_d(x_d, \boldsymbol{U}_d, \boldsymbol{W}_d) - \boldsymbol{X}_{d+1}\Big] \geq 0 , & (49b) \\[2mm] & \boldsymbol{U}_d = (\boldsymbol{U}_{d,0}, \ldots, \boldsymbol{U}_{d,m}, \ldots, \boldsymbol{U}_{d,M}) , & (49c) \\[1mm] & \boldsymbol{W}_d = (\boldsymbol{W}_{d,0}, \ldots, \boldsymbol{W}_{d,m}, \ldots, \boldsymbol{W}_{d,M}) , & (49d) \\[1mm] & \sigma(\boldsymbol{U}_{d,m}) \subset \sigma(\boldsymbol{W}_{d,0:m}) , & (49e) \end{cases}$$

*That kind of simplification is also applied in [8].* ◇

$V_{(d,\geq,\mathbb{E})}$ is the value of a more constrained maximization problem, with unchanged objective, than $\underline{V}_d$. Hence we have the following inequality

$$V_{(d,\geq,\mathbb{E})} \leq \underline{V}_d \leq V_d . \tag{50}$$

### 3.3.2 Statement of the SAWA algorithm with deterministic weights and periodicity classes

To summarize, the algorithm to compute daily value functions approximations for relaxed intraday problems with deterministic weights is presented in Algorithm 2,

23

once again with periodicity classes as introduced in Lemma 3.6.

---

**Algorithm 2:** Two-time scales dynamic programming with weights

**Data:** Periodicity classes $(\mathbb{I}_1, \ldots, \mathbb{I}_{N_p})$
**Result:** Daily value functions approximations $(V_{(d,\geq,\mathbb{E})})_{d=0,\ldots,D+1}$
Initialization: $V_{(D+1,\geq,\mathbb{E})} = K$;
**for** $i = 1, \ldots, N_p$ **do**
    Let $d \in \mathbb{I}_i$;
    **for** $(x_d, \lambda_{d+1}) \in \mathbb{X}_d \times \Lambda_{d+1}$ **do**
        Compute $\psi_d(x_d, \lambda_{d+1})$;
    **end**
**end**
**for** $d = D, D-1 \ldots, 0$ **do**
    **for** $x_d \in \mathbb{X}_d$ **do**
        Solve $V_{(d,\geq,\mathbb{E})}(x_d) = \sup_{\lambda_{d+1} \in \Lambda_{d+1}} \; \psi_d(x_d, \lambda_{d+1}) - V^\star_{(d+1,\geq,\mathbb{E})}(\lambda_{d+1})$;
    **end**
**end**

---

**Remark 3.14.** *The whole interest of the two algortihms 1 and 2 to compute daily value functions approximations is that we can solve intraday problems in parallel, or distribute the resolution of the intraday problems across days. Moreover, we can theoretically apply any stochastic optimization method to solve the intraday problems. Without stagewise independence assumption we may use Stochastic Programming techniques (for example scenario trees) to solve the intraday problems. With the stagewise assumption we may apply Stochatic Dynamic Programming.* $\diamond$

## 3.4 Producing an online policy using the daily value functions

We assume that we have at disposal daily value functions $\{\widetilde{V}_d\}_{d=0,\ldots,D}$ either obtained by the targets algorithm ($\widetilde{V}_d = V_{(d,\geq,\mathbb{X}_{d+1})}$), or by the weights algorithm ($\widetilde{V}_d = V_{(d,\geq,\mathbb{E})}$).

**Proposition 3.15.** *Under Assumption 3.3 and Proposition* (3.4)*, we have the following inequality*

$$V_{(d,\geq,\mathbb{E})} \leq V_d \leq V_{(d,\geq,\mathbb{X}_{d+1})} \; . \tag{51}$$

*Proof.* It is a reminder of Equation (50) and Proposition 3.4. $\qquad\square$

Now, for each given day $d \in \{0, \ldots, D\}$ and given a current state $x_d \in \mathbb{X}_d$, we can use the daily value functions $\widetilde{V}_d$ as daily value functions approximation in order to state a new intraday problem on day $d$ as follows:

$$\min_{\boldsymbol{X}_d, \boldsymbol{U}_d} \mathbb{E}\Big[ L_d(\boldsymbol{X}_d, \boldsymbol{U}_d, \boldsymbol{W}_d) + \widetilde{V}_d\Big( f_d(\boldsymbol{X}_d, \boldsymbol{U}_d, \boldsymbol{W}_d) \Big) \Big] , \tag{52a}$$

$$\text{s.t } \boldsymbol{U}_d = (\boldsymbol{U}_{d,0}, \ldots, \boldsymbol{U}_{d,m}, \ldots, \boldsymbol{U}_{d,M}) , \tag{52b}$$

$$\boldsymbol{W}_d = (\boldsymbol{W}_{d,0}, \ldots, \boldsymbol{W}_{d,m}, \ldots, \boldsymbol{W}_{d,M}) , \tag{52c}$$

$$\sigma(\boldsymbol{U}_{d,m}) \subset \sigma(\boldsymbol{X}_d, \boldsymbol{W}_{d,0:m}) , \tag{52d}$$

$$\boldsymbol{X}_d = x_d . \tag{52e}$$

This problem can be solved by any method that provides an online policy as presented in [6]. The presence of a final cost $\widetilde{V}_d$ ensures that the long term effect on battery health of decisions made every minute is taken into account inside the intraday problem policy.

The simulation of an online policy for a stochastic optimization problem is often made offline as part of the verification process of a stochastic optimal control problem resolution. Here it would be time consuming to produce online policies using the resolution of problem (52) for every day of the horizon in simulation. These policies are more relevant for the real control of the system, hence the resolution of problem (52) can be distributed across days. We present in the next two paragraphs how to simulate two-time scales policies with targets or weights in a smaller amount of time.

### 3.4.1 Simulating a policy using targets

In the case we decomposed the problem using deterministic targets, we eventually solved intraday problems whose values are $\{\phi_{(d,\geq)}\}_{d=0,\ldots,D}$ for every couple of initial state and deterministic target $(x_d, x_{d+1}) \in \mathbb{X}_d \times \mathbb{X}_{d+1}$. In the process, a policy for every intraday problem has been computed. For $d \in \{0, \ldots, D\}$ and all $(x_d, x_{d+1}) \in \mathbb{X}_d \times \mathbb{X}_{d+1}$, we call $\pi^t_{(d,\geq)}(x_d, x_{d+1}) : \mathbb{W}_d \to \mathbb{U}_{d,0} \times, \ldots, \mathbb{U}_{d,M}$ a policy solving $\mathcal{P}_{(d,\geq)}\Big[x_d, x_{d+1}\Big]$ whose value is $\phi_{(d,\geq)}(x_d, x_{d+1})$.

We computed the value $\phi_{(d,\geq)}(x_d, x_{d+1})$ only on $\mathbb{X}_d \times \mathbb{X}_{d+1}$ as we replaced stochastic targets by deterministic ones. Therefore, we can only compute a target decision solving the problem

$$x^t_{d+1} \in \arg\min_{x \in \mathbb{X}_{d+1}} \Big( \phi_{(d,\geq)}(x_d, x) + V_{(d+1,\geq,\mathbb{X}_{d+1})}(x) \Big) . \tag{53a}$$

25

A deterministic target $x_{d+1}^t$ is computed and we apply the corresponding intraday policy $\pi_{(d,\geq)}^t(x_d, x_{d+1}^t)$ to simulate intraday decisions and states drawing a scenario $w_d$ out of $\boldsymbol{W}_d$. The next state $x_{d+1}$ at the beginning of day $d+1$ is then $x_{d+1} = f_d\Big(x_d, \pi_{(d,\geq)}^t(x_d, x_{d+1}^t)(w_d), w_d\Big)$.

### 3.4.2 Simulating a policy using weights

In the case we decomposed the problem using deterministic weights, we eventually solved relaxed intraday problems whose values are $\{\psi_d\}_{d=0,\ldots,D}$ for every couple of initial ate and deterministic weight $(x_d, \lambda_{d+1}) \in \mathbb{X}_d \times \Lambda_{d+1}$. In the process, a policy for every relaxed intraday problems has been computed. For $d \in \{0, \ldots, D\}$ and all $(x_d, \lambda_{d+1}) \in \mathbb{X}_d \times \Lambda_{d+1}$, we call $\pi_d^w(x_d, \lambda_{d+1}) : \mathbb{W}_d \to \mathbb{U}_{d,0} \times, \ldots, \mathbb{U}_{d,M}$ a policy solving the problem whose value is $\psi_d(x_d, \lambda_{d+1})$.

At the beginning of day $d$ in a state $x_d \in \mathbb{X}_d$, we compute a weight $\lambda_{d+1}^w \in \Lambda_{d+1}$ solving the following optimization problem

$$\lambda_{d+1}^w \in \arg\max_{\lambda \in \Lambda_{d+1}} \Big( \psi_d(x_d, \lambda) - V_{(d+1,\geq,\mathbb{E})}^\star(\lambda) \Big) . \tag{54a}$$

Thanks to this deterministic weight $\lambda_{d+1}^w$, we apply the corresponding intraday policy $\pi_d^w(x_d, \lambda_{d+1}^w)$ to simulate intraday decisions and states drawing a scenario $w_d$ out of $\boldsymbol{W}_d$. The next state $x_{d+1}$ at the beginning of day $d+1$ is then $x_{d+1} = f_d\Big(x_d, \pi_d^w(x_d, \lambda_{d+1}^w)(w_d), w_d\Big)$.

In the next section, we present numerical experiments using the targets and weights algorithms.

## 4 Numerical experiments

In this section, we apply the previous theoretical results to three long term battery management problems. First, we describe the realistic data used for the three experiments. Then, we present the first experiment that consists in solving the battery charge/discharge, aging and renewal management problem introduced in Section 2, with targets decomposition. We compare the results with a daily management approach that ignores aging. Finally, we present a battery aging management without renewal, with fixed capacity, over a few days. It makes it possible to apply targets and weights decomposition algorithms and to compare them to a straightforward application of Stochastic Dynamic Programming and Stochastic Dual Dynamic Programming over the whole horizon. Finally, we apply targets decomposition with SDDP for the sizing of a battery taking aging into account.

## 4.1 Experimental setup

We use a realistic instance of the problem: a house with solar panels and battery. The problem presents three sources of randomness, namely, solar panels production, electrical demand and prices of batteries per kWh.

### 4.1.1 Data to model demand and production

We assume that the house is equipped with 12 kW of solar panels. One year scenarios of solar exposure in Zambia with a time step of 1 minute are openly available[1]. Using these solar scenarios, we can generate realistic solar panels production scenarios using Python library PVlib[2]. We display in Figure 3 the distribution of solar panels production every hour as a boxplot.

For the demand data we obtained openly available scenarios from Ausgrid[3]. This is electrical demand data from a customer in kWh with 1 minute time step as well. We display in Figure 4 the hourly distribution of electrical demand.
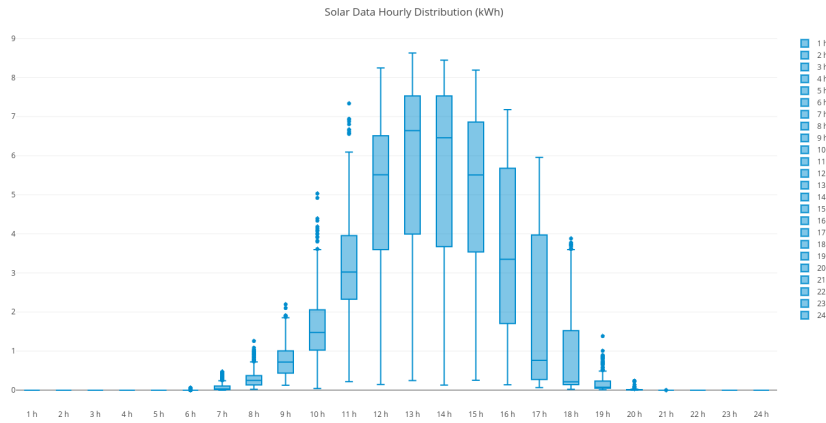


Figure 3: Daily solar panels production hourly distribution (kWh)

---

[1]energydata.info/en/dataset/zambia-solar-radiation-measurement-data-2015-2017
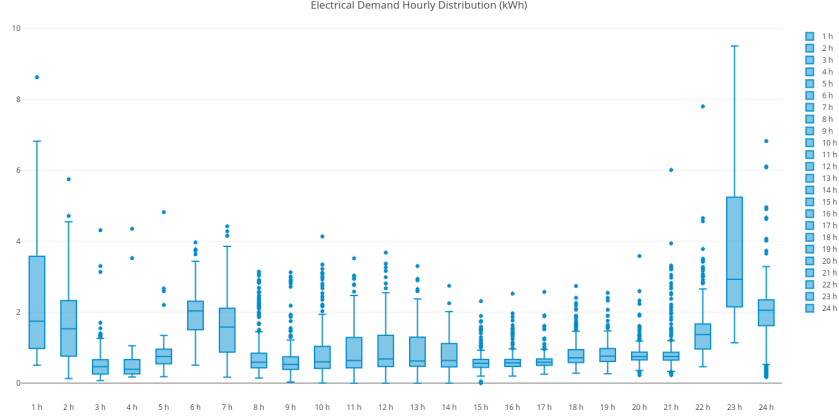[2]github.com/pvlib/pvlib-python
[3]www.ausgrid.com.au/datatoshare

Figure 4: Daily electrical demand hourly distribution (kWh)

### 4.1.2 Data to model the cost of batteries and electricity

For the cost of batteries, we obtained a yearly forecast between 2010 and 2030 from Bloomberg[4]. We added an arbitrary white gaussian noise to generate synthetic random batteries prices scenarios. We display in Figure 5 on the left, the forecast (in blue) and the scenarios we generated (in gray). For the price of electricity, we use a realistic scenario, displayed in Figure 5 on the right, based on EDF blue tariff[5] for a 6 or 9 kVA subscription with peak and off-peak hours.
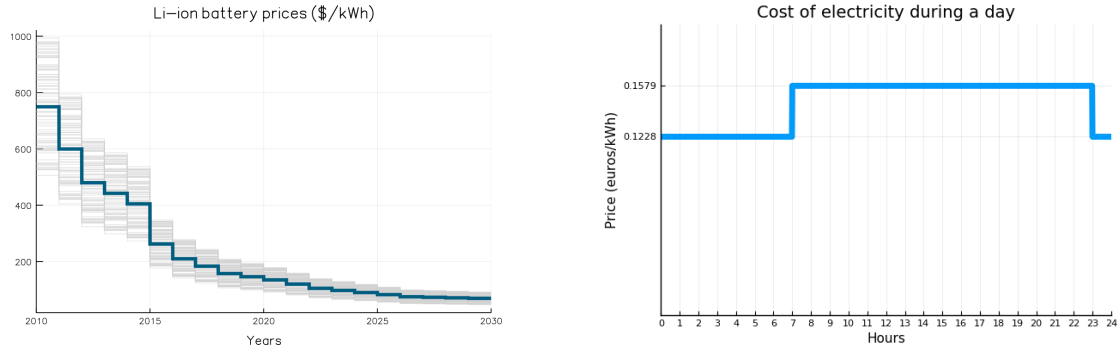


Figure 5: Batteries cost scenarios between 2010 and 2030, and cost of electricity

---

[4]https://www.bloomberg.com/quicktake/batteries
[5]https://particulier.edf.fr/en/home/energy-at-home/electricity/blue-tariff.html

## 4.2 Long term aging and renewal of batteries

We presented the battery aging and renewal problem in Section 2. We recall, with the problem notations, the expression of the daily Bellman value functions for this particular problem:

$$V_d(c_d, b_{d,0}, h_{d,0}) = \min_{\boldsymbol{E}^B_{d,0:M}, \boldsymbol{R}_d} \mathbb{E} \left[ \gamma_d \sum_{m=0}^{M-1} p^e_{d,m} \times \boldsymbol{E}^E_{d,m+1} + \gamma_d \boldsymbol{P}^b_d \times \boldsymbol{R}_d \right.$$

$$\left. + V_{d+1}(\boldsymbol{C}_{d+1}, \boldsymbol{B}_{d+1,0}, \boldsymbol{H}_{d+1,0}) \right] , \tag{55a}$$

$$\text{s.t } (9a), (9b) , \tag{55b}$$

$$(13), (14), (15) , \tag{55c}$$

$$\sigma(\boldsymbol{E}^B_{d,m}) \subset \sigma(\boldsymbol{C}_d, \boldsymbol{B}_d, \boldsymbol{H}_d, \boldsymbol{E}^S_{d,0:m}) , \tag{55d}$$

$$\sigma(\boldsymbol{R}_d) \subset \sigma(\boldsymbol{C}_d, \boldsymbol{B}_d, \boldsymbol{H}_d, \boldsymbol{P}^b_d) , \tag{55e}$$

$$\boldsymbol{C}_d = c_d , \quad \boldsymbol{B}_d = b_{d,0} , \quad \boldsymbol{H}_d = h_{d,0} . \tag{55f}$$

Intuitively, the daily value functions are non-increasing in the state of charge $b$ and the state of health $h$ because it is always preferable to have a full and healthy battery. We prove in Appendix 5.2 that this problem presents all the features required to apply our decomposition algorithms, namely, that the value functions $V_d$ are non-increasing in $b_d$ and $h_d$.

### 4.2.1 Splitting slow and fast decision variables

We introduce the intraday problems for all $d \in \{0, \ldots, D+1\}$:

$$\phi_{(d,\geq)}(c_d, b_{d,0}, h_{d,0}, \boldsymbol{B}, \boldsymbol{H}) = \min_{\boldsymbol{E}^B_{d,0:M}} \mathbb{E} \left[ \sum_{m=0}^{M-1} p^e_{d,m} \times \boldsymbol{E}^E_{d,m+1} \right] , \tag{56a}$$

$$\text{s.t } (9a), (9b) , \tag{56b}$$

$$\boldsymbol{B}_{d,M} \geq \boldsymbol{B} , \quad \boldsymbol{H}_{d,m} \geq \boldsymbol{H} , \tag{56c}$$

$$\sigma(\boldsymbol{E}^B_{d,m}) \subset \sigma(\boldsymbol{C}_d, \boldsymbol{B}_d, \boldsymbol{H}_d, \boldsymbol{E}^S_{d,0:m}) , \tag{56d}$$

$$\boldsymbol{C}_d = c_d , \quad \boldsymbol{B}_d = b_{d,0} , \quad \boldsymbol{H}_d = h_{d,0} . \tag{56e}$$

There is a small difference in the definition of the intraday problem as compared to Equation (25) because we keep the cost $\mathbb{E}\gamma_d \boldsymbol{P}^b_d \times \boldsymbol{R}_d$ outside. It allows to keep the capacity dynamic (13) and its associated target constraint (25b) outside the intraday

problem. All the previous results can still be applied, but this decomposition is less computationally costly. We obtain the following expression for the daily value functions with the deterministic targets simplification:

$$V_{(d,\geq,\mathbb{X}_{d+1})}(c_d, b_{d,0}, h_{d,0}) = \min_{\boldsymbol{R}_d, b, h} \gamma_d \phi_{(d,\geq)}(c_d, b_{d,0}, h_{d,0}, b, h) + \mathbb{E}\left[\gamma_d \boldsymbol{P}_d^b \times \boldsymbol{R}_d\right.$$

$$\left. + V_{d+1}(\boldsymbol{C}_{d+1}, \boldsymbol{B}_{d+1,0}, \boldsymbol{H}_{d+1,0})\right], \tag{57a}$$

$$\text{s.t } (13), (14), (15), \tag{57b}$$

$$\sigma(\boldsymbol{R}_d) \subset \sigma(\boldsymbol{C}_d, \boldsymbol{B}_{d,0}, \boldsymbol{H}_{d,0}, \boldsymbol{P}_d^b), \tag{57c}$$

$$\boldsymbol{C}_d = c_d, \quad \boldsymbol{B}_{d,0} = b_{d,0}, \quad \boldsymbol{H}_{d,0} = h_{d,0}. \tag{57d}$$

It falls down to choose end of the day maximum aging and minimum state of charge at the beginning of the day as well as battery renewal once price of batteries is observed. We recall that, due to the dynamics (13), (14) and (15), the random variables $\boldsymbol{C}_{d+1}$, $\boldsymbol{B}_{d+1,0}$ and $\boldsymbol{H}_{d+1,0}$ depend on $b$ and $h$ in (57).

### 4.2.2 Simplifying the intraday problem

In this application, we are in the situation described in Assumption 3.9 regarding the aging dynamics:

$$\phi_{(d,\geq)}(c_d, b_{d,0}, h_{d,0}, b_{d,M}, h_{d,M}) = \phi_{(d,\geq)}(c_d, b_{d,0}, h_{d,0} - h_{d,M}, b_{d,M}, 0). \tag{58}$$

Finally, as we want to focus on the aging, we neglect the state of charge target replacing it by an empty state of charge target. In fact, we assume that the battery is empty at the end and at the beginning of everyday, which is a pessimistic assumption. We have to compute the following functions:

$$\widetilde{\phi}_{(d,\geq)}(c_d, h_{d,0} - h_{d,M}) = \phi_{(d,\geq)}(c_d, 0, h_{d,0} - h_{d,M}, 0, 0). \tag{59}$$

Then, we can compute the daily value functions approximation $V_{(d,\geq,\mathbb{X}_{d+1})}$ by exhaustive search in discretized daily states, controls and targets spaces. In our numerical experimentation, we use the same uncertainties model everyday day for the noises and we assume that the prices of electricity are the same everyday. Then we compute only $\widetilde{\phi}_{(0,\geq)}$.

To compute the function $\widetilde{\phi}_{(0,\geq)}$, we apply the SDDP algorithm, assume stagewise independence of the noises, for every possible capacities $c_d \in \mathbb{C}$, that is, in our case for all $\mathbb{C} = \{0, 1 \ldots, 19, 20\}$. Applying SDDP for a capacity $c_d$, we obtain a convex polyhedral lower approximation of $\widetilde{\phi}_{(0,\geq)}(c_d, \cdot)$.

### 4.2.3 Numerical simulation of a two-time scales policy

To simulate a policy, we draw one scenario of solar production and electrical demand for every minutes step of the 20 years horizon. We display in Figure 6 left, a net production (solar production minus demand) over one week. We also draw one scenario of cost of batteries for everyday of the horizon in Figure 6 right. We apply the simulation strategy described in §3.4.1.
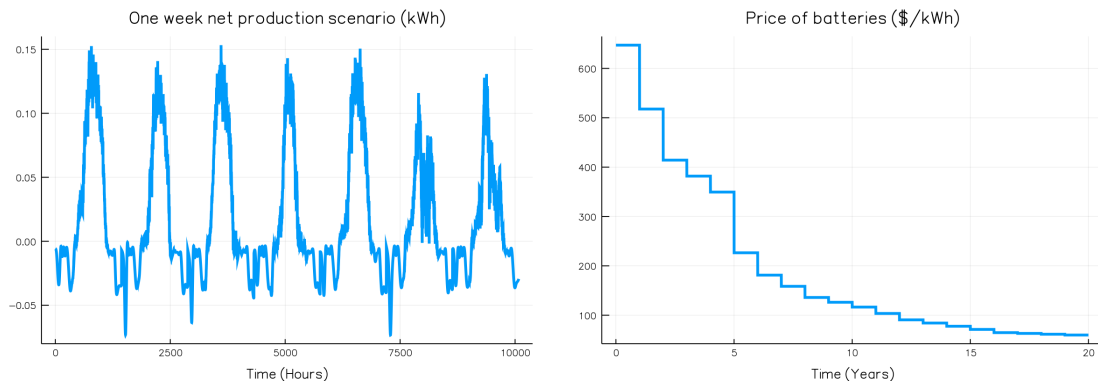


Figure 6: Net production scenario over one week and price of batteries scenario over 20 years

Our instance has then the following numerical features:

- Horizon: 20 years,

- Number of time steps: $10,512,000$ minutes,

- Battery capacity: between 0 and 20 kWh,

- One periodicity class: all days are similar.

### 4.2.4 Comparison of two policies

We compare two policies. One policy is computed using the value functions (57) with the simplification that the state of charge target is restricted to 0, that is, we do not constraint the state of charge of the battery at the end of every day. We call this strategy "aging control" or AC. The other policy is computed using the value functions (57) without state of charge constraints as well. However there is another simplification: the health of the battery at the end of the day has only to remain above 0, that is, there is no health target every day. This policy is called "without

31

aging control" or WAC. We compare hereby the two policies on one twenty years simulation. The reference case is just the electricity bill of the house along the drawn scenario and without battery at all.
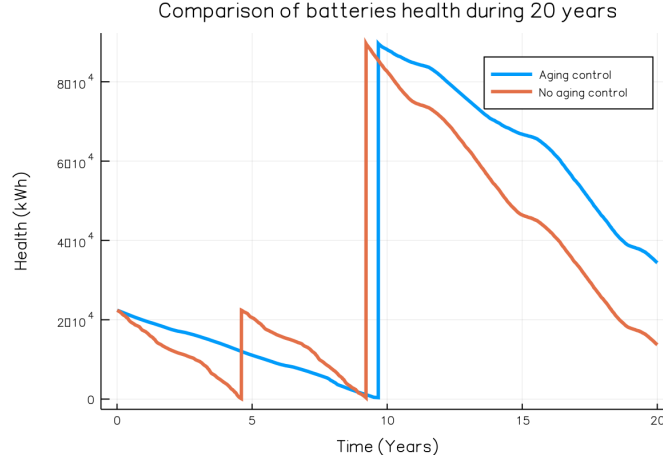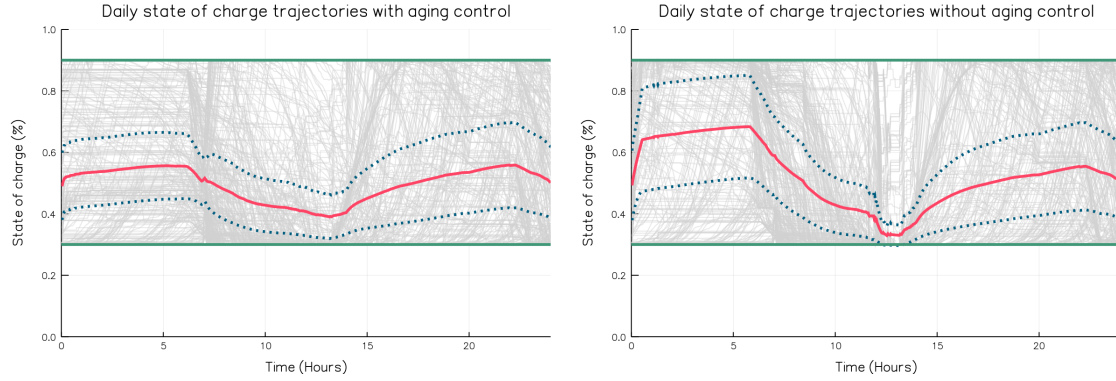


Figure 7: Health decrease comparison



Figure 8: Daily state of charge trajectories comparison (kWh)

We observe unsurprisingly in Figure 7 that the AC strategy renews the battery one less time than the WAC strategy. We observe that both strategies buy the same first capacity, that is a 4 kWh battery. The WAC buys one more 4 kWh battery after about 5 years while the ac ones waits 10 years. At year 10, both strategies buy a 16 kWh battery that has more health at the end of the horizon than with the WAC strategy.

We display in Figure 8 the impact of both strategies on the state of charge of the batteries everyday. In red is the mean state of charge. We observe unsurprisingly that the WAC strategy uses more intensively the battery everyday.

Finally, on this single simulation we obtain the following discounted expenses for electricity bill and battery purchases:

- Reference case (electricity bill without battery and solar panels): 25404 €,

- Without aging control: 24702 €, minus 3 % compared to reference,

- Aging control: 22613 €, minus 11% compared to reference.

On one simulation we observe that the best economical strategy is to buy batteries and to control their aging, against one particular battery scenario. We should make many simulations over multiple battery prices to conclude due to the stochasticity of these battery prices.

**Remark 4.1** (On computation time). *The targets decomposition algorithm is here applied because the problem is way too large to apply SDP straightforwardly (10512000 time steps with 3 state variables). The computation time would be tremendous even if SDP can be parallelized and moreover the memory needed is way above the memory of a regular computer. With the targets decomposition, we display reasonable computation times: the intraday problem $\widetilde{\phi}_{(0,\geq)}$ took around 1 hour to compute, the daily value functions around 45 minutes and a simulation around 45 minutes as well. We present a more rigorous discussion on complexity and computation times in the next experimentation.* ◇

## 4.3 Decomposition methods comparison on a simple aging problem

In this part, we focus on aging of a battery with a given capacity over five days. The objective is to compare targets and weights decomposition algorithms to the results obtained using SDP, and SDDP, straightforwardly over the whole horizon. We assume that the house is equipped with a battery with a given capacity $c_0$ that will not change during the five days. This assumption removes the *slow scale* renewal decision variable from the previous experiment (in §4.2). In this the application of this new paragraph, we demonstrate that the decomposition algorithms are also efficient to solve stochastic optimization problems with many time stages but a single scale

### 4.3.1 Problem instance

We present hereby the different parameters of the problem we solve.

- Horizon: 5 days.

- Time step : 15 minutes.

- Number of time steps : 480.

- Capacity: $c_0 = 13$ kWh battery.

- Initial health: $h_{0,0} = 100$ kWh of exchangeable energy.

### 4.3.2 Target daily value functions without renewal

Without battery renewal, we obtain the following daily target value functions defined by backward induction:

$$V_{(d,\geq,\mathbb{X}_{d+1})}(b_{d,0}, h_{d,0}) = \min_{b,h} \ \phi_{(d,\geq)}(c_0, b_{d,0}, h_{d,0}, b, h) + V_{(d+1,\geq,\mathbb{X}_{d+2})}(b, h) \ , \qquad (60a)$$

where $\phi_{(d,\geq)}$ is the intraday problem defined in (56).

Once again, we decide to neglect state of charge target to focus on health here. We fix state of charge target to zero, or empty battery. We also take the same net production uncertainties model everyday of the five days horizon. As in the previous experimentation, we then compute the single intraday problem $\widetilde{\phi}_{(0,\geq)}(c_0, \cdot)$ with only one capacity this time. In practice, this problem is solved using SDDP provided a polyhedral lower approximation of $\widetilde{\phi}_{(0,\geq)}(c_0, \cdot, \cdot)$. We call the target value functions $V^T_{(d,\geq,\mathbb{X}_{d+1})}$:

$$V^T_{(d,\geq,\mathbb{X}_{d+1})}(h_{d,0}) = \min_{h} \ \widetilde{\phi}_{(d,\geq)}(c_0, b_{d,0}, h_{d,0} - h,) + V^T_{(d+1,\geq,\mathbb{X}_{d+2})}(h) \ . \qquad (61a)$$

As the $\widetilde{\phi}_{(0,\geq)}(c_0, \cdot, \cdot)$ is convex polyhedral, we can solve the backward recursion (61a) using linear programming. Moreover, we can obtain a convex polyhedral lower approximation of $V^T_{(d,\geq,\mathbb{X}_{d+1})}$. In the sequel, when we refer to $V^T_{(d,\geq,\mathbb{X}_{d+1})}$, we refer to this polyhedral approximation.

### 4.3.3 Weights daily value functions without renewal

As we focus only on aging we do not dualize the whole intraday dynamics here. We dualize only the aging dynamic. We define the relaxed intraday problem:

$$\psi_d(c_0, h_{d,0}, \lambda_d) = \tag{62a}$$

$$\min_{\boldsymbol{E}^B_{d,0:M}} \mathbb{E}\left[\sum_{m=0}^{M-1} p^e_{d,m} \boldsymbol{E}^E_{d,m+1}\right] + \mathbb{E}\left[\lambda_d \times \sum_{m=0}^{M-1} \frac{1}{\rho_d} \boldsymbol{E}^{B-}_{d,m} - \rho_c \boldsymbol{E}^{B+}_{d,m}\right], \tag{62b}$$

$$\text{s.t } \boldsymbol{B}_{d,m+1} = \boldsymbol{B}_{d,m} - \frac{1}{\rho_d} \boldsymbol{E}^{B-}_{d,m} + \frac{1}{\rho_d} \rho_c \boldsymbol{E}^{B+}_{d,m}, \tag{62c}$$

$$\boldsymbol{B}_{d,M} \geq \underline{B}, \tag{62d}$$

$$\sigma(\boldsymbol{E}^B_{d,m}) \subset \sigma(\boldsymbol{B}_d, \boldsymbol{H}_d, \boldsymbol{E}^S_{d,0:m}), \tag{62e}$$

$$\boldsymbol{B}_d = \underline{B}, \quad \boldsymbol{H}_d = h_d. \tag{62f}$$

So, as $\lambda_d$ deterministic, the objective turns into

$$\mathbb{E}\left[\sum_{m=0}^{M-1} p^e_{d,m} \boldsymbol{E}^E_{d,m+1} + \frac{\lambda_d}{\rho_d} \boldsymbol{E}^{B-}_{d,m} - \lambda_d \rho_c \boldsymbol{E}^{B+}_{d,m}\right]. \tag{63}$$

It makes it possible to solve the problem (62) using SDDP producing a convex polyhedral approximation of $\psi_d(c_0, \cdot, \lambda_d)$. We make the periodicity assumption as in the targets decomposition.

Then, we compute weights daily value functions solving the backward recursion:

$$V^W_{(d,\geq,\mathbb{E})}(h_{d,0}) = \sup_{\lambda \in \Lambda_{d+1}} \left[\psi_0(c_0, h_{d,0}, \lambda) + \min_{h \in \mathbb{H}} \left(-\lambda \cdot h + V^W_{(d+1,\geq,\mathbb{E})}(h)\right)\right]. \tag{64}$$

In this case we computed many intraday problems value $\psi_0(c_0, \cdot, \lambda)$ for different $\lambda \in [0, 2]$. We need to compute the values $\psi_d$ only for $d = 0$ using the periodicity assumption. It appears that, above $\overline{\lambda} = 0.08$, the mapping $\lambda \to \psi_0(c_0, \cdot, \lambda)$ is constant. Therefore, we perform the maximization (64) by exhaustive search in the discrete space $\{-0.08, -0.0784, \ldots, -0.0016, 0\}$, that is, all the weights between $-0.08$ and $0$ with step $0.0016$, which gives 51 weights. The nested minimization over $h$ in (64) is performed by exhaustive search in the discretized health space as well.

### 4.3.4 Computing minute value functions

Both targets and weights decomposition methods provide daily value functions $V^T_{(d,\geq,\mathbb{X}_{d+1})}$ and $V^W_{(d,\geq,\mathbb{E})}$. Next, we compute *intraday value functions* using these value functions

as final cost in new intraday problems. For example, in the targets case, we compute the family of intraday value functions $V_{d,m}^T$ by applying the SDDP algorithm to the problem, because the final cost $V_{(d+1,\geq,\mathbb{X}_{d+2})}^T$ is convex polyhedral:

$$\min_{\boldsymbol{E}_{d,0:M}^B} \ \mathbb{E}\left[\sum_{m=0}^{M-1} p_{d,m}^e \times \boldsymbol{E}_{d,m+1}^E + V_{(d+1,\geq,\mathbb{X}_{d+2})}^T(\boldsymbol{H}_{d,M})\right], \tag{65a}$$

$$\text{s.t } (9a), (9b), \tag{65b}$$

$$\sigma(\boldsymbol{E}_{d,m}^B) \subset \sigma(\boldsymbol{B}_{d,m}, \boldsymbol{H}_{d,m}, \boldsymbol{E}_{d,m}^S). \tag{65c}$$

For the weights case, we apply SDP as we computed an approximation of $V_{(d+1,\geq,\Lambda_{d+2})}^W$ on a grid.

### 4.3.5 Two straightforward references

We compare these two daily decomposition methods to two straightforward approaches because the problem is not too large to apply them. However we will observe that numerically these classical methods perform poorly to produce the "true" value functions. We apply SDP and SDDP to the following global problem assuming stagewise (minutes) independence of the noises:

$$\min_{\boldsymbol{E}_{d,0:M}^B} \ \mathbb{E}\left[\sum_{d=0}^{D-1}\sum_{m=0}^{M-1} p_{d,m}^e \times \boldsymbol{E}_{d,m+1}^E\right], \tag{66a}$$

$$\text{s.t } (9a), (9b), \tag{66b}$$

$$\boldsymbol{B}_{d+1,0} = \boldsymbol{B}_{d,M}, \quad \boldsymbol{H}_{d+1,0} = \boldsymbol{H}_{d,M}, \tag{66c}$$

$$\sigma(\boldsymbol{E}_{d,m}^B) \subset \sigma(\boldsymbol{B}_{d,m}, \boldsymbol{H}_{d,m}, \boldsymbol{E}_{d,m}^S). \tag{66d}$$

With both algorithms, we obtain a family of intraday value functions respectively called $V_{d,m}^{SDP}$ and $V_{d,m}^{SDDP}$. To be consistent with the two previous methods, at the beginning of each day in a given state $b_{d,0}, h_{d,0}$ we force the value function $V_0^{SDP}$ to satisfy the equality $V_{d,0}^{SDP}(b_{d,0}, h_{d,0}) = V_{d,0}^{SDP}(0, h_{d,0})$ in order to ignore the state of charge at the end and beginning of each day as well. We do the same for SDDP.

### 4.3.6 Numerical results

We now present numerical results comparing the four methods. Figure 9 presents in-sample simulation results with the four algorithms along one scenario. It means that

we have drawn one scenario from the uncertainties distribution we used to compute the daily value functions. We observe that the aging of the battery as well as its state of charge over the 5 days seems to be the same for all algorithms. We observed the same fact on 10,000 scenarios, that are not displayed here.
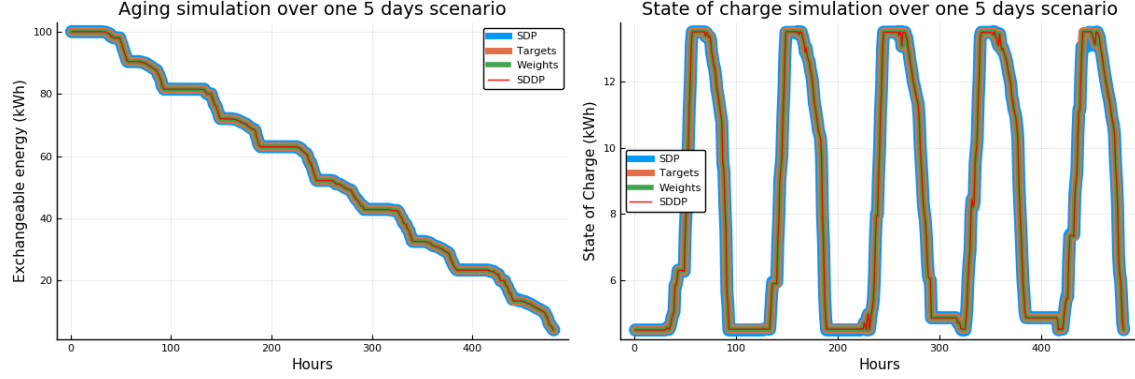


Figure 9: Aging and state of charge simulation over 5 days with different methods

Figure 10 presents the distribution of the difference of costs between each pair of algorithms along 10,000 scenarios. All the histograms are centered around 0, hence it seems that all algorithms perform equivalently with a small win for the weights decomposition over the targets decomposition. The mean difference is approximately zero between all methods.
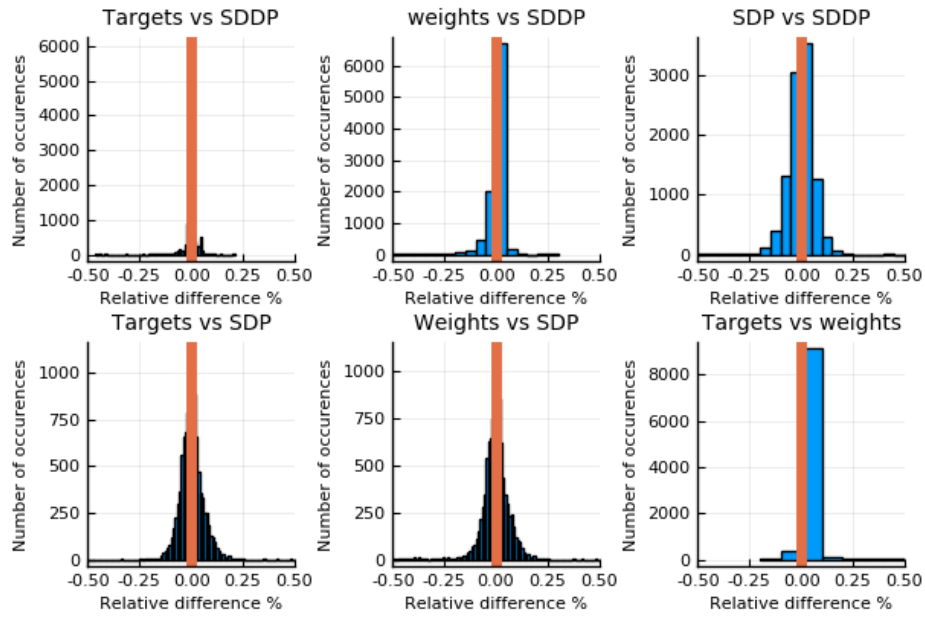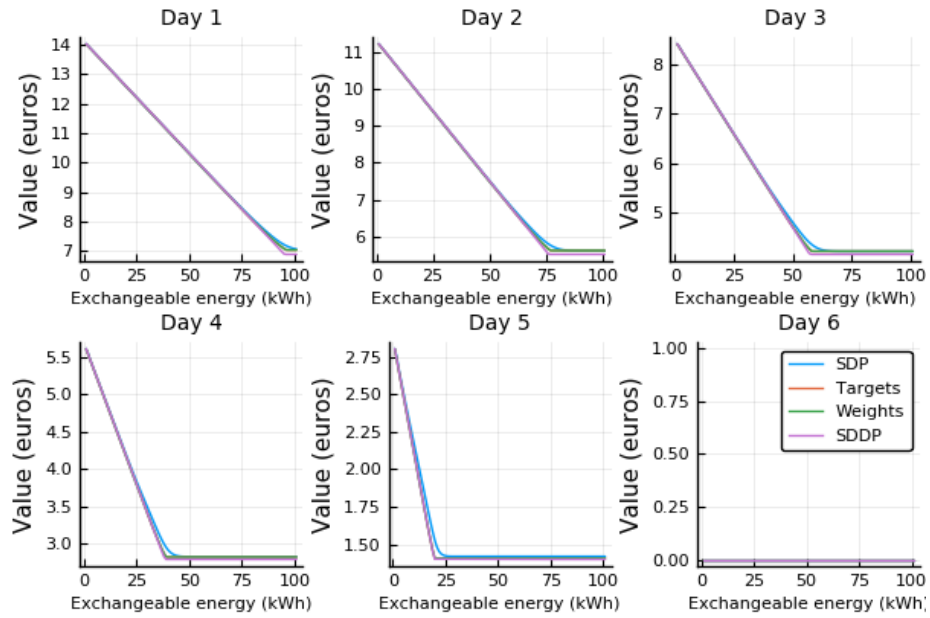
Figure 10: Simulation costs comparison



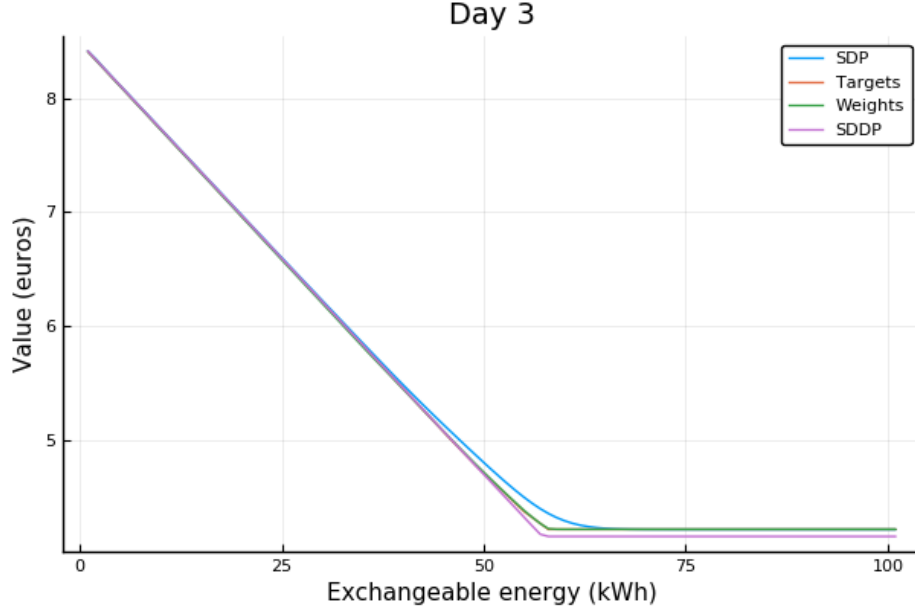Figure 11: Daily value functions comparison

Figure 12: Focus on daily value function on day number 3

Figure 11 displays the six daily value functions, the sixth one being the final cost equal to zero. We observe that all methods compute approximately the same daily value functions. On Figure 12 we focus on day 3 and we observe the following order between value functions: $V_{3,0}^{SDDP} \leq V_{(3,\geq,\mathbb{E})}^{W} \leq V_{(3,\geq,\mathbb{X}_4)}^{T} \leq V_{3,0}^{SDP}$. We observe the same thing on all the days. This is consistent with the fact that SDDP provides a lower approximation of the true value functions, while SDP, because of discretization of state and control spaces, provides an upper approximation. It is surprising to find the value functions computed using targets and weights decomposition between these two bounds even with the deterministic weights and targets simplification. We recall that the "true" value functions $V_d$ satisfy the inequality:

$$V_{(d,\geq,\mathbb{E})}^{W} \leq V_d \leq V_{(d,\geq,\mathbb{X}_{d+1})}^{T} \; . \tag{67}$$

|  | SDP | Weights | SDDP | Targets |
|---|---|---|---|---|
| Intraday resolution (SDDP) | $\emptyset$ | $51 \times 14$ sec | $\emptyset$ | 14 sec |
| Daily values functions | $\emptyset$ | 0.15 sec | $\emptyset$ | 0.59 sec |
| Minute values functions | 22.5 min | $5 \times 4.5$ min | 3.6 min | $5 \times 14$ sec |
| Total CPU time | 22.5 min | 34.4 min | 3.6 min | 84.6 sec |
| Total time (with parallelisation) | 22.5 min | 5.0 min | 3.6 min | 24.6 sec |
| Gap ($200 \times \frac{mc-v}{mc+v}$) | 0.91 % | 0.32 % | 0.90 % | 0.28 % |

Table 1: Algorithms numerical results comparison

Table 4.3.6 presents the computation times of the algorithms as well as their gap which is measured as the relative difference between the initial value $V_{0,0}(0, 100)$ computed by the algorithm and the mean cost obtained by simulation over 10000 scenarios from state $(0, 100)$. SDP and SDDP do not display intraday resolution and daily value functions times as they are applied directly to the global problem, computing both daily and minute value functions.

- We observe that daily value functions computation for targets and weights algorithms is really fast, but that the intraday problems resolution for weights is costly. This is due to the exhaustive search in the weights space (of cardinal 51 here). This time is significantly lower in the targets case because SDDP already explores the initial state space.

- We observe that targets and weights algorithms have the best gaps. We could improve the one of SDDP but we did not manage to improve it significantly after more than 1 hour. The convergence of SDDP (measured with the gap) is sensitive to the number of time stages [18].

- The time required to compute value functions in the weights case is the same as SDP, as it is 5 times a 5 time smaller SDP. However the weights algorithm permits to parallelize this phase or even to distribute it accross days, witch is impossible with SDP.

Finally, we observe that the targets algorithm, where all the intraday problems are solved using SDDP is the fastest algorithm. It has the best gap, displays approximately the same costs and value functions as the other algorithms. Moreover, the resolution of the intraday problems with final cost can be parallelized or distributed accross days. In fact it is a way to accelerate the resolution of the problem with multiple applications of SDDP, instead of one straighforward application, when a convex problem displays a high number of time steps, monotonicity and some good

properties. That kind of approach could be appealing for the algorithm Mixed Integer Dynamic Approximation Scheme [22] as it is really sensible to the number of time steps and relies on monotonicity as well.

## 4.4 Sizing of a battery using targets decomposition and Stochastic Dual Dynamic Programming

In this part, we use the lower convex polyhedral approximations of the intraday target problems values $\phi_{(d,\geq)}$ provided by SDDP, to compute an optimal capacity for the house.

### 4.4.1 A sizing problem without battery renewal

We apply the targets decomposition scheme, introduced in §3.2 to the same aging problem without battery renewals to compute the net present value for a given battery capacity. We introduce the intraday problems:

$$\phi_{(d,\geq)}(c_d, h_{d,0} - h_{d,M}) = \min_{\boldsymbol{E}^B_{d,0:M}} \ \mathbb{E}\Big[ \sum_{m=0}^{M-1} p^e_{d,m} \times \boldsymbol{E}^E_{d,m+1} \Big] , \tag{68a}$$

$$\text{s.t } (9a), (9b) , \tag{68b}$$

$$\boldsymbol{B}_{d,M} \geq \underline{B} , \ \ \boldsymbol{H}_{d,m} \geq 0 , \tag{68c}$$

$$\boldsymbol{C}_{d,m} = \boldsymbol{C}_{d,m+1} , \tag{68d}$$

$$\sigma(\boldsymbol{E}^B_{d,m}) \subset \sigma(\boldsymbol{C}_d, \boldsymbol{B}_{d,0}, \boldsymbol{H}_{d,0}, \boldsymbol{E}^S_{d,0:m}) , \tag{68e}$$

$$\boldsymbol{C}_{d,0} = c_d , \ \ \boldsymbol{B}_{d,0} = \underline{B} , \ \ \boldsymbol{H}_{d,0} = h_{d,0} - h_{d,M} . \tag{68f}$$

The intraday problem remains unchanged compared to the previous experiment 4.2 except that we augment the state with a constant dynamic for the capacity, see (68d). We recover the value $\phi_{(d,\geq)}$ as we can eliminate trivially the capacity state using (68d). This trick makes it possible to compute a lower convex polyhedral approximation $\widetilde{\phi}_{(d,\geq)}$ by applying the SDDP algorithm with a grid of initial states.

It is then possible to apply SDDP to the daily target value functions recursion without battery renewal:

$$V_{(d,\geq,\mathbb{X}_{d+1})}(c_d, h_{d,0}) = \min_h \ \gamma_d \phi_{(d,\geq)}(c_d, h_{d,0} - h) + V_{(d+1,\geq,\mathbb{X}_{d+2})}(c_d, h_{d,0} - h) . \tag{69a}$$

The application of a two-time scale SDDP is relevant only if daily value functions are required. Here, only the sizing of the battery matters; hence we can just solve a linear program to compute the optimal capacity as presented in the next section.

We worked on an industrial case where the economic profit was not the only objective. We had to determine also the rate of self consumption and self production for a given capacity. For this purpose, we needed to compute daily value functions so as to produce a simulation policy to evaluate these rates using Monte Carlo simulation.

### 4.4.2 Computation of the optimal capacity

We obtained a convex polyhedral approximation of $\phi_{(d,\geq)}$ for every relevant $d \in \{0, \ldots, D\}$ possibly using periodicity. We can then compute an optimal capacity at the first day 0 by solving the following linear program for a given price of batteries per kWh $p_0^b$:

$$\min_{c \in [\underline{c}, \bar{c}]} \min_{h_{0,0}, \ldots, h_{D+1,0}} \quad p_0^b \times c + \sum_{d=0}^{D} \gamma_d \phi_{(d,\geq)}(c, h_{d,0} - h_{d+1,0}) , \tag{70a}$$

$$\text{s.t } h_{0,0} = 2N_{cycles} \times c , \tag{70b}$$

$$h_{d,0} \geq h_{d+1,0} . \tag{70c}$$

This is a linear program using the convex lower polyhedral approximation of $\phi_{(d,\geq)}$. The constraint (70c) makes it possible to lower the size of the search space but is implicit as a negative initial age $h_{d,0} - h_{d+1,0}$ leads to $\phi_{(d,\geq)}(c, h_{d,0} - h_{d+1,0}) = +\infty$.

### 4.4.3 Numerical results

We compute the value of Problem (68) using SDDP with 500 iterations in 1.7 minute. We test convergence by MonteCarlo simulation and we reach a gap of 0.1% between the upper bound, computed by Monte Carlo, and the lower bound, computed by SDDP. Then, we were able to solve Problem (70) for a given horizon and price of batteries in 1.7 second for a 1 year horizon and 71 seconds for 12 years horizon using CPLEX. We present numerical results as contour plots on Figure 13 and 14. The first one presents the optimal battery capacity (black is 0 kWh, yellow is 20 kWh) as a function of the investment horizon $D$ and the prices of batteries $p_0^b$. The second presents the corresponding discounted benefit (black is 0 €, yellow is above 2100 €) as a function of horizon and prices as well. The third one presents the corresponding expected lifetime for the battery (black is 0 years, yellow is 7 years), as a function of horizon and prices. We observe that a battery would be economically interesting if we consider the investment over at least 2 years and below 150 euros per kWh, below the optimal battery has capacity 0 kWh. Over a 12 years horizon, the best

capacity is the largest one, that is 20 kWh capacity, and the expected net benefit would be 2100 euros. We finally observe on Figure 14 that even the largest battery is not expected to last more than 7 years. This is consistent with the plateau we observe on Figure 13 above 7 years.
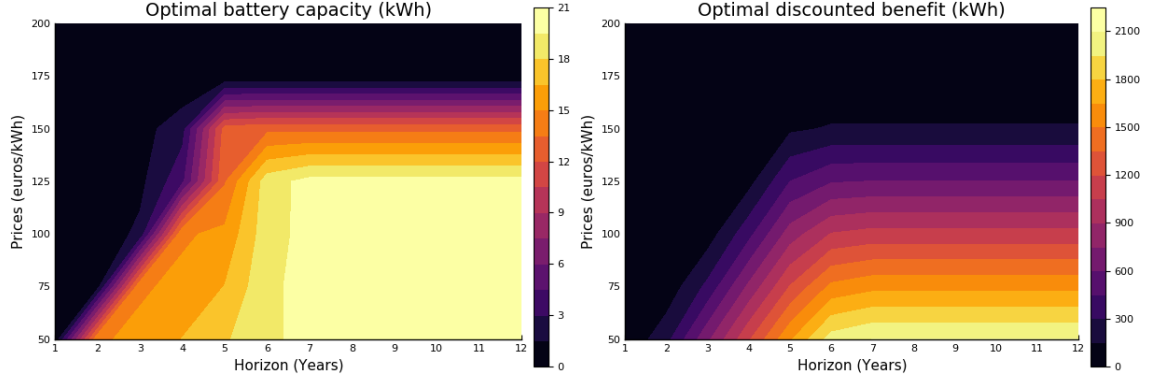


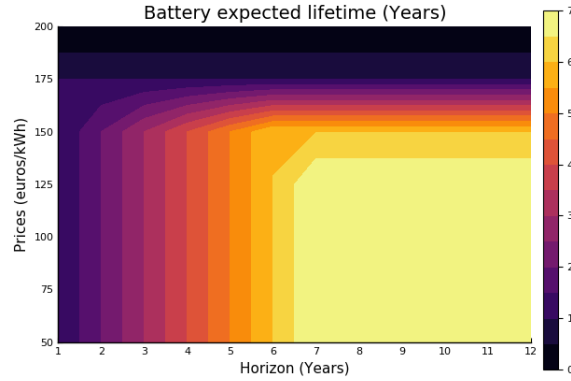Figure 13: Optimal battery capacity and benefit as a function of prices and horizon



Figure 14: Expected battery lifetime as a function of prices and horizon

## Conclusion

We introduced a two-time scales stochastic optimization problem for a battery charge/discharge, aging and renewal management problem. The motivation for two-time scales modeling originated from the existence of decisions that have to be made every minute (charge/dicharge) and decisions that have to be made once a day (renewal). We presented two algorithmic methods to compute daily value functions to solve these

problems with an important number of time steps and decisions on different time scales. We applied these algorithms to three realistic applications for the original problem, a simple aging problem and a sizing problem. We conclude that these methods make it possible to solve problems with different time scales as well as an important number of time steps. Moreover, with the aging application (in §4.3) we observe that these methods make it possible to decompose a single time scale long problem in time. Our two algorithms perform better on a long problem displaying periodicity than a straightforward use of SDP or SDDP. It might generally be useful in order to improve algorithms whose convergence is sensitive to the number of time steps. It makes it possible for example to parallelize SDP over time, not states, and to speed up the convergence of SDDP.

Algorithms such as Mixed Integer Dynamic Approximation Scheme produce $T \times \epsilon$ optimal solutions for multistage stochastic optimization problems with binary variables, where $T$ is the number of stages and $\epsilon$ a user tuned parameter. It might be interesting to apply such time decomposition methods to improve the quality of the solution and to speed up the convergence of these algorithms.

Theses methods are to be compared to value iteration and policy iteration applied to infinite horizon problems. Value iteration and policy iteration require both a stationary assumption. Our methods relax this assumption, proving efficient for problems displaying periodicity and monotonicity.

Finally, these methods could make it possible to mix Stochastic Programming and Stochastic Dynamic Programming methods. Stochastic Programming and scenario decomposition methods display a complexity exponential in the number of time steps. Targets and weights decomposition could be a way to make these methods more tractable on problems with an important number of time steps by splitting into subproblems with less time steps.

# 5 Appendix

In this section, we present theoretical results to decompose two-time scales stochastic optimization problems as well as proofs of monotonicity and convexity of a battery management problem.

## 5.1 An abstract optimization problem

Let $(n_v, n_u) \in \mathbb{N}^* \times \mathbb{N}^*$ be given and two subsets $\mathbb{V} \subset \mathbb{R}^{n_v}$ and $\mathbb{U} \subset \mathbb{R}^{n_u}$ equipped with the element-wise partial order $\leq$. Let two proper extended real valued functions

$l: \mathbb{U} \to (-\infty, +\infty]$ and $V: \mathbb{V} \to (-\infty, +\infty]$ and a mapping $f: \mathbb{U} \to \mathbb{V}$. We study different ways to solve or approximate the following optimization problem:

$$v = \inf_{u \in \mathcal{U}} \Big( l(u) + V(f(u)) \Big) \tag{71}$$

where $\mathcal{U}$ is a subset of $\mathbb{U}$. We call equation (71) an *abstract Bellman equation*.

**Remark 5.1.** *We study this general equation for its application in two-time scales dynamic programming. We assume that a decision maker has to make one decision every minute. The function l represents a cost incurred daily by the decision made every minute. V models a cost of the future incurred by the decision made every minute that change a state through a dynamic equation f. The decision maker wants to minimize a compromise between these two costs.* ◇

In the whole section, we make a monotonicity assumption to decompose the optimization problem (71).

**Assumption 5.2.** *V is a non decreasing function.*

### 5.1.1 Decomposition by targets

For all $\alpha \in \mathbb{V}$ we introduce the following parametrized problems:

$$L_{(=)}(\alpha) = \inf_{u \in \mathcal{U} \cap f^{=\alpha}} l(u) \quad \text{and} \quad L_{(\leq)}(\alpha) = \inf_{u \in \mathcal{U} \cap f^{\leq \alpha}} l(u) , \tag{72}$$

where the level sets $f^{=\alpha}$ and $f^{\leq \alpha}$ are respectively given by

$$f^{=\alpha} = \{ u \in \mathbb{U} | f(u) = \alpha \} \tag{73}$$

and

$$f^{\leq \alpha} = \{ u \in \mathbb{U} | f(u) \leq \alpha \} . \tag{74}$$

In the next lemma, we use the value functions $L_{(=)}$ and $L_{\leq}$ to obtain lower bounds to the optimization problem (71).

**Lemma 5.3.** *The value $v_{(=)}$ and $v_{\leq}$ of the two optimization problems*

$$v_{(=)} = \inf_{\alpha \in \mathbb{V}} \Big( L_{(=)}(\alpha) + V(\alpha) \Big) , \tag{75}$$

$$v_{(\leq)} = \inf_{\alpha \in \mathbb{V}} \Big( L_{(\leq)}(\alpha) + V(\alpha) \Big) , \tag{76}$$

45

give lower bounds to the optimization Problem (71):

$$v_{(\leq)} \leq v_{(=)} = v \ . \tag{77}$$

Moreover, under the monotonicity Assumption 5.2 we have that

$$v_{(\leq)} = v_{(=)} = v \ . \tag{78}$$

*Proof.* $(v_{(=)} = v)$:

$$\begin{aligned}
v_{(=)} &= \inf_{\alpha \in \mathbb{V}} \left( L_{(=)}(\alpha) + V(\alpha) \right) \\
&= \inf_{\alpha \in \mathbb{V}} \inf_{u \in \mathcal{U}} \ l(u) + V(\alpha) \\
&\qquad \text{s.t } f(u) = \alpha \\
&= \inf_{u \in \mathcal{U}} \inf_{\alpha \in \mathbb{V}} \ l(u) + V(\alpha) \\
&\qquad \text{s.t } f(u) = \alpha \\
&= \inf_{u \in \mathcal{U}} l(u) + V(f(u)) = v
\end{aligned}$$

$(v_{(\leq)} \leq v)$: Let $u \in \mathcal{U}$ be given and set $\alpha = f(u)$. We successively have

$$\begin{aligned}
v_{(\leq)} &\leq L_{(\leq)}(\alpha) + V(\alpha) &\text{(by (76))} \\
&\leq l(u) + V(\alpha) &\text{($u$ is admissible for $L_{(\leq)}(\alpha)$)} \\
&\leq l(u) + V(f(u)) &\text{($\alpha = f(u)$)}
\end{aligned}$$

so finally, as the inequality holds for any $u \in \mathcal{U}$:

$$v_{(\leq)} \leq \inf_{u \in \mathcal{U}} \left( l(u) + V(f(u)) \right) = v \qquad\qquad \text{(by (71))}$$

$(v \leq v_{\leq}$ under monotonicity assumption). For any $\epsilon > 0$ let $\alpha_\epsilon \in \mathbb{V}$ be a $\epsilon$-optimal solution for the optimization problem $v_{(\leq)}$ and $u_\epsilon \in \mathbb{U}$ be an $\epsilon$-optimal solution for the optimization problem $L_{(\leq)}(\alpha_\epsilon)$.

$$\begin{aligned}
v_{(\leq)} + 2\epsilon &\geq l(u_\epsilon) + V(\alpha_\epsilon) \\
&\geq l(u_\epsilon) + V(f(u_\epsilon)), \ \ (\text{ monotonicity of } V \text{ and admissibility of } u_\epsilon \text{ for } L_{(\leq)}) \\
&\geq v
\end{aligned}$$

The proof is complete. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

**Remark 5.4.** *The same relation holds between $v_{(=)}$ and $v_{(\geq)}$ in the converse case where $V$ is non-increasing.* $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\diamondsuit$

### 5.1.2 Decomposition by weights using Fenchel duality

Let $(\Lambda, \leq)$ be a subset of $\mathbb{R}^{n_\lambda}$ equipped with the element-wise partial order $\leq$ and $\langle \cdot, \cdot \rangle : \Lambda \times \mathbb{V} \to [-\infty, +\infty]$ a bilinear coupling. For all $\lambda \in \Lambda$ we introduce the following relaxed version of $L_=$:

$$H(\lambda) = \inf_{u \in \mathcal{U}} \quad l(u) + \langle \lambda, f(u) \rangle . \tag{79}$$

We introduce as well the Fenchel conjugate of a function $\phi$.

$$\phi^*(\lambda) = \sup_{\alpha \in \mathbb{V}} \quad \langle \lambda, \alpha \rangle - \phi(\alpha) . \tag{80}$$

**Lemma 5.5.** *The following equality holds*

$$H(\lambda) = -L_=^*(-\lambda) . \tag{81}$$

*Proof.* For any function $\phi$ we have $-\phi^*(-\lambda) = \inf_{\alpha \in \mathbb{V}} \quad \phi(\alpha) + \langle \lambda, \alpha \rangle$.

$$\begin{aligned}
H(\lambda) &= \inf_{u \in \mathcal{U}} \; l(u) + \langle \lambda, f(u) \rangle , \\
&= \inf_{u \in \mathcal{U}} \inf_{\alpha \in \mathbb{V}} \; l(u) + \langle \lambda, \alpha \rangle , \\
&\qquad \text{s.t } \alpha = f(u) , \\
&= \inf_{\alpha \in \mathbb{V}} \inf_{u \in \mathcal{U}} \; l(u) + \langle \lambda, \alpha \rangle , \\
&\qquad \text{s.t } \alpha = f(u) , \\
&= \inf_{\alpha \in \mathbb{V}} \langle \lambda, \alpha \rangle + L_=(\alpha) = -L_=^*(-\lambda) .
\end{aligned}$$

$\square$

This makes it possible to apply a weak duality theorem to our original problem (71). Without further assumptions we can state the following lemma.

**Lemma 5.6.**

$$v \geq \sup_{\lambda \in \Lambda} \quad H(\lambda) - V^*(\lambda) . \tag{82}$$

*Proof.* We recall that $v = \inf_{\alpha \in \mathbb{V}} \; L_=(\alpha) + V(\alpha)$. $l$ is proper so $L_=$ is proper as well. Applying twice Fenchel-Young inequality [4] we know, that for all $(\lambda, \alpha) \in \Lambda \times \mathbb{V}$,

- $L_=(\alpha) \geq -L_=^*(-\lambda) + \langle -\lambda, \alpha \rangle$,

- $V(\alpha) \geq -V^*(\lambda) + \langle \lambda, \alpha \rangle$.

47

Therefore, summing the inequalities, we obtain:

$$L_=(\alpha) + V(\alpha) \geq -L_=^*(-\lambda) - V^*(\lambda) = H(\lambda) - V^*(\lambda) \ . \tag{83}$$

This ends the proof. □

**Proposition 5.7.** *If $L_=$ and $V$ are convex and one of the following condition holds*

- $0 \in ri(\mathrm{dom}(L_=) - \mathrm{dom}(V))$,

- *or the stronger* $\mathrm{dom}(L_=) \cap cont(V) \neq \emptyset$,

*then the following equality holds:*

$$v = \sup_{\lambda \in \Lambda} \quad H(\lambda) - V^*(\lambda) \ . \tag{84}$$

*Proof.* We apply Fenchel duality theorem [4, 27]. □

**Proposition 5.8.** *Let $\lambda \in \Lambda$ such that the function $\langle \lambda, \cdot \rangle : \alpha \in \mathbb{V} \mapsto \langle \lambda, \alpha \rangle$ is non-decreasing. Then the following equality holds:*

$$H(\lambda) = -L_{\leq}^*(-\lambda) \ . \tag{85}$$

*Proof.* It is a direct application of Lemma 5.3 with $V = \langle \lambda, \cdot \rangle$. □

## 5.2 Proving monotonicity and linearity of a battery management problem

We show in this part that we can linearize a battery control problem with aging, which is useful to apply Model Predictive Control or SDDP.

We focus on the following problem where the decision variable $\boldsymbol{U}_t$ is the charge discharge of the battery at time $t$ and $\boldsymbol{W}_t$ the uncertain net production of the grid connected to the battery. The objective is to minimize to consumption of power of the national grid, that is, the charge/discharge minus the net production. We take the positive part assuming that we cannot sell electricity to the grid. We call $x^+ = \max(0, x)$ the positive part of a variable $x$ and $x^- = -\min(0, x)$ the negative part. We write the problem in an hazard-decision setting for the sake of simplicity; the results are the same in a decision-hazard setting. For the sake of simplicity as well, we assume that the noises $(\boldsymbol{W}_0, \ldots, \boldsymbol{W}_T)$ are stagewise independent. It makes it possible to restrict the search to functions $\boldsymbol{U}_{t+1}(\boldsymbol{B}_t, \boldsymbol{H}_t, \boldsymbol{W}_{t+1})$ of the state and the next noise. The problem we study is the following:

$$\inf_{\boldsymbol{U}_{t=0,\dots,T-1}} \mathbb{E}\left[\sum_{t=0}^{T-1} c_t \times \left(\boldsymbol{U}_{t+1} - \boldsymbol{W}_{t+1}\right)\right]^+ , \tag{86a}$$

$$\text{s.t } \boldsymbol{B}_{t+1} = \boldsymbol{B}_t + \rho_c \boldsymbol{U}_{t+1}^+ - \rho_d^{-1}\boldsymbol{U}_{t+1}^- , \tag{86b}$$

$$\boldsymbol{H}_{t+1} = \boldsymbol{H}_t - \boldsymbol{U}_{t+1}^+ - \boldsymbol{U}_{t+1}^- , \tag{86c}$$

$$\underline{B} \leq \boldsymbol{B}_t \leq \overline{B} , \tag{86d}$$

$$\boldsymbol{H}_t \geq 0 , \tag{86e}$$

$$\underline{U} \leq \boldsymbol{U}_t \leq \overline{U} , \tag{86f}$$

$$\sigma(\boldsymbol{U}_t) \subset \sigma(\boldsymbol{B}_{t-1}, \boldsymbol{H}_{t-1}, \boldsymbol{W}_t) , \tag{86g}$$

$$\boldsymbol{B}_0 = b_0 , \quad \boldsymbol{H}_0 = h_0 . \tag{86h}$$

First, we prove that the value functions $V_t^\sharp$ of problem (86) are non increasing in state of charge $b$ and health $h$. The value functions satisfy the following backward recursion

$$V_T^\sharp = 0 , \tag{87a}$$

$$V_t^\sharp(b, h) = \mathbb{E}V_t(b, h, \boldsymbol{W}), \ \forall (b, h) \in \mathbb{B} \times \mathbb{H} , \tag{87b}$$

where

$$V_t(b, h, w) = \inf_u c_t \times (u - w)^+ + V_{t+1}^\sharp(b + \rho_c u^+ - \rho_d^{-1}u^-, h - u^+ - u^-) , \tag{87c}$$

$$\text{s.t } \underline{B} - b \leq \rho_c u^+ - \rho_d^{-1}u^- \leq \overline{B} - b , \tag{87d}$$

$$u^+ + u^- \leq h , \tag{87e}$$

$$\underline{U} \leq u \leq \overline{U} . \tag{87f}$$

**Lemma 5.9.** *The value functions $\{V_t^\sharp\}_{t=0,\dots,T}$ are non-increasing.*

*Proof.* The last value function $V_T^\sharp$ is obviously non-increasing.

Assume that $V_{t+1}^\sharp$ is non increasing. $V_t$ is obviously non increasing in $h$ as decreasing $h$ constrains the problem further and increases the objective as $V_{t+1}^\sharp$ is non increasing.

Let $b' \geq b$, let $\epsilon > 0$ and let $u_b^\epsilon$ an $\epsilon$-optimal for $V_t(b, h, w)$. By definition, we have

$$c_t \times (u_b^\epsilon - w)^+ + V_{t+1}^\sharp(b + \rho_c u_b^{\epsilon+} - \rho_d^{-1}u_b^{\epsilon-}, h - u_b^{\epsilon+} - u_b^{\epsilon-}) \leq V_t(b, h, w) + \epsilon . \tag{88a}$$

We distinguish two cases.

$u_b^\epsilon \leq 0$: then $u_b^\epsilon$ is admissible for $V_t(b', h, w)$ because

$$\underline{B} - b' \leq \underline{B} - b \leq \rho_c u_b^{\epsilon+} - \rho_d^{-1} u_b^{\epsilon-} = \rho_d^{-1} u_b^\epsilon \leq 0 \leq \overline{B} - b' , \qquad (88b)$$

moreover as $V_{t+1}$ is non increasing

$$V_{t+1}^\sharp(b' + \rho_c u_b^{\epsilon+} - \rho_d^{-1} u_b^{\epsilon-}, h - u_b^{\epsilon+} - u_b^{\epsilon-}) \leq V_{t+1}^\sharp(b + \rho_c u_b^{\epsilon+} - \rho_d^{-1} u_b^{\epsilon-}, h - u_b^{\epsilon+} - u_b^{\epsilon-}) , \qquad (88c)$$

then we have

$$V_t(b', h, w) \leq V_t(b, h, w) + \epsilon . \qquad (88d)$$

$u_b^\epsilon > 0$: let $u_{b'}^\epsilon = \min(\rho_c^{-1} \times (\overline{B} - b'), u_b^\epsilon)$. $u_{b'}^\epsilon$ is admissible for $V_t(b', h, w)$ as

$$\underline{U} \leq 0 < u_{b'}^\epsilon \leq u_b^\epsilon \leq \overline{U} , \qquad (88e)$$
$$\underline{B} - b' \leq 0 < \rho_c u_{b'}^\epsilon \leq \rho_c \rho_c^{-1} \times (\overline{B} - b') = B - b' , \qquad (88f)$$
$$u_{b'}^{\epsilon+} \leq u_b^{\epsilon+} \leq h . \qquad (88g)$$

Moreover we have

$$b' + \rho_c u_{b'}^\epsilon = b + \rho_c u_b^\epsilon , \qquad (88h)$$

or

$$b' + \rho_c u_{b'}^\epsilon = \overline{B} \geq b + \rho_c u_b^\epsilon , \qquad (88i)$$

so

$$b' + \rho_c u_{b'}^\epsilon \geq b + \rho_c u_b^\epsilon , \qquad (88j)$$

so these inequalities and the fact that $V_{t+1}$ in non increasing lead to

$$c_t \times (u_{b'}^\epsilon - w)^+ + V_{t+1}^\sharp(b + \rho_c u_{b'}^\epsilon, h - u_{b'}^\epsilon) \leq c_t \times (u_b^\epsilon - w)^+ + V_{t+1}^\sharp(b + \rho_c u_b^\epsilon, h - u_b^\epsilon) \qquad (88k)$$

Finally

$$V_t(b', h, w) \leq V_t(b, h, w) + \epsilon , \qquad (88l)$$

then we conclude that $V_t$ is non increasing.

$\square$

Now we would like to remove the positive and negative parts from the problem to apply SDDP or linear programming in a Model Predictive Control method.

**Lemma 5.10.** *The value functions $\{V_t^\sharp\}_{t=0,...,T}$ are convex polyhedral.*

*Proof.* $V_T^\sharp$ is trivially convex polyhedral. Assume that $V_{t+1}^\sharp$ is convex polyhedral. Let $\mathcal{B} : V_{t+1}^\sharp \mapsto V_t^\sharp = \mathbb{E}V_t$. We prove that this is a linear Bellman operator as it is demonstrated in [18] from [4] that in this case $V_t^\sharp$ is convex polyhedral. We introduce a new equivalent definition for $V_t$:

$$V_t(b, h, w) = \inf_{u_c, u_d, l} l + V_{t+1}^\sharp(b + \rho_c u_c - \rho_d^{-1} u_d, h - u_c - u_d) , \tag{89a}$$

$$\text{s.t } \underline{B} - b \leq \rho_c u_c - \rho_d^{-1} u_d \leq \overline{B} - b , \tag{89b}$$

$$u_c + u_d \leq h , \tag{89c}$$

$$0 \leq u_c \leq \overline{U}, \quad 0 \leq u_d \leq -\underline{U} , \tag{89d}$$

$$u_c \times u_d = 0 , \tag{89e}$$

$$l \geq 0 , \tag{89f}$$

$$l \geq c_t \times (u_c - u_d - w)^+ . \tag{89g}$$

In this new formulation, we introduce three non negative control variables $l, u_c, u_d$. The first non negative one $l$ is used to linearize the objective by adding the constraint (89g). This is a classical trick. The other two $(u_c, u_d)$ are used to replace the positive and negative parts on controls of the original problem but require to introduce the nonlinear (binary) constraint (89e). We show hereby that we can remove this binary constraint (89e).

Let $(l, u_c, u_d)$ be an admissible solution to $V_t(b, h, w)$ without the binary constraint such that $u_c \times u_d > 0$. We distinguish two cases.

$u_c \leq u_d$ We introduce a new solution $(l', u_c', u_d')$ such that $u_c' = 0$ and $u_d' = u_d - u_c$ with $l' = l \geq c_t \times (u_c' - u_d' - w)^+ = (u_c - u_d - w)^+$. This solution satisfies the binary constraint $u_c' \times u_d' = 0$. And this solution is admissible as

$$0 = u_c' \leq \overline{U}, 0 \leq u_d' \leq u_d \leq -\underline{U} ,$$
$$u_c' + u_d' = u_d - u_c \leq u_d + u_c \leq h ,$$
$$\text{and as } \rho_c \leq 1 \text{ and } \rho_d \leq 1 ,$$
$$\overline{B} - b \geq 0 \geq \rho_c u_c' - \rho_d^{-1} u_d' = \rho_d^{-1} u_c - \rho_d^{-1} u_d \geq \rho_c u_c - \rho_d^{-1} u_d \geq \underline{B} - b .$$

These inequalities plus the fact that $V_{t+1}^\sharp$ is non-increasing makes it possible to say that $(l', u_c', u_d')$ is admissible and achieves the same cost.

$u_c > u_d$ We introduce a new solution $(l', u_c', u_d')$ such that $u_c' = \min(u_c - u_d, \rho_c^{-1}(\overline{B} - b))$ and $u_d' = 0$ with $l' \geq c_t \times (u_c' - u_d' - w)^+ \leq (u_c - u_d - w)^+$. So at optimality we

51

have $l \leq (u_c - u_d - w)^+$. This solution satisfies the binary constraint $u'_c \times u'_d = 0$. And this solution is admissible as

$$0 \leq u'_c \leq u_c \overline{U}, 0 = u'_d \leq -\underline{U} \ ,$$
$$u'_c + u'_d = u_c - u_d \leq u_c + u_d \leq h \ ,$$
$$\overline{B} - b \geq \rho_c u'_c - \rho_d^{-1} u'_d \geq 0 \geq \underline{B} - b \ .$$

Moreover we have

$$b + \rho_c u'_c = b + \rho_c u_c - \rho_c u_d \geq b + \rho_c u_c - \rho_d^{-1} u_d \ ,$$

or

$$b + \rho_c u_c = \overline{B} \geq b + \rho_c u_c - \rho_d^{-1} u_d \ ,$$

so as $V_{t+1}^{\sharp}$ is non increasing we have an admissible solution that achieves a better cost.

We conclude that, from any admissible solution without the binary constraint, we can build an admissible solution satisfying the binary constraint and achieving a lower cost. We prove recursively that this cost is strictly lower if $\rho_c < 1$ and $\rho_d < 1$. Hence, we can remove the binary constraint. The function $V_t$ is therefore the value of a linear program where constraints are linear in the parameters $b, h, w$. Due to the linearity of the expectation, we conclude that $\mathcal{B}$ is a linear Bellman operator. $\square$

**Remark 5.11.** *In the battery renewal problem, we show that the intraday problems are also non-increasing in the capacity $c_d$ because a lower capacity constrains the problem further without changing the objective. We prove by backward induction that the daily value functions are decreasing because in the targets decomposition the instantaneous cost is decreasing and the value function as well. Moreover the problem does not have any constraint.* $\diamond$

# References

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

[2] D. P. Bertsekas. Convergence of discretization procedures in dynamic programming. *IEEE Transactions on Automatic Control*, 20(3):415–419, June 1975.

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

[4] J. Borwein and A. S. Lewis. *Convex analysis and nonlinear optimization: theory and examples.* Springer Science & Business Media, 2010.

[5] P. Carpentier, J.-P. Chancelier, G. Cohen, and M. De Lara. *Stochastic Multi-Stage Optimization. At the Crossroads between Discrete Time Stochastic Control and Stochastic Programming.* Springer-Verlag, Berlin, 2015.

[6] P. Carpentier, J.-P. Chancelier, M. De Lara, F. Pacaud, and T. Rigaut. A template to design online policies for multistage stochastic optimization problems. working paper, Jan. 2019.

[7] P. Carpentier, J.-P. Chancelier, M. De Lara, and T. Rigaut. Time blocks decomposition of multistage stochastic optimization problem. 2018.

[8] P. Carpentier, J.-P. Chancelier, V. Leclère, and F. Pacaud. Stochastic decomposition applied to large-scale hydro valleys management. *European Journal of Operational Research*, 270(3):1086 – 1098, 2018.

[9] J.-P. Chancelier and M. De Lara. Fenchel-moreau conjugation inequalities with three couplings and application to stochastic bellman equation. *arXiv preprint arXiv:1804.03034*, 2018.

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

[11] P. Haessig, B. Multon, H. Ben Ahmed, S. Lascaud, and L. Jamy. Aging-aware NaS battery model in a stochastic wind-storage simulation framework. In *PowerTech 2013*, pages 1–6, Grenoble, France, June 2013.

[12] O. Hernández-Lerma and J. B. Lasserre. *Discrete-time Markov control processes: basic optimality criteria*, volume 30. Springer Science & Business Media, 2012.

[13] B. Heymann, J. Frédéric Bonnans, F. Silva, and G. Jimenez. A Stochastic Continuous Time Model for Microgrid Energy Management. In *ECC2016*, Aalborg, Denmark, June 2016.

[14] B. Heymann and P. Martinon. Optimal Battery Aging : an Adaptive Weights Dynamic Programming Algorithm. *Journal of Optimization Theory and Applications*, Aug. 2018.

[15] B. Heymann, P. Martinon, and F. Bonnans. Long term aging : an adaptative weights dynamic programming algorithm. working paper, July 2016.

[16] A. Iovine, S. B. Siad, G. Damm, E. D. Santis, and M. D. D. Benedetto. Nonlinear control of a dc microgrid for the integration of photovoltaic panels. *IEEE Transactions on Automation Science and Engineering*, 14(2):524–535, April 2017.

[17] R. Le Goff Latimier. *Management and Sizing of an Electric Vehicle Fleet Associated with a Photovoltaic Plant : Stochastic and Distributed Co-optimizationStationary Valorisation of Electric Vehicle Batteries taking into account their aging and availibility.* Theses, Université Paris-Saclay, Sept. 2016.

[18] V. Leclère, P. Carpentier, J.-P. Chancelier, A. Lenoir, and F. Pacaud. Exact converging bounds for stochastic dual dynamic programming via fenchel duality. 2018.

[19] D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi, C. A. Canizares, R. Iravani, M. Kazerani, A. H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, and R. e. a. Palma-Behnke. Trends in Microgrid Control. *IEEE Trans. Smart Grid*, 5(4):1905–1919, 2014.

[20] F. Pacaud, P. Carpentier, J.-P. Chancelier, and M. De Lara. Stochastic optimal control of a domestic microgrid equipped with solar panel and battery. preprint, Jan. 2018.

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

[22] A. Philpott, F. Wahid, and F. Bonnans. MIDAS: A Mixed Integer Dynamic Approximation Scheme. page 22, June 2016.

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

[24] T. Rigaut, P. Carpentier, J. Chancelier, M. D. Lara, and J. Waeytens. Stochastic optimization of braking energy storage and ventilation in a subway station. *IEEE Transactions on Power Systems*, pages 1–1, 2018.

[25] R. T. Rockafellar. Integrals which are convex functionals. *Pacific J. Math.*, 24:525–539, 1968.

[26] R. T. Rockafellar. Integrals which are convex functionals. II. *Pacific J. Math.*, 39:439–469, 1971.

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

[28] R. T. Rockafellar and R. J.-B. Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

[29] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on stochastic programming: modeling and theory.* SIAM, 2009.

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