



Optimization of a domestic microgrid equipped with solar panel and battery: Model Predictive Control and Stochastic Dual Dynamic Programming approaches

François Pacaud, Pierre Carpentier, Jean-Philippe Chancelier, Michel de Lara

► To cite this version:

François Pacaud, Pierre Carpentier, Jean-Philippe Chancelier, Michel de Lara. Optimization of a domestic microgrid equipped with solar panel and battery: Model Predictive Control and Stochastic Dual Dynamic Programming approaches. *Energy Systems*, 2022, 10.1007/s12667-022-00522-7 . hal-03667607

HAL Id: hal-03667607

<https://hal.science/hal-03667607>

Submitted on 13 May 2022

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

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

Optimization of a domestic microgrid equipped with solar panel and battery: Model Predictive Control and Stochastic Dual Dynamic Programming approaches

François Pacaud*, Pierre Carpentier[†], Jean-Philippe Chancelier*,
Michel De Lara*

May 13, 2022

Abstract

In this study, a microgrid with storage (battery, hot water tank) and solar panel is considered. We benchmark two algorithms, MPC and SDDP, that yield online policies to manage the microgrid, and compare them with a rule based policy. Model Predictive Control (MPC) is a well-known algorithm which models the future uncertainties with a deterministic forecast. By contrast, Stochastic Dual Dynamic Programming (SDDP) models the future uncertainties as stagewise independent random variables with known probability distributions. We present a scheme, based on out-of-sample validation, to fairly compare the two online policies yielded by MPC and SDDP. Our numerical studies put to light that MPC and SDDP achieve significant gains compared to the rule based policy, and that SDDP overperforms MPC not only on average but on most of the out-of-sample assessment scenarios.

1 Introduction

1.1 Background introduction

A microgrid is a local energy network that produces part of its energy and controls its own demand. Such systems are complex to control because, on the one hand, of the different stocks and interconnections, and, on the other hand, of electrical demands and weather conditions (heat demand and renewable energy production) that are highly variable and hard to predict at local scale (see [1, 2] for a panorama of the challenges faced when controlling microgrids).

*CERMICS, Ecole des Ponts, Marne-la-Vallée, France

[†]UMA, ENSTA Paris, Institut Polytechnique de Paris, Palaiseau, France

Abbreviations	
AR	Auto-regressive
EMS	Energy Management System
MPC	Model Predictive Control
SDP	Stochastic Dynamic Programming
SDDP	Stochastic Dual Dynamic Programming
Physical variables	
t	Time
Δ	Time step (15mn)
T_0	Horizon (24h)
p_t^e	Electricity price (Euro €)
p_t^d	Thermal comfort (virtual price)
A_p	Surface of solar panel (m^2)
ρ_c	Battery charge rate
ρ_d	Battery discharge rate
β_h	Conversion yield for hot water tank
Decision variables and uncertainties	
$(\Omega, \mathcal{F}, \mathbb{P})$	Probability space
\mathbf{B}_t	Level of energy in battery (kWh)
\mathbf{H}_t	Level of energy in hot water tank (kWh)
\mathbf{D}_t^{el}	Electrical demand (kW)
\mathbf{D}_t^{th}	Hot water demand (kW)
Φ_t^{pv}	Production of the solar panel (kW)
θ_t^i	Inner temperature ($^{\circ}C$)
θ_t^w	Walls temperature ($^{\circ}C$)
θ_t^e	Outdoor temperature ($^{\circ}C$)
\mathbf{F}_t^b	Energy exchanged with the battery (kW)
\mathbf{F}_t^w	Energy injected in the hot water tank (kW)
\mathbf{F}_t^h	Energy injected in the electrical heater (kW)
$\mathbf{U}_t = (\mathbf{F}_t^b, \mathbf{F}_t^w, \mathbf{F}_t^h)$	Controls
$\mathbf{W}_t = (\mathbf{D}_t^{el}, \mathbf{D}_t^{th}, \Phi_t^{pv})$	Uncertainties
$\mathbf{X}_t = (\mathbf{B}_t, \mathbf{H}_t, \theta_t^w, \theta_t^i)$	States
Mappings	
$f_t : \mathbb{X}_t \times \mathbb{U}_t \times \mathbb{W}_{t+1} \rightarrow \mathbb{X}_{t+1}$	Linear dynamics
$L_t : \mathbb{X}_t \times \mathbb{U}_t \times \mathbb{W}_{t+1} \rightarrow \mathbb{R}$	Convex operational cost
$K : \mathbb{X}_T \rightarrow \mathbb{R}$	Convex final cost
$\pi_t : \mathbb{X}_t \rightarrow \mathbb{U}_t$	Control policy

Table 1: Nomenclature

We consider here a domestic microgrid equipped with a battery, an electrical hot water tank and a solar panel, as in Figure 1. The microgrid is connected to an external grid to import electricity when needed. The battery stores energy when external grid prices are low or when the production of the solar panel is above the electrical demand. The house’s envelope also plays the role of heat storage. As a consequence, the system has four stocks to store energy: a battery, a hot water tank, and two passive stocks being the house’s walls and inner rooms. Two kinds of uncertainties affect the system: the electrical and domestic hot water demands are not known in advance; the production of the solar panel is substantially perturbed by the variable weather nebulosity.

We aim to compare two classes of algorithms to tackle uncertainties in a microgrid Energy Management System (EMS). The Model Predictive Control (MPC) algorithm (or its stochastic variant, Stochastic Model Predictive Control) relies on a mathematical representation of the future uncertainties as deterministic forecasts; then, MPC computes decisions online as solutions of a deterministic multistage optimization problem. Stochastic Dual Dynamic Programming (SDDP) relies on a mathematical representation of the future uncertainties as stagewise independent random variables with known probability distributions; then, SDDP computes offline a set of value functions by backward induction, and computes online decisions as solutions of a single stage stochastic optimization problem, using the value functions. We present a fair comparison of these two algorithms, and highlight the pros and cons of both methods.

1.2 Literature review

Optimization and energy management systems

EMS are integrated automated tools used to monitor and control energy systems. In [1], the authors give an overview of the use of optimization methods in the design of EMS. The MPC algorithm [3] and its stochastic variant, Stochastic Model Predictive Control (SMPC) [4], have been widely used to control EMS. We refer the reader to [5] for applications of MPC in buildings. In [6], the authors use MPC for the optimal control of a domestic microgrid, and investigate how to balance the uncertainties of renewable energies with the microgrid. In [7], the authors apply SMPC to the management of an isolated microgrid, and highlight the benefit of this method; an application of SMPC in buildings is presented in [8]; a variant based on robust optimization is proposed in [9].

Stochastic optimization

At local scale, electrical demand and production are highly variable, especially as microgrids are expected to absorb renewable energies. This leads to pay attention to stochastic optimization approaches [10]. Stochastic optimization has been widely applied to hydrovalleys management [11]. Other applications have arisen recently, such as integration of wind energy and storage [12] or insulated microgrids management [13, 14].

Stochastic Dynamic Programming (SDP) [15] is a general method to solve stochastic

optimal control problems. In energy applications, a variant of SDP, Stochastic Dual Dynamic Programming (SDDP), has demonstrated its adequacy for large scale convex applications. SDDP was first described in the seminal paper [11]; we refer to [16] for a generic description of the algorithm and its application to the management of hydrovalleys; a proof of convergence in the linear case is given in [17], and in the convex case in [18]. Recent articles have applied SDDP to the management of energy systems. In [19], numerical experiments show that SDDP yields better results than a myopic policy. In [20], SDDP is applied to the dispatch of energy inside the German national grid, under time correlated uncertainties; the authors observe that SDDP achieves better performances than those of a deterministic-based policy. Other Approximate Dynamic Programming algorithms have been designed to tackle different stochastic optimal control problems like, for instance, incorporating probability constraints [21].

1.3 Main contributions and structure of the paper

We provide a rigorous mathematical formulation of the optimal management of a domestic microgrid — equipped with a battery, an electrical hot water tank and a solar panel, and connected to an external supply network — under stochasticity of demand and of renewable energy production. To manage the microgrid, we design online policies using two different algorithms, MPC and SDDP. Then, we develop a fair benchmark methodology to compare the two algorithms, based on a realistic use case. The comparison reveals that SDDP overperforms MPC not only on average but, interestingly, on most of the out-of-sample assessment scenarios.

The paper is organized as follows. In Sect. 2, we detail the modeling of a small residential microgrid and formulate a mathematical multistage stochastic optimization problem. Then, we outline the two algorithms, MPC and SDDP, in Sect. 3. Finally, in Sect. 4 we detail the benchmark methodology and we provide numerical results on the systematic comparison of MPC, SDDP and a rule based policy. Sect. 5 concludes and the Appendix 6 provides details on the physical equations of the microgrid.

2 Optimization problem statement

We consider the optimal management of a microgrid which, as depicted in Figure 1, consists of a single house equipped with a battery and an electrical hot water tank. An electrical heater can produce heat in winter, and a solar panel can produce energy locally. The decision maker aims at minimizing the energy bill — that is, the cost of the possible recourse energy supplied by the external network — while satisfying the energy demands (hot water and electricity), and ensuring a minimal thermal comfort.

In this section, we write up a multistage stochastic optimization problem. As decisions are taken at discrete time steps, we start by discretizing the time interval in §2.1. Then, we introduce the uncertainties in §2.2, the controls and the stocks in §2.3 and §2.4. We detail

the nonanticipativity constraints in §2.5, the bounds constraints in §2.6 and the objective function in §2.7. Finally, we formulate a multistage stochastic optimization problem in §2.8.

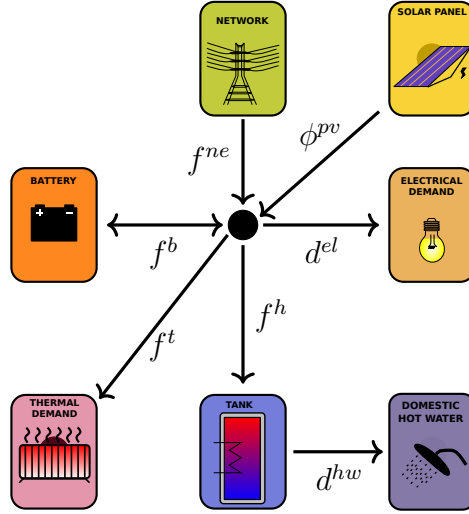


Figure 1: Electrical microgrid

2.1 Decisions are taken at discrete time steps

The EMS takes decisions every 15 minutes to control the system. We consider a time interval $\Delta = 15\text{mn}$, a time horizon $T_0=24\text{h}$ and a number $T=T_0/\Delta=96$ of time steps. We adopt the following convention for discrete time processes: for each time step $t \in \{0, 1, \dots, T\}$, x_t denotes the value of the variable x at the beginning of the interval $[t\Delta, (t+1)\Delta)$. Otherwise stated, we will denote by $[t, t+1)$ the continuous time interval $[t\Delta, (t+1)\Delta)$.¹

2.2 Modeling uncertainties as random variables

Because of their unpredictable nature, the EMS cannot anticipate the values of electrical and thermal demands, nor the production of the solar panel. We choose to model these quantities as random variables over a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. We adopt the following convention: a random variable will be denoted by an uppercase bold letter \mathbf{Z} , and its realization for a given outcome $\omega \in \Omega$ will be denoted in lowercase $z = \mathbf{Z}(\omega)$. We denote by \mathbf{D}_t^{el} the electrical demand, \mathbf{D}_t^{th} the hot water demand and Φ_t^{pv} the production of the solar panel, all being real-valued random variables. For each time step $t = 1, \dots, T$, we define the (uncertainty)

¹Here, by using the notation $[t, t+1)$, we mean *the time interval between t and $t+1$, excluding $t+1$* . Indeed, we denote a time interval between two decisions by $[t, t+1)$, and not by $[t, t+1]$, to indicate that a decision is taken at the beginning of the time interval $[t, t+1)$, and that a new one will be taken at the beginning of the time interval $[t+1, t+2)$, and that these two consecutive intervals do not overlap.

random variable

$$\mathbf{W}_t = (\mathbf{D}_t^{el}, \mathbf{D}_t^{th}, \Phi_t^{pv}) . \quad (1)$$

The uncertainty \mathbf{W}_t is a multivariate random variable taking values in $\mathbb{W}_t = \mathbb{R}^3$.

2.3 Modeling controls as random variables

As decisions depend on the previous uncertainties, controls are random variables. At the beginning of the time interval $[t, t+1)$, the EMS takes three decisions:

- \mathbf{F}_t^b , how much energy to charge in/discharge from the battery,
- \mathbf{F}_t^w , how much energy to store in the electrical hot water tank,
- \mathbf{F}_t^h , how much energy to inject in the electrical heater,

during the time interval $[t, t+1)$. For each time step $t = 1, \dots, T$, we define the decision multivariate random variable, taking values in $\mathbb{U}_t = \mathbb{R}^3$, as

$$\mathbf{U}_t = (\mathbf{F}_t^b, \mathbf{F}_t^w, \mathbf{F}_t^h) . \quad (2)$$

During the time interval $[t, t+1)$, the EMS imports an energy quantity \mathbf{F}_{t+1}^{ne} from the external network in order to fulfill the load balance equation

$$\mathbf{F}_{t+1}^{ne} + \Phi_{t+1}^{pv} = \mathbf{F}_t^b + \mathbf{F}_t^w + \mathbf{F}_t^h + \mathbf{D}_{t+1}^{el} , \quad (3)$$

whatever the demand \mathbf{D}_{t+1}^{el} and the production of the solar panel Φ_{t+1}^{pv} , unknown at the beginning of the time interval $[t, t+1)$. On the left-hand side of Equation (3), the load consists of

- \mathbf{F}_{t+1}^{ne} , the energy surplus or shortage (when $\mathbf{F}_{t+1}^{ne} < 0$, one wastes the surplus; when $\mathbf{F}_{t+1}^{ne} > 0$, one imports the shortage from the network),
- Φ_{t+1}^{pv} , the production of the solar panel,

all during the time interval $[t, t+1)$. On the right-hand side of Equation (3), the electrical demand is the sum of

- \mathbf{F}_t^b , the energy exchanged with the battery,
- \mathbf{F}_t^w , the energy injected into the electrical hot water tank,
- \mathbf{F}_t^h , the energy injected in the electrical heater,
- \mathbf{D}_{t+1}^{el} , the electrical demands (lightning, cooking...), aggregated in a single demand,

all during the time interval $[t, t+1)$.

Later, we will aggregate the solar panel production Φ_{t+1}^{pv} with the demands \mathbf{D}_{t+1}^{el} in Equation (3), as these two quantities appear only by their difference.

2.4 States and dynamics

The state is the multivariate random variable

$$\mathbf{X}_t = (\mathbf{B}_t, \mathbf{H}_t, \boldsymbol{\theta}_t^w, \boldsymbol{\theta}_t^i), \quad (4)$$

which consists of the stocks \mathbf{B}_t in the battery and \mathbf{H}_t in the electrical hot water tank, plus the two temperatures $(\boldsymbol{\theta}_t^w, \boldsymbol{\theta}_t^i)$ of the thermal envelope. Thus, the state random variable \mathbf{X}_t takes values in $\mathbb{X}_t = \mathbb{R}^4$.

The discrete time dynamics $f_t : \mathbb{X}_t \times \mathbb{U}_t \times \mathbb{W}_{t+1} \rightarrow \mathbb{X}_{t+1}$ describes the time evolution

$$\mathbf{X}_{t+1} = f_t(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}) \quad (5)$$

of the state, where f_t is a piecewise linear function (a property that will prove important for the SDDP algorithm), that corresponds to the integration of the continuous dynamics (21)-(22)-(24) given in Appendix 6. We suppose that we start from a given state $x_0 \in \mathbb{X}_0 = \mathbb{R}^4$, thus adding the constraint $\mathbf{X}_0 = x_0$.

2.5 Nonanticipativity constraints

The future realizations of uncertainties are unknown. Thus, decisions at time step t are functions of previous history only, that is, the information collected between step 0 and step t . Such a constraint is encoded as an algebraic constraint, using the tools of Probability theory [22], in the so-called *nonanticipativity constraints* written as

$$\sigma(\mathbf{U}_t) \subset \mathcal{F}_t, \quad (6a)$$

where $\sigma(\mathbf{U}_t)$ is the σ -algebra generated by the random variable \mathbf{U}_t and $\mathcal{F}_t = \sigma(\mathbf{W}_1, \dots, \mathbf{W}_t)$ is the σ -algebra generated by the previous uncertainties $(\mathbf{W}_1, \dots, \mathbf{W}_t)$. If Constraint (6a) holds true, the Doob Lemma [22] ensures that there exists a measurable function $\pi_t : \mathbb{W}_1 \times \dots \times \mathbb{W}_t \rightarrow \mathbb{U}_t$ such that

$$\mathbf{U}_t = \pi_t(\mathbf{W}_1, \dots, \mathbf{W}_t). \quad (6b)$$

This is how we turn an (abstract) algebraic constraint into a more practical functional constraint. The function π_t is called a *policy*.

2.6 Bounds constraints

The stocks \mathbf{B}_t in the battery and \mathbf{H}_t in the tank are bounded:

$$\underline{b} \leq \mathbf{B}_t \leq \bar{b}, \quad 0 \leq \mathbf{H}_t \leq \bar{h}. \quad (7)$$

At time step t , the control \mathbf{F}_t^b must ensure that the next state \mathbf{B}_{t+1} is admissible, that is, satisfies $\underline{b} \leq \mathbf{B}_{t+1} \leq \bar{b}$, which, by time discretization of (21), is equivalent to the two inequalities²

$$\underline{b} \leq \mathbf{B}_t + \Delta \left[\rho_c (\mathbf{F}_t^b)^+ + \frac{1}{\rho_d} (\mathbf{F}_t^b)^- \right] \leq \bar{b}, \quad (8)$$

²We have used the notation $f^+ = \max\{0, f\}$ and $f^- = \max\{0, -f\}$.

with ρ_c and ρ_d being the charge and discharge efficiencies of the battery. Thus, the constraints on \mathbf{F}_t^b depends on the stock \mathbf{B}_t . The same reasoning applies for the tank energy \mathbf{F}_t^w and the stock \mathbf{H}_t . Furthermore, we set bound constraints on controls:

$$-\bar{f}^b \leq \mathbf{F}_t^b \leq \bar{f}^b, \quad 0 \leq \mathbf{F}_t^w \leq \bar{f}^w, \quad 0 \leq \mathbf{F}_t^h \leq \bar{f}^h. \quad (9)$$

Finally, the load-balance equation (3) also acts as a constraint on the controls. We gather all these constraints into an admissible subset $\mathcal{U}_t^{ad}(\mathbf{X}_t)$ of $\mathbb{U}_t = \mathbb{R}^3$, depending on the current state \mathbf{X}_t , giving the constraint³

$$\mathbf{U}_t \in \mathcal{U}_t^{ad}(\mathbf{X}_t). \quad (10)$$

Note that we do not enforce any explicit bounds on the inner temperature. Instead, we choose to add a penalization term in the objective function if the temperature is below a given threshold, as explained below.

2.7 Objective function

At time step t , the instantaneous cost $L_t : \mathbb{X}_t \times \mathbb{U}_t \times \mathbb{W}_{t+1} \rightarrow \mathbb{R}$ aggregates two different costs as in the formula

$$L_t(x_t, u_t, w_{t+1}) = p_t^e \times \max\{0, f_{t+1}^{ne}\} + p_t^d \times \max\{0, \bar{\theta}_t^i - \theta_t^i\}, \quad (11)$$

where f_{t+1}^{ne} is a function of (u_t, w_{t+1}) by (3), and θ_t^i is part of the state x_t in (4).

First, one pays a unitary price p_t^e to import electricity from the external network between step t and $t+1$; hence, the electricity cost is equal to $p_t^e \times \max\{0, f_{t+1}^{ne}\}$. Second, if the indoor temperature is below a given threshold, we penalize the induced discomfort with a cost $p_t^d \times \max\{0, \bar{\theta}_t^i - \theta_t^i\}$, where p_t^d is a virtual price of discomfort (we choose not to penalize the temperature in case it is above a given threshold, as we do not consider any air conditioning in this study). The cost L_t is a convex piecewise linear function, a property that will prove important for the SDDP algorithm. To ensure that stocks are not empty at the final time step T , we add a convex piecewise linear final cost $K : \mathbb{X}_T \rightarrow \mathbb{R}$ of the form

$$K(x_T) = \kappa \times \max\{0, x_0 - x_T\}, \quad (12)$$

where κ is a positive penalization coefficient (calibrated by trial and error).

As decisions \mathbf{U}_t and states \mathbf{X}_t are random, the costs $L_t(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1})$ and $K(\mathbf{X}_T)$ become also random variables. We choose to minimize the expected value of the daily operational cost, that is,

$$\mathbb{E} \left[\sum_{t=0}^{T-1} L_t(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}) + K(\mathbf{X}_T) \right], \quad (13)$$

yielding the expected value of a convex piecewise linear cost.

³More formally, for each time step $t = 1, \dots, T$, \mathcal{U}_t^{ad} is a nonempty set-valued mapping $\mathcal{U}_t^{ad} : \mathbb{X}_t \rightrightarrows \mathbb{U}_t$.

2.8 Stochastic optimal control formulation

Finally, the EMS problem is written as a stochastic optimal control problem

$$\min_{\mathbf{X}, \mathbf{U}} \mathbb{E} \left[\sum_{t=0}^{T-1} L_t(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}) + K(\mathbf{X}_T) \right], \quad (14a)$$

$$\text{s.t. } \mathbf{X}_{t+1} = f_t(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}), \quad \mathbf{X}_0 = x_0, \quad t = 0, \dots, T-1, \quad (14b)$$

$$\mathbf{U}_t \in \mathcal{U}_t^{ad}(\mathbf{X}_t), \quad t = 0, \dots, T-1, \quad (14c)$$

$$\sigma(\mathbf{U}_t) \subset \mathcal{F}_t, \quad t = 0, \dots, T-1. \quad (14d)$$

Problem (14) expresses that the microgrid manager aims to minimize the expected value of the costs while satisfying the dynamics, the control bounds and the nonanticipativity constraints.

3 Resolution methods

The exact resolution of Problem (14) is out of reach in general. We propose two different algorithms that provide policies⁴ $\pi_t : \mathbb{W}_1 \times \dots \times \mathbb{W}_t \rightarrow \mathbb{U}_t$ that map available information w_1, \dots, w_t at step t to a decision u_t .

In §3.1, we start by presenting how to design management policies with the MPC algorithm. Then, we depict SDDP-based policies in §3.2. Both methods use the dynamics f_t in (5), the constraints sets $\mathcal{U}_t^{ad}(\cdot)$ in (10), and the cost functions L_t in (11) and K in (12).

3.1 Model Predictive Control (MPC)

Be it ordinary or stochastic, MPC is a classical algorithm that is commonly used to solve stochastic optimization problems. Regarding MPC, we follow [15]. At time step t , we consider a deterministic forecast $(\bar{w}_{t+1}, \dots, \bar{w}_T)$ of the future uncertainties $(\mathbf{W}_{t+1}, \dots, \mathbf{W}_T)$ (see §6.4 for more details) and we solve the following deterministic problem, where the state x_t is given:

$$\min_{(u_t, \dots, u_{T-1})} \sum_{j=t}^{T-1} L_j(x_j, u_j, \bar{w}_{j+1}) + K(x_T), \quad (15a)$$

$$\text{s.t. } x_{j+1} = f_j(x_j, u_j, \bar{w}_{j+1}), \quad j = t, \dots, T-1, \quad (15b)$$

$$u_j \in \mathcal{U}_j^{ad}(x_j), \quad j = t, \dots, T-1. \quad (15c)$$

Then, we retrieve the optimal decisions $(u_t^\#, \dots, u_{T-1}^\#)$ and only keep the first decision $u_t^\#$ to control the system at the beginning of the time interval $[t, t+1)$. This procedure is then restarted at step $t+1$. Thus, MPC solves an optimization problem at each time step, with

⁴See §2.5 on how the algebraic nonanticipativity constraint (14d) can be turned into a more practical functional constraint, making it possible to search for solutions that are policies.

a time span going from the current time step t to the final time step T . Then, at the next time $t + 1$, the optimizer updates the scenario $(\bar{w}_{t+1}, \dots, \bar{w}_T)$ to take into account the latest observation made at $t + 1$.

3.2 Stochastic Dual Dynamic Programming (SDDP)

As said in §1.2, SDDP is an algorithm widely used to optimize energy systems. The SDDP algorithm provides, at each time step, a value function and an online control policy. Whereas the value functions are computed offline (hence with offline data), online control policies are computed taking into account an online probability distribution on the next period noise.

Dynamic Programming and Bellman principle

The Dynamic Programming method [23] provides solutions of Problem (14) as state feedbacks $\pi_t : \mathbb{X}_t \rightarrow \mathbb{U}_t$ (these feedbacks are optimal when the noise process is made of independent random variables). Dynamic Programming makes use of a sequence of value functions, obtained offline by setting $V_T(x_T) = K(x_T)$ and by solving backward in time the recursive functional equations

$$V_t(x_t) = \min_{u \in \mathcal{U}_t^{ad}(x_t)} \int_{\mathbb{W}_{t+1}} \left[L_t(x_t, u, w_{t+1}) + V_{t+1}(f_t(x_t, u, w_{t+1})) \right] \mu_{t+1}^{of}(dw_{t+1}), \quad (16)$$

where μ_{t+1}^{of} is a (offline) probability distribution on \mathbb{W}_{t+1} . Once these functions obtained, we compute a decision at time step t as a state feedback

$$\pi_t(x_t) \in \arg \min_{u \in \mathcal{U}_t^{ad}(x_t)} \int_{\mathbb{W}_{t+1}} \left[L_t(x_t, u, w_{t+1}) + V_{t+1}(f_t(x_t, u, w_{t+1})) \right] \mu_{t+1}^{on}(dw_{t+1}), \quad (17)$$

where μ_{t+1}^{on} is an online probability distribution on \mathbb{W}_{t+1} . This method proves to be optimal when the random variables $\mathbf{W}_1, \dots, \mathbf{W}_T$ are stagewise independent and when $\mu_t^{on} = \mu_t^{of}$ is the probability distribution of \mathbf{W}_t .

Description of Stochastic Dual Dynamic Programming

Dynamic Programming suffers from the well-known curse of dimensionality [23]: its numerical resolution fails for state dimension typically greater than 4 when value functions are computed on a numerical grid. When uncertainties are stagewise independent random variables, costs L_t and K are convex and dynamics f_t are linear, the SDDP algorithm provides a solution to Problem (14) where the value functions are represented by convex polyhedral functions [18]. Most achievements of SDDP have been obtained in the linear case — when

all the costs are linear — as each subproblem (16) can be solved with the simplex algorithm. However, the extension to the generic convex case has not been studied extensively.

SDDP provides an outer approximation of the value function V_t in (16). Provided a bundle of k supporting hyperplanes $\{(\lambda_t^j, \beta_t^j)\}_{j=1, \dots, k}$ (with, for each $j = 1, \dots, k$, $\lambda_t^j \in \mathbb{R}^4$ and $\beta_t^j \in \mathbb{R}$), the corresponding outer approximation \underline{V}_t^k is given by

$$\underline{V}_t^k(x_t) = \min_{\varrho_t \in \mathbb{R}} \varrho_t, \quad (18a)$$

$$\text{s.t. } \langle \lambda_t^j, x_t \rangle + \beta_t^j \leq \varrho_t, \quad \forall j = 1, \dots, k. \quad (18b)$$

SDDP takes as input an initial value function \underline{V}_{t+1}^0 (usually $-\infty$) and an initial state x_0 . In the algorithm, we assume that the probability distributions have finite support $\{w_{t+1}^1, \dots, w_{t+1}^S\}$. The offline distribution μ_{t+1}^{of} is then written in the form $\mu_{t+1}^{of} = \sum_{s=1}^S p_{t+1}^s \delta_{w_{t+1}^s}$, where $\delta_{w_{t+1}^s}$ is the Dirac measure at w_{t+1}^s and $p_{t+1}^1, \dots, p_{t+1}^S$ are probability weights. Then, each iteration k of SDDP encompasses two passes.

- During the *forward pass*, we draw a scenario w_1^k, \dots, w_T^k of uncertainties, and obtain a state trajectory $\{x_t^k\}_{t=0 \dots T}$ along this scenario as follows. Starting from initial state x_0 , we compute x_{t+1}^k from x_t^k in an iterative fashion: i) we obtain a control u_t^k at time step t , using the available \underline{V}_{t+1}^k function, by

$$u_t^k \in \arg \min_{u \in \mathcal{U}_t^{ad}(x_t)} \sum_{i=1}^S p_{t+1}^i [L_t(x_t^k, u, w_{t+1}^i) + \underline{V}_{t+1}^k(f_t(x_t^k, u, w_{t+1}^i))] \quad , \quad (19a)$$

and ii), we set $x_{t+1}^k = f_t(x_t^k, u_t^k, w_{t+1}^k)$ where f_t is the piecewise linear dynamics in (5).

- During the *backward pass*, we update the approximated value functions $\{\underline{V}_t^k\}_{t=0, \dots, T}$ backward in time along the trajectory $\{x_t^k\}_{t=0, \dots, T}$. At time step t , we solve the problem

$$\varrho_t^{k+1} = \min_{u \in \mathcal{U}_t^{ad}(x_t)} \sum_{i=1}^S p_{t+1}^i [L_t(x_t^k, u, w_{t+1}^i) + \underline{V}_{t+1}^{k+1}(f_t(x_t^k, u, w_{t+1}^i))] \quad , \quad (19b)$$

and we obtain a new cut $(\lambda_t^{k+1}, \beta_t^{k+1})$ where λ_t^{k+1} is a subgradient of the optimal cost function (19b) evaluated at the point $x_t = x_t^k$,

$$\lambda_t^{k+1} \in \sum_{i=1}^S p_{t+1}^i [\partial_x L_t(x_t^k, u_t^{k+1}, w_{t+1}^i) + \partial_x \underline{V}_{t+1}^{k+1}(f_t(x_t^k, u_t^{k+1}, w_{t+1}^i))] ,$$

and where $\beta_t^{k+1} = \varrho_t^{k+1} - \langle \lambda_t^{k+1}, x_t^k \rangle$. This new cut makes it possible to update the function \underline{V}_t^{k+1} by the formula $\underline{V}_t^{k+1} = \max\{\underline{V}_t^k, \langle \lambda_t^{k+1}, \cdot \rangle + \beta_t^{k+1}\}$.

We use the stopping criterion introduced in [16] to stop SDDP once the gap between the upper and lower bounds is lower than 0.1 %. In practice, the subproblems (19a) and (19b)

are linear, and can be solved efficiently with any linear programming solver. At iteration k , the lower bound is given directly by $\underline{V}_0(x_0)$. The upper bound is computed by running N_{sim} forward passes of SDDP on a fixed set of scenarios $w^1, \dots, w^{N_{sim}}$; the costs associated with each scenario are then averaged to get the upper bound (the method is thus akin to a Monte Carlo simulation). We obtain a sequence $\{\underline{V}_t\}_{t=0, \dots, T}$ of functions, that are lower approximations of the original Bellman functions.

Obtaining online controls with SDDP

We obtain an online policy by means of the following procedure:

- approximated value functions $\{\underline{V}_t\}$ are computed offline with the SDDP algorithm (see §3.2),
- the approximated value functions $\{\underline{V}_t\}$ are then used to compute online a decision at any time step t for any state x_t as follows.

We compute the SDDP policy π_t^{sddp} by

$$\pi_t^{sddp}(x_t) \in \arg \min_{u \in \mathcal{U}_t^{ad}(x_t)} \sum_{i=1}^S p_{t+1}^i [L_t(x_t, u, w_{t+1}) + \underline{V}_{t+1}(f_t(x_t, u, w_{t+1}))] \quad , \quad (20)$$

which corresponds to replacing the value function V_{t+1} in Equation (17) with its approximation \underline{V}_{t+1} . The decision $\pi_t^{sddp}(x_t)$ is used to control the system between steps t and $t+1$. Then, we solve Problem (20) at step $t+1$.

3.3 Discussion

In this section, we have introduced two methods to design management policies, the first one based on MPC, the second on SDDP. Both methods differ on how they model the future uncertainties. In SDDP, one represents the future as a sequence of independent random variables, whereas in MPC it is with a deterministic forecast. It remains now to compare the policy π^{sddp} with the policy π^{mpc} .

In the next Sect. 4, we compare the performances of π^{mpc} and π^{sddp} on a set of assessment scenarios, using a simulator. We depict in Figure 2 the flow chart of the simulation procedure. The devised policies are used to compute a decision u_t at each time step t , using the information already available at step t . Then, by comparing the total costs obtained and repeating the procedure on a bundle of scenarios, we are able to draw conclusions about the respective performances of each policy.

4 Numerical results

In §4.1, we describe a case study. In §4.2, we develop a protocol to fairly compare the MPC and SDDP algorithms altogether with a rule based policy, then discuss the results

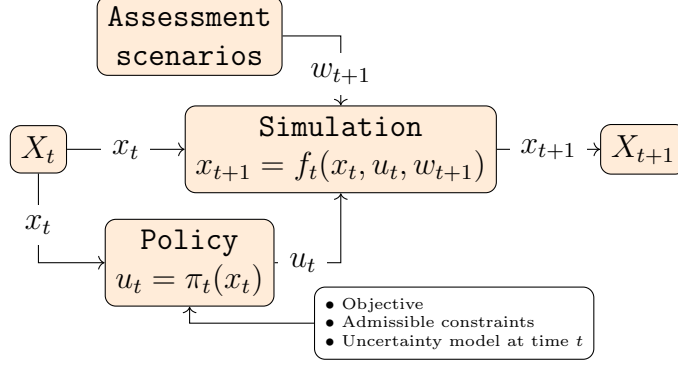


Figure 2: Flow chart of the simulation procedure

obtained. In §4.3, we quantify the robustness of MPC and SDDP with respect to the level of uncertainty.

4.1 Case study

Settings We aim to solve the stochastic optimization problem (14) over one day, with 96 time steps. The battery size is 3 kWh, and the hot water tank has a capacity of 120 l. We suppose that the house has a surface $A_p = 20 \text{ m}^2$ of solar panel at disposal, oriented south, and with a yield of 15%. We penalize the recourse variable \mathbf{F}_{t+1}^{ne} in (11) with on-peak and off-peak tariff, corresponding to Électricité de France’s (EDF) individual tariffs. The house’s thermal envelope corresponds to the French RT2012 specifications [24]. Meteorological data comes from Météo France measurements corresponding to the year 2015.

Implementing the algorithms We implement MPC and SDDP in Julia 0.6, using JuMP [25] as a modeler, StochDynamicProgramming.jl as a SDDP solver, and Gurobi 7.02 [26] as a LP solver. All computations run on a Core i7 2.5 GHz processor, with 16GB RAM.

Rule based method We choose to compare the MPC and SDDP algorithms with the following basic decision rule: the battery is charged whenever the solar production Φ^{pv} is available, and discharged to fulfill the demand if there remains enough energy in the battery; the tank is charged ($\mathbf{F}_t^w > 0$) if the tank energy \mathbf{H}_t is lower than \mathbf{H}_0 , the heater \mathbf{F}_t^h is switched on when the temperature is below the setpoint $\bar{\theta}_t^i$, and switched off whenever the temperature is above the setpoint plus a given margin.

4.2 Benchmark

Demand scenarios

Scenarios of electrical and domestic hot water demands, at time steps evenly spaced every $\Delta = 15$ minutes, are generated with the software StRoBe [27]. In Figure 3, 100 scenarios

of electrical and hot water demands are displayed. We observe almost null demand during the night, and demand peaks around midday and 8 pm; peaks in hot water demand correspond to showers. We aggregate the electrical demand \mathbf{D}^{el} minus the production Φ^{pv} of the solar panel in a single variable \mathbf{D}^{el} , so that we consider only two uncertainties $\mathbf{W}_t = (\mathbf{D}_t^{el}, \mathbf{D}_t^{th})$.

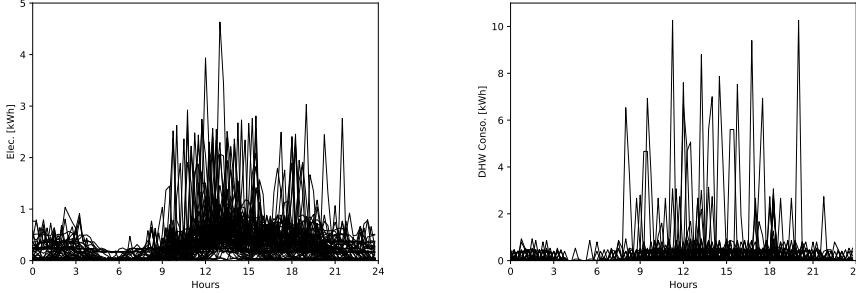


Figure 3: Electrical (left) and domestic hot water (right) demand scenarios

Building offline probability distributions for SDDP

We use the optimization scenarios to build marginal probability distributions μ_t^{of} that will feed the SDDP procedure in (19a)-(19b). In order to obtain a discrete probability distribution at each time t , we use a Lloyd-Max quantization scheme [28] to compute S representative points from the N_{opt} optimization scenarios.

Out of sample assessment of policies

To obtain a fair comparison between SDDP and MPC, we use an out-of-sample validation. We generate 2,000 scenarios of electrical and hot water demands, and we split these scenarios in two distinct parts: the first $N_{opt} = 1,000$ scenarios are called *optimization scenarios*, and the remaining $N_{sim} = 1,000$ scenarios are called *assessment scenarios*.

First, during the offline phase, we use the optimization scenarios to build models for the uncertainties, under the mathematical form required by each algorithm (see Sect. 3). Second, during the online phase, we use the assessment scenarios to compare the policies produced by these algorithms. At time step t during the assessment, the algorithms cannot use the future values of the assessment scenarios, but can take advantage of the observed values up to t to update their statistical models of future uncertainties. Our method differs from [19, 20] where scenarios serve to fit a probability distribution which is used, on the one hand, to design the SDDP algorithm and, on the other hand, to generate new scenarios to assess the performances of SDDP.

MPC procedure

Electrical and thermal demands are naturally correlated in time [29]. To take into account such a dependence across the different time steps, we choose to model the pro-

cess $\mathbf{W}_1, \dots, \mathbf{W}_T$ with an auto-regressive (AR) model and use the deterministic trend to yield the forecast in (15). We detail the overall procedure in §6.4.

SDDP procedure

Computing value functions offline We fit a sequence of finite probability distributions $\{\mu_t^{of}\}_{t=0, \dots, T}$ using only the optimization scenarios. Then we compute a set of value functions with the procedure described in §3.2.

Using value functions online Once the value functions have been obtained by SDDP, we compute online decisions at step t with Equation (20), using a finite online probability distribution μ_t^{on} , fitted with both the optimization scenarios and the past of the current assessment scenario (respecting thus the nonanticipativity constraint).

Assessing on different meteorological conditions

We assess the algorithms on three different days, with different meteorological conditions (see Table 2). Therefore, we use three distinct sets of $N_{opt} + N_{sim}$ scenarios for demands (with N_{opt} the number of optimization scenarios and N_{sim} the number of assessment scenarios), one for each typical day.

	Date	Temp. ($^{\circ}C$)	PV Production (kWh)
Winter Day	February, 19th	3.3	8.4
Spring Day	April, 1st	10.1	14.8
Summer Day	May, 31st	14.1	23.3

Table 2: Different meteorological conditions

These three different days correspond to different heating needs. During Winter day, the heating is maximal, whereas it is medium during Spring day and null during Summer day. The production of the solar panel varies accordingly.

Comparing the algorithms performances

During assessment, we use MPC (see (15)) and SDDP (see (20)) policies to compute online decisions along N_{sim} assessment scenarios. Then, we compare the average electricity bill obtained with these two policies and with the rule based policy. The assessment results are given in Table 3, where means and standard deviation σ are computed with the $N_{sim} = 1,000$ scenarios; the notation \pm corresponds to the 95% confidence interval $\pm 1.96 \frac{\sigma}{\sqrt{N_{sim}}}$.

Regarding mean electricity bills, we observe, on the three last rows in Table 3, that MPC and SDDP yield much better results than the rule based policy. On Winter day, SDDP is slightly better than MPC; on Spring day, they are equivalent; on Summer day, SDDP is able to almost halve the mean costs yielded by MPC (in both case, these costs are low).

	SDDP	MPC	rule based policy
Offline time	50 s	0 s	0 s
Online time	1.5 ms	0.5 ms	0.005 ms
Electricity bill (€)			
Winter day	4.38 ± 0.02	4.59 ± 0.02	5.55 ± 0.02
Spring day	1.46 ± 0.01	1.45 ± 0.01	2.83 ± 0.01
Summer day	0.10 ± 0.01	0.18 ± 0.01	0.33 ± 0.02

Table 3: Comparison of SDDP, MPC and rule based policies

Figure 4 displays the histogram of the difference between SDDP and MPC total costs during Summer day. In this way, for each scenario (and not only in the mean), we can measure the discrepancy between SDDP and MPC performances. We observe that SDDP is better than MPC for about 93% of the scenarios. The distribution in Figure 4 exhibits a heavy tail that reveals the superiority of SDDP on extreme scenarios. Thus, not only SDDP achieves better performance than MPC in the mean, but also for the vast majority of scenarios. Similar analyses hold for Winter and Spring days.

Comparing the algorithms running times

In terms of numerical performance, it takes less than one minute to compute approximated Bellman functions \underline{V}_t as in §3.2 with SDDP on a particular day. Then, the online computation of a single decision takes 1.5 ms on average, compared to 0.5 ms for MPC. Indeed, MPC is favored by the linearity of the optimization Problem (15), whereas, for SDDP, the higher the quantization size S of the online probability distribution μ_t^{on} , the slower is the online resolution of Problem (20), but the more accurate μ_t^{on} is. MPC’s offline resolution time is equal to 0s, as MPC is not an algorithm based on cost-to-go functions, and as we do not include the offline time devoted to find coefficients of an AR process. As we are not in a distributed setting, the computation time is not a challenge here, and computing the value functions or the online decisions is a doable task for a central controller.

Analyzing the use of storage capacities

We analyze now the trajectories of stocks in assessment during Summer day, where heating is off and production of the solar panel is nominal at midday.

Figure 5 displays the state of charge of the battery along a subset of assessment scenarios, for SDDP and MPC. We observe that SDDP charges earlier the battery at its maximum. On the contrary, MPC charges the battery later and does not use the full potential of the battery⁵. The two algorithms discharge the battery to fulfill the evening demands. We notice

⁵In Figure 5, we observe that SDDP charges the battery to its maximum level during day-time, when the solar panels are producing electricity. Then, the battery is fully discharged during the evening to satisfy the local demand when the energy prices are high (evening peak). On the contrary, MPC does not charge the

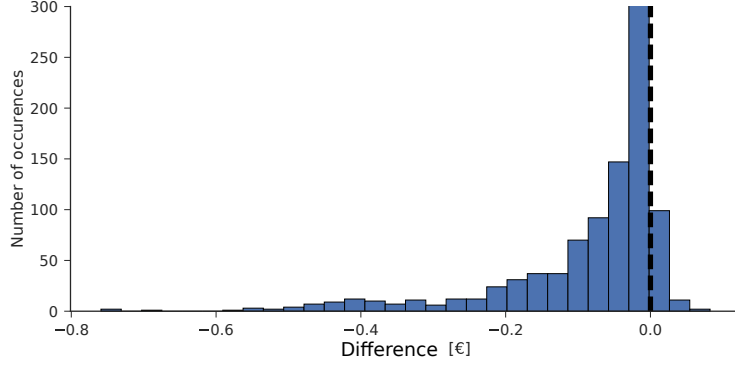


Figure 4: SDDP total costs minus MPC total costs during Summer day

that each trajectory exhibits a single cycle of charge/discharge, thus preserving the battery's aging.

Figure 6 displays the charge of the domestic hot water tank along the same subset of assessment scenarios. We observe a similar behavior as for the battery trajectories: SDDP uses more the electrical hot water tank to store the excess of PV energy, and the level of the tank is greater at the end of the day than in MPC.

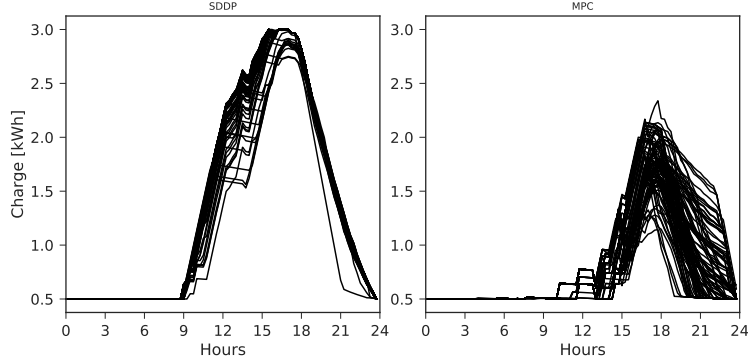


Figure 5: Battery charge trajectories for SDDP (left) and MPC (right) during Summer day

This analysis suggests that SDDP makes a better use of storage capacities than MPC.

battery to its maximum level. Indeed, its forecast only comprises the average energy demand; the algorithm is filling the battery only to satisfy this average demand (in a sense, we are "overfitting" the average scenario). Hence, MPC is unable to anticipate a demand higher than usual, leading to an increase use of the recourse variable (the importation from the external grid) when MPC encounters an unexpected scenario. SDDP does not fall into this pitfall, as it considers a broader range of demands at each time step with its quantized probability laws (and not just the average demand). This pushes SDDP to fill the battery to its maximum level, in anticipation of potential high demands during the evening.

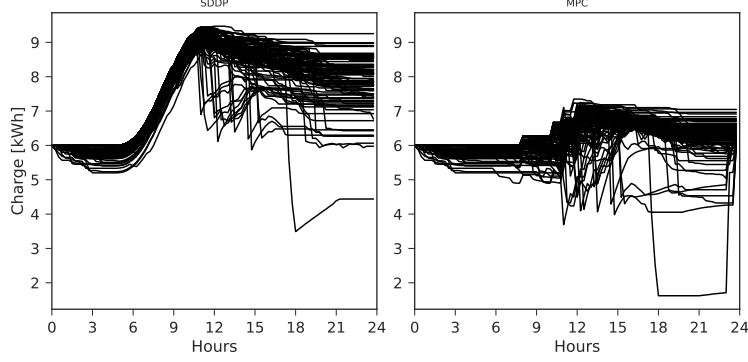


Figure 6: Hot water tank trajectories for SDDP (left) and MPC (right) during Summer day

4.3 Quantifying the sensitivity of MPC and SDDP with respect to the level of uncertainty

Here, we compare the performances of the two algorithms (MPC and SDDP) when the level of uncertainty increases.

Uncertainty model We suppose that, at each time step t , the random variable Φ_t^{pv} (the photovoltaic production in Equation (1)) is given by $\Phi_t^{pv} = \mu_t \times (1 + \epsilon_t)$, where μ_t is deterministic (it corresponds to the forecasted value of Φ_t^{pv}) and where ϵ_t is a zero-mean Gaussian random variable, with standard-deviation $\sigma_t = \sigma_0 + (\sigma_T - \sigma_0)t/T$ increasing linearly over time (where σ_0 is the initial standard-deviation and σ_T the final standard-deviation). We define the level of uncertainty through the value of σ_T : the greater it is, the more difficult it becomes to predict the value of the random variable Φ_t^{pv} . With this setting, we have that $\mathbb{E}(\Phi_t^{pv}) = \mu_t$ and $\text{Var}(\Phi_t^{pv}) = \mu_t^2 \sigma_t^2$. In Figure 7, we display scenarios that are realizations of sequences $\Phi_0^{pv}, \dots, \Phi_T^{pv}$.

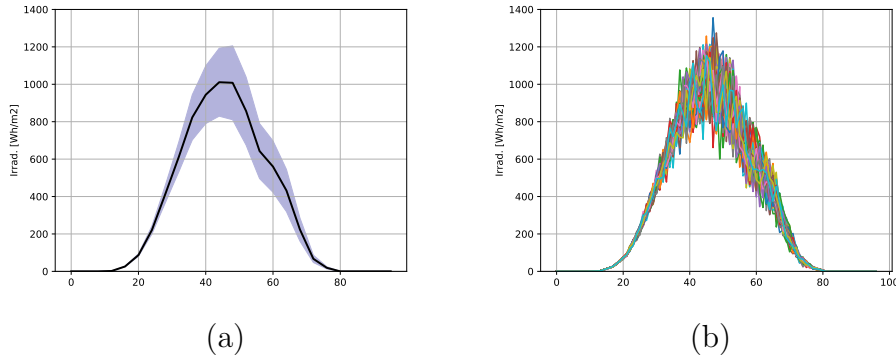


Figure 7: Confidence interval (a) and generation of corresponding assessment scenarios (b) for production of solar panel Φ^{pv} . These scenarios are generated with $\sigma_0 = 0$ and $\sigma_T = 0.2$.

Results We assess SDDP and MPC with different level of uncertainties, that is, with increasing values of σ_T . The costs correspond to the management costs to operate the microgrid during a particular day in Summer, where the production of the solar panel is nominal. The detailed results are given in Figure 8, which shows the evolution of the performance of the two algorithms as a function of the level σ_T of uncertainty. The costs of each algorithm are obtained via Monte-Carlo simulation, with 10,000 assessment scenarios. We observe that MPC’s cost increases quicker than SDDP’s cost when the level of uncertainty increases.

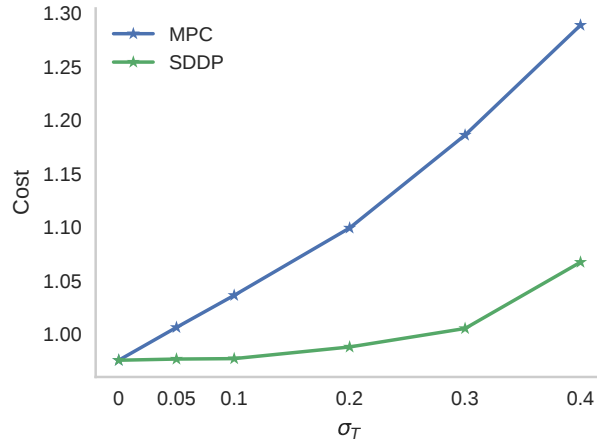


Figure 8: Evolution of the performance of MPC and SDDP as a function of the level σ_T of uncertainty

This second use case allows to quantify the sensitivity of the algorithms with respect to the level of uncertainty. It seems that SDDP behaves better than MPC when facing high uncertainties.

5 Conclusion

We have presented the optimal management of a domestic microgrid, and have compared different management policies (the core of an Energy Management System) under uncertainties. Our results show that the two optimization-based policies (MPC and SDDP) outperform the proposed rule based policy in terms of money savings. Furthermore, SDDP outperforms MPC during Winter and Summer days — and displays similar performance as MPC during Spring day. Even when SDDP and MPC exhibit close average performances, a comparison scenario by scenario shows that SDDP beats MPC most of the time (more than 90% of scenarios during Summer day). Thus, SDDP proves better than MPC to manage uncertainties in our study, although MPC displays also good performances. SDDP makes better use of storage capacities too.

What if we had compared SDDP not only with “ordinary” MPC but with its stochastic variant, Stochastic Model Predictive Control (SMPC)? A systematic analysis done in [30],

using a large dataset of microgrids, reveals that our conclusions remain valid: algorithms based on the offline computation of cost-to-go functions (SDP, SDDP) outperform lookahead algorithms (MPC, SMPC).

Our study can be extended in different directions. First, we could mix SDDP and MPC to grasp the benefits of these two algorithms. Indeed, SDDP is designed to handle the uncertainties variability but fails to capture the time correlation (as it relies on an assumption of stagewise independence), whereas ordinary MPC ignores the uncertainties variability, but considers time correlation by means of a future scenario. Second, we have extended this study in [31], where we applied decomposition methods to optimize microgrids comprising several buildings connected together. Finally, a comparison of MPC and SDDP with novel methods based on reinforcement learning could also be of interest.

References

- [1] D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi, C. A. Cañizares, R. Iravani, M. Kazerani, A. H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke, G. A. Jiménez-Estévez, and N. D. Hatziargyriou, “Trends in microgrid control,” *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1905–1919, 2014.
- [2] T. Morstyn, B. Hredzak, and V. G. Agelidis, “Control strategies for microgrids with distributed energy storage systems: An overview,” *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3652–3666, 2016.
- [3] C. E. Garcia, D. M. Prett, and M. Morari, “Model predictive control: theory and practice—a survey,” *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
- [4] A. Mesbah, “Stochastic model predictive control: An overview and perspectives for future research,” *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 30–44, 2016.
- [5] J. Široký, F. Oldewurtel, J. Cigler, and S. Prívara, “Experimental analysis of model predictive control for an energy efficient building heating system,” *Applied energy*, vol. 88, no. 9, pp. 3079–3087, 2011.
- [6] N. Holjevac, T. Capuder, and I. Kuzle, “Adaptive control for evaluation of flexibility benefits in microgrid systems,” *Energy*, vol. 92, pp. 487–504, 2015.
- [7] D. E. Olivares, J. D. Lara, C. A. Cañizares, and M. Kazerani, “Stochastic-predictive energy management system for isolated microgrids,” *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 2681–2693, 2015.
- [8] R. R. Appino, J. Á. G. Ordiano, R. Mikut, T. Faulwasser, and V. Hagenmeyer, “On the use of probabilistic forecasts in scheduling of renewable energy sources coupled to storages,” *Applied energy*, vol. 210, pp. 1207–1218, 2018.

- [9] K. Paridari, A. Parisio, H. Sandberg, and K. H. Johansson, “Robust scheduling of smart appliances in active apartments with user behavior uncertainty,” *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 247–259, 2016.
- [10] M. De Lara, P. Carpentier, J.-P. Chancelier, and V. Leclère, *Optimization Methods for the Smart Grid*. Conseil Francais de l’Energie, 2014.
- [11] M. V. Pereira and L. M. Pinto, “Multi-stage stochastic optimization applied to energy planning,” *Mathematical programming*, vol. 52, no. 1-3, pp. 359–375, 1991.
- [12] P. Haessig, H. B. Ahmed, and B. Multon, “Energy storage control with aging limitation,” in *2015 IEEE Eindhoven PowerTech*, pp. 1–6, IEEE, 2015.
- [13] B. Heymann, J. F. Bonnans, F. Silva, and G. Jimenez, “A stochastic continuous time model for microgrid energy management,” in *European Control Conference (ECC)*, pp. 2084–2089, IEEE, 2016.
- [14] C. Alasseur, A. Balata, S. B. Aziza, A. Maheshwari, P. Tankov, and X. Warin, “Regression Monte Carlo for microgrid management,” *ESAIM: Proceedings and Surveys*, vol. 65, pp. 46–67, 2019.
- [15] D. P. Bertsekas, *Dynamic programming and optimal control*, vol. 1. Athena Scientific Belmont, MA, third ed., 2005.
- [16] A. Shapiro, “Analysis of stochastic dual dynamic programming method,” *European Journal of Operational Research*, vol. 209, no. 1, pp. 63–72, 2011.
- [17] A. B. Philpott and Z. Guan, “On the convergence of stochastic dual dynamic programming and related methods,” *Operations Research Letters*, vol. 36, no. 4, pp. 450–455, 2008.
- [18] P. Girardeau, V. Leclère, and A. B. Philpott, “On the convergence of decomposition methods for multistage stochastic convex programs,” *Mathematics of Operations Research*, vol. 40, no. 1, pp. 130–145, 2014.
- [19] A. Bhattacharya, J. P. Kharoufeh, and B. Zeng, “Managing energy storage in microgrids: A multistage stochastic programming approach,” *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 483–496, 2018.
- [20] A. Papavasiliou, Y. Mou, L. Cambier, and D. Scieur, “Application of stochastic dual dynamic programming to the real-time dispatch of storage under renewable supply uncertainty,” *IEEE Transactions on Sustainable Energy*, vol. 9, no. 2, pp. 547–558, 2018.
- [21] A. Balata, M. Ludkovski, A. Maheshwari, and J. Palczewski, “Statistical learning for probability-constrained stochastic optimal control,” *European Journal of Operational Research*, vol. 290, no. 2, pp. 640–656, 2021.

- [22] O. Kallenberg, *Foundations of Modern Probability*. Springer-Verlag, New York, second ed., 2002.
- [23] R. Bellman, *Dynamic Programming*. New Jersey: Princeton University Press, 1957.
- [24] Journal Officiel de la République Française, *Arrêté du 28 décembre 2012 relatif aux caractéristiques thermiques et aux exigences de performance énergétique des bâtiments nouveaux*, 2013.
- [25] I. Dunning, J. Huchette, and M. Lubin, “JuMP: A modeling language for mathematical optimization,” *SIAM Review*, vol. 59, no. 2, pp. 295–320, 2017.
- [26] Gurobi Optimization Inc, “Gurobi Optimizer Reference Manual,” 2014.
- [27] R. Baetens and D. Saelens, “Modelling uncertainty in district energy simulations by stochastic residential occupant behaviour,” *Journal of Building Performance Simulation*, vol. 9, no. 4, pp. 431–447, 2016.
- [28] S. Lloyd, “Least squares quantization in PCM,” *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [29] J. Widén and E. Wäckelgård, “A high-resolution stochastic model of domestic activity patterns and electricity demand,” *Applied Energy*, vol. 87, no. 6, pp. 1880–1892, 2010.
- [30] A. Le Franc, P. Carpentier, J.-P. Chancelier, and M. De Lara, “EMSx: A numerical benchmark for energy management systems,” *Energy Systems*, Feb. 2021.
- [31] P. Carpentier, J. P. Chancelier, M. De Lara, and F. Pacaud, “Mixed spatial and temporal decompositions for large-scale multistage stochastic optimization problems,” *Journal of Optimization Theory and Applications*, vol. 186, no. 3, pp. 985–1005, 2020.
- [32] T. Berthou, P. Stabat, R. Salvazet, and D. Marchio, “Development and validation of a gray box model to predict thermal behavior of occupied office buildings,” *Energy and Buildings*, vol. 74, pp. 91–100, 2014.

6 Appendix

In this Appendix, we depict physical equations of the energy system in Figure 1. These equations are naturally written in continuous time t . We model the battery and the hot water tank with stock dynamics, and the dynamics of the house’s temperatures with an electrical analogy.

6.1 Energy storage

We consider a battery, whose state of charge at time t is denoted by $b(t)$. The battery dynamics is given by the differential equation

$$\frac{db}{dt} = \rho_c(f^b(t))^+ - \frac{1}{\rho_d}(f^b(t))^- , \quad (21)$$

with ρ_c and ρ_d being the charge and discharge efficiency and $f^b(t)$ denoting the energy exchange with the battery.

6.2 Electrical hot water tank

We use a simple linear model for the electrical hot water tank dynamics. The enthalpy balance equation writes

$$\frac{dh}{dt} = \beta_h f^w(t) - d^{th}(t) , \quad (22)$$

where

- $f^h(t)$ is the electrical energy used to heat the tank, satisfying

$$0 \leq f^w(t) \leq \bar{f}^w , \quad (23)$$

- $d^{th}(t)$ is the domestic hot water demand,
- β_h is a conversion yield.

6.3 Thermal envelope

We model the evolution of the temperatures inside the house with an electrical analogy: we view temperatures as voltages, walls as capacitors, and thermal flows as currents. A model with 6 resistances and 2 capacitors (R6C2) proves to be accurate to describe small buildings [32]. The model takes into account two temperatures:

- the wall's temperature $\theta^w(t)$,
- the inner temperature $\theta^i(t)$.

Their evolution is governed by the two following differential equations

$$c_m \frac{d\theta^w}{dt} = \underbrace{\frac{\theta^i(t) - \theta^w(t)}{R_i + R_s}}_{\text{Exchange Indoor/Wall}} + \underbrace{\frac{\theta^e(t) - \theta^w(t)}{R_m + R_e}}_{\text{Exchange Outdoor/Wall}} + \underbrace{\gamma f^t(t)}_{\text{Heater}} + \underbrace{\frac{R_i}{R_i + R_s} \Phi^{int}(t)}_{\text{Radiation through windows}} + \underbrace{\frac{R_e}{R_e + R_m} \Phi^{ext}(t)}_{\text{Radiation through wall}} , \quad (24a)$$

$$c_i \frac{d\theta^i}{dt} = \underbrace{\frac{\theta^w(t) - \theta^i(t)}{R_i + R_s}}_{\text{Exchange Indoor/Wall}} + \underbrace{\frac{\theta^e(t) - \theta^i(t)}{R_v}}_{\text{Ventilation}} + \underbrace{\frac{\theta^e(t) - \theta^i(t)}{R_f}}_{\text{Windows}} + \underbrace{(1 - \gamma)f^t(t)}_{\text{Heater}} + \underbrace{\frac{R_s}{R_i + R_s}\Phi^{int}(t)}_{\text{Radiation through windows}}, \quad (24b)$$

where we denote

- the energy injected in the heater by $f^h(t)$,
- the external temperature by $\theta^e(t)$,
- the radiation through the wall by $\Phi^{ext}(t)$,
- the radiation through the windows by $\Phi^{int}(t)$.

The time-varying quantities $\theta^e(t)$, $\Phi^{int}(t)$ and $\Phi^{ext}(t)$ are exogenous. We denote by $R_i, R_s, R_m, R_e, R_v, R_f$ the different resistances of the R6C2 model, and by c_i, c_m the capacities of the inner rooms and the walls. We denote by γ the proportion of heating dissipated in the wall through conduction, and by $(1 - \gamma)$ the proportion of heating dissipated in the inner room through convection. We detail the numerical values in Table 4.

R_i	4.81×10^{-4}	SI
R_s	2.94×10^{-4}	SI
R_m	4.51×10^{-3}	SI
R_e	1.48×10^{-4}	SI
R_v	4.51×10^{-3}	SI
R_f	2.00×10^{-2}	SI
c_i	8.30×10^7	SI
c_m	5.85×10^6	SI

Table 4: Numerical values for the electrical analogy

6.4 MPC

Building offline an AR model for MPC We fit an AR(1) model using the optimization scenarios (we do not consider higher order lags for the sake of simplicity). For $i \in \{el, hw\}$, the AR model writes

$$d_{t+1}^i = \alpha_t^i d_t^i + \beta_t^i + \varepsilon_t^i, \quad (25a)$$

where the nonstationary coefficients (α_t^i, β_t^i) are, for any time step t , solutions of the least-square problem

$$(\alpha_t^i, \beta_t^i) = \arg \min_{a, b} \sum_{s=1}^{N_{opt}} \|d_{t+1}^{i,s} - a d_t^{i,s} - b\|_2^2. \quad (25b)$$

The points $(d_t^{i,1}, \dots, d_t^{i,N_{opt}})$ correspond to the optimization scenarios. The AR residuals $(\varepsilon_t^{el}, \varepsilon_t^{hw})$ are a white noise process.

Updating the forecast online Once the AR model is calibrated, we use it to update the forecast during assessment (see §3.1). The update procedure is threefold:

- i) we observe the demands $w_t = (d_t^{el}, d_t^{hw})$ between time steps $t - 1$ and t ,
- ii) we update the forecast \bar{w}_{t+1} at time step $t + 1$ with the AR model

$$\bar{w}_{t+1} = (\bar{d}_{t+1}^{el}, \bar{d}_{t+1}^{hw}) = \left(\alpha_t^{el} d_t^{el} + \beta_t^{el}, \alpha_t^{hw} d_t^{hw} + \beta_t^{hw} \right),$$

- iii) we set the forecast between time steps $t + 2$ and T by using the mean values

$$\bar{w}_\tau = \frac{1}{N_{opt}} \sum_{i=1}^{N_{opt}} w_\tau^i \quad \forall \tau = t + 2, \dots, T$$

of the optimization scenarios

Once the forecast $(\bar{w}_{t+1}, \dots, \bar{w}_T)$ is available, it serves as input into the optimization Problem (15) (the MPC algorithm).