Distributed Multistage Optimization of Large-Scale Microgrids under Stochasticity

François Pacaud* Michel De Lara† Jean-Philippe Chancelier† Pierre Carpentier‡

June 17, 2021

Abstract

Microgrids are recognized as a relevant tool to absorb decentralized renewable energies in the energy mix. However, the sequential handling of multiple stochastic productions and demands, and of storage, make their management a delicate issue. We add another layer of complexity by considering microgrids where different buildings stand at the nodes of a network and are connected by the arcs; some buildings host local production and storage capabilities, and can exchange with others their energy surplus. We formulate the problem as a multistage stochastic optimization problem, corresponding to the minimization of the expected temporal sum of operational costs, while satisfying the energy demand at each node, for all time. The resulting mathematical problem has a large-scale nature, exhibiting both spatial and temporal couplings. However, the problem displays a network structure that makes it amenable to a mix of spatial decomposition-coordination with temporal decomposition methods. We conduct numerical simulations on microgrids of different sizes and topologies, with up to 48 nodes and 64 state variables. Decomposition methods are faster and provide more efficient policies than a state-of-the-art Stochastic Dual Dynamic Programming algorithm. Moreover, they scale almost linearly with the state dimension, making them a promising tool to address more complex microgrid optimal management problems.

1 Introduction

1.1 Problem statement

Power networks are organized more and more in a decentralized fashion, with microgrids coordinating the production of local renewable energies integrated with distributed storage. A broad overview of the emergence of consumer-centric electricity markets is given in [21], and challenges associated with the integration of renewable energy can be found in [12].

As renewable energies, like sun and wind, are stochastic, the Energy Management System (EMS) problem can naturally be formulated as a multistage stochastic optimization problem [24]. If we consider a microgrid consisting of ten buildings each equipped with a hot water tank and a battery, controlled every quarter of an hour during one day, such a problem is already large-scale (a hundred stages and a state with dimension twenty) and a direct resolution is out of reach. The large-scale nature of the EMS problem makes spatial decomposition methods appealing [13].

Different distributed variants of the Model Predictive Control algorithm have been proposed to control microgrids [22]. Decomposition methods are also adapted to the resolution of large-scale unit-commitment problems [1, 15]. We refer the reader to [11] for a recent survey of distributed optimization methods for electric power systems, and to [9] for an example of a distributed optimization algorithm applied to the control of a large power network. When the system is dynamical, as is the case with storage, temporal decomposition methods are also appealing. Stochastic Dynamic Programming (SDP) [4] is the reference

*MCS, Argonne National Laboratory, Lemont, USA
†CERMICS, Ecole des Ponts, Marne-la-Vallée, France
‡UMA, ENSTA Paris, IP Paris, France

algorithm but is hampered by the well-known curse of dimensionality when the state dimension exceeds five. Stochastic Dual Dynamic Programming (SDDP) takes the relay under linear-convex assumptions with nice results, for example in dam management [20], but is efficient up to a certain state dimension.

However, despite limitations inherent to SDP, one can go further by mixing decomposition methods together. For example, recent developments have mixed, in a stochastic setting, spatial decomposition methods with SDP to effectively solve large-scale multistage stochastic optimization problems, by means of the so-called Dual Approximate Dynamic Programming (DADP) algorithm [3]. In this paper, the EMS problems that we consider display a structure that makes them amenable to a mix of spatial and temporal decomposition methods, as developed previously in [5,14] for general coupling constraints. Indeed the (global) problem is naturally formulated as a sum of local multistage stochastic optimization subproblems coupled together via the global network constraints (flow conservation on the graph).

1.2 Contributions

The contributions of this article are threefold. i) We implement DADP in an extended framework that supports generic coupling constraints, specified on a directed graph (whereas the previous implementations of DADP [6] considered only a unique central coupling constraints, or coupling constraints formulated on a tree). ii) We implement a new algorithm called *Primal Approximate Dynamic Programming* (PADP), based on resource decomposition. iii) Thus equipped, on the one hand, we readily compute an exact upper bound (PADP) and an exact lower bound (DADP) of the global minimization problem and, on the other hand, we yield two online control policies implementable by the EMS. We provide numerical comparisons with a state-of-the-art SDDP algorithm, and we show the effectiveness of the two decomposition algorithms: for problems with more than 12 nodes, the decomposition algorithms converge faster than SDDP and yield control policies with lower costs.

1.3 Structure of the paper

The paper is organized as follows. In Sect. 2, we outline the class of optimal energy management problems that we address, and in Sect. 3 we present the associated mathematical formulation. In Sect. 4, we detail how to design algorithms by a mix of spatial decomposition and of SDP. In Sect. 5, we present the results of numerical simulations for different microgrids of increasing size and complexity.

2 Management of large-scale microgrids

Efficacity is the French urban Energy Transition Institute (ITE) devoted to develop and implement innovative solutions to build and manage energy-efficient cities. It was created in 2014 with both the French government support and contributions from companies, small and large. Efficacity has solicited us to address the optimal energy management of urban electrical microgrids, proposing several microgrid configurations, both in terms of topology (structure of the districts) and in terms of equipment (energy production and storage).

Buildings are heterogeneous: all are equipped with an electrical hot water tank, but only some have solar panels and some others have batteries. Indeed, as batteries and solar panels are expensive, they are shared out across the network. All units have the possibility to import and export energy to and from the other buildings. Moreover, we suppose that, if the local production is unable to fulfill the local demand, even after exchanges between buildings, energy can be imported from an external (regional or national) grid as a recourse. Thus, each building is a decision unit able to locally consume, produce, store, and also to exchange energy with other units and with the external grid. The flows in the microgrid are impacted by uncertainties, both in demand (e.g. electrical) and in production (e.g. solar panels). We suppose that all actors are benevolent, allowing a central planner (namely the EMS) to coordinate the local units. The EMS aims at satisfying the balance between production and demand at each node (building), but at least cost.

We manage the microgrids over one day, with decisions taken every 15 min. The local solar energy productions match realistic data corresponding to a summer day in Paris. The local demands are generated

using a stochastic simulator experimentally validated [18].

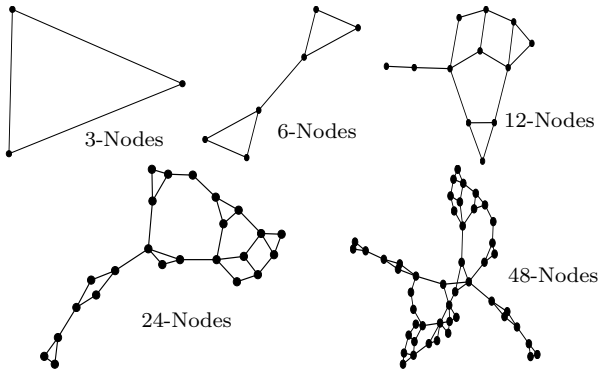


Figure 1: Examples of microgrid topologies

We consider five different microgrids with growing sizes and different topologies, that is, different nodes and connecting arcs. The structure of the microgrids (see Figure 1) as well as the location of batteries and solar panels come from case studies provided by Efficacity. As an example (see Table 1), the 12-Nodes problem consists of 12 buildings; 4 buildings are equipped with a 3 kWh battery, and 4 other buildings are equipped with 16 m^2 of solar panels. The devices are dispatched so that a building equipped with a solar panel is connected to at least one building with a battery.

3 Mathematical formulation

We now address the mathematical formulation of a multistage stochastic optimization problem that corresponds to satisfying the supply-demand constraint at each node, at least expected cost. After describing the basic mathematical objects in §3.1, we write mathematical equations related to arcs in §3.2, to nodes in §3.3, and finally present the multistage stochastic optimization problem formulation in §3.4.

3.1 Network, stages and probability space

We represent a district microgrid by a directed graph $(\mathcal{N}, \mathcal{A})$, with $\mathcal{N}$ the set of nodes and $\mathcal{A}$ the set of arcs. We denote by $|\mathcal{N}|$ the number of nodes, and by $|\mathcal{A}|$ the number of arcs. We suppose that decisions are made at regular stages, labeled by $t \in [\![0, T]\!]$, where $T \in \mathbb{N}^\star$ is a finite horizon, and where we use the notation $[\![r, s]\!] = \{r, r+1, \ldots, s-1, s\}$ for two integers $r \leq s$. We have $T = 96$ for a daily management with decisions taken every 15 min.

We introduce a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, denote the mathematical expectation by $\mathbb{E}$, and write all random variables with uppercase bold letters.

3.2 Equations related to arcs

3.2.1 Exchanging flows through arcs

during the time interval $[t, t+1)$, each arc $a \in \mathcal{A}$ transports an energy flow $\boldsymbol{Q}_t^a \in \mathbb{R}$, and each node $n \in \mathcal{N}$ imports or exports a flow $\boldsymbol{F}_t^n \in \mathbb{R}$. The node flows $\boldsymbol{F}_t^n$ and the arc flows $\boldsymbol{Q}_t^a$ are related via a balance equation (Kirchhoff's current law) written in matrix form as $C\boldsymbol{Q}_t + \boldsymbol{F}_t = 0$, where $\boldsymbol{F}_t = (\boldsymbol{F}_t^n)_{n \in \mathcal{N}} \in \mathbb{R}^{\mathcal{N}}$ is the family of node flows at time t, $\boldsymbol{Q}_t = (\boldsymbol{Q}_t^a)_{a \in \mathcal{A}} \in \mathbb{R}^{\mathcal{A}}$ is the family of arc flows at time t and where $C \in \{-1, 0, 1\}^{\mathcal{N} \times \mathcal{A}}$ is the node-arc incidence matrix of the directed graph $(\mathcal{N}, \mathcal{A})$. We identify $\boldsymbol{F}_t$ with a vector in $\mathbb{R}^{|\mathcal{N}|}$ and $\boldsymbol{Q}_t$ with a vector in $\mathbb{R}^{|\mathcal{A}|}$, C being identified with a matrix with $|\mathcal{N}|$ rows and $|\mathcal{A}|$ columns.

We denote by $\boldsymbol{F} = (\boldsymbol{F}_t)_{t\in[\![0,T-1]\!]} \in \mathbb{R}^{T\cdot|\mathcal{N}|}$ the vector of node flows and by $\boldsymbol{Q} = (\boldsymbol{Q}_t)_{t\in[\![0,T-1]\!]} \in \mathbb{R}^{T\cdot|\mathcal{A}|}$ the vector of arc flows.

3.2.2 Transportation cost on arcs

a quadratic cost $l_t^a(\boldsymbol{Q}_t^a) = c_2^a(\boldsymbol{Q}_t^a)^2 + c_1^a\boldsymbol{Q}_t^a + c_0^a$ (with given coefficients $c_2^a > 0, c_1^a, c_0^a$, so that each function l_t^a is strongly convex) is associated with transporting the flow $\boldsymbol{Q}_t^a$ through arc $a \in \mathcal{A}$. Such costs can arise from a difference in pricing, a fixed toll between the different nodes, or by energy losses through the network. We aggregate, in the *global arc cost*

$$J_{\mathcal{A}}(\boldsymbol{Q}) = \mathbb{E}\Big[\sum_{a\in\mathcal{A}}\sum_{t=0}^{T-1} l_t^a(\boldsymbol{Q}_t^a)\Big] , \tag{1}$$

all transport costs over all arcs in the graph, over all times $t \in [\![0, T-1]\!]$ and over all random outcomes (hence the mathematical expectation term $\mathbb{E}$).

3.3 Equations related to nodes

We adopt a state space formalism to describe the physical equations related to each node.

3.3.1 State, control and uncertainty variables

let $\{\mathbb{X}_t^n\}_{t\in[\![0,T]\!]}$, $\{\mathbb{U}_t^n\}_{t\in[\![0,T-1]\!]}$ and $\{\mathbb{W}_t^n\}_{t\in[\![1,T]\!]}$ be sequences of Euclidean spaces of type $\mathbb{R}^p$, with appropriate dimensions p (possibly depending on time t and node $n \in \mathcal{N}$).

As all buildings hold a hot water tank, and some possibly also hold a battery, the nodal state $\boldsymbol{X}_t^n$ has dimension 1 or 2. If the building at node n hosts both a hot water tank and a battery, the state is $\boldsymbol{X}_t^n = (\boldsymbol{B}_t^n, \boldsymbol{H}_t^n)$ with values in $\mathbb{X}_t^n = \mathbb{R}^2$, where $\boldsymbol{B}_t^n$ (resp. $\boldsymbol{H}_t^n$) is the energy level inside the battery (resp. hot water tank) at the beginning of the time interval $[t, t+1)$; if the building at node n only hosts a hot water tank, the state is $\boldsymbol{X}_t^n = \boldsymbol{H}_t^n$ with values in $\mathbb{X}_t^n = \mathbb{R}$. The state at time 0 is supposed to be deterministic and known, equal to x_0^n.

In the same way, the nodal control $\boldsymbol{U}_t^n$ has dimension 3 or 2. If the building at node n hosts a battery, the control is $\boldsymbol{U}_t^n = (\boldsymbol{U}_t^{b,n}, \boldsymbol{U}_t^{t,n}, \boldsymbol{U}_t^{ne,n})$ with values in $\mathbb{U}_t^n = \mathbb{R}^3$, where $\boldsymbol{U}_t^{b,n}$ (resp. $\boldsymbol{U}_t^{t,n}$) is the amount of energy flowing into the battery (resp. hot water tank), and $\boldsymbol{U}_t^{ne,n}$ is the amount of electricity exchanged with the external grid, during the time interval $[t, t+1)$; otherwise, the control is $\boldsymbol{U}_t^n = (\boldsymbol{U}_t^{t,n}, \boldsymbol{U}_t^{ne,n})$ with values in $\mathbb{U}_t^n = \mathbb{R}^2$.

Insofar, as the inhabitants of the different buildings have different lifestyles, we suppose that each building has its own electrical and domestic hot water demand profiles, and possibly its own solar panel production. At node $n \in \mathcal{N}$, the uncertainty $\boldsymbol{W}_{t+1}^n = (\boldsymbol{D}_{t+1}^{hw,n}, \boldsymbol{D}_{t+1}^{el,n}) \in \mathbb{W}_{t+1}^n = \mathbb{R}^2$ is made of the domestic hot water demand $\boldsymbol{D}_{t+1}^{hw,n}$ and of $\boldsymbol{D}_{t+1}^{el,n}$, the local electricity demand minus the production of the solar panel (if any), both during the time interval $[t, t+1)$.

3.3.2 Dynamics inside each node

we detail the dynamics in a building at node $n \in \mathcal{N}$. We model the temporal evolution of a battery with the linear dynamics: $\forall t \in [\![0, T-1]\!]$,[1]

$$\boldsymbol{B}_{t+1}^n = \alpha_b \boldsymbol{B}_t^n + \Delta T\Big(\rho_c(\boldsymbol{U}_t^{b,n})^+ - \frac{1}{\rho_d}(\boldsymbol{U}_t^{b,n})^-\Big) , \tag{2a}$$

where α_b is the auto-discharge rate and (ρ_d, ρ_c) are given yields. We model the temporal evolution of an electrical hot water tank with the linear dynamics: $\forall t \in [\![0, T-1]\!]$,

$$\boldsymbol{H}_{t+1}^n = \alpha_h \boldsymbol{H}_t^n + \Delta T\big(\beta_h \boldsymbol{U}_t^{t,n} - \boldsymbol{D}_{t+1}^{hw,n}\big) , \tag{2b}$$

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

where α_h is a discharge rate corresponding to losses by conduction and β_h is a conversion coefficient.

We gather equations (2a)-(2b) in a *nodal dynamics* function $g_t^n : \mathbb{X}_t^n \times \mathbb{U}_t^n \times \mathbb{W}_{t+1}^n \to \mathbb{X}_{t+1}^n$, stating that at each time t, the next state $\boldsymbol{X}_{t+1}^n$ depends on the current state $\boldsymbol{X}_t^n$, the current decision $\boldsymbol{U}_t^n$ and the uncertainty $\boldsymbol{W}_{t+1}^n$ occurring between time t and $t+1$.

3.3.3 Load balance

the load balance between production and demand at node $n \in \mathcal{N}$ writes $\boldsymbol{U}_t^{ne,n} - \boldsymbol{D}_{t+1}^{el,n} - \boldsymbol{U}_t^{b,n} - \boldsymbol{U}_t^{t,n} = \boldsymbol{F}_t^n$, where we recall that $\boldsymbol{F}_t^n$ is the energy exchanged with the adjacent nodes. Thus, for each time $t \in [\![0, T-1]\!]$ and node $n \in \mathcal{N}$, we introduce the *nodal load balance function*

$$\Delta_t^n(\boldsymbol{X}_t^n, \boldsymbol{U}_t^n, \boldsymbol{W}_{t+1}^n) = \boldsymbol{U}_t^{ne,n} - \boldsymbol{D}_{t+1}^{el,n} - \boldsymbol{U}_t^{b,n} - \boldsymbol{U}_t^{t,n} \ . \tag{3}$$

If $\Delta_t^n(\cdot) < 0$ (resp. $\Delta_t^n(\cdot) > 0$), the node n imports (resp. exports) energy from (resp. to) adjacent nodes

3.3.4 Cost function

during the time interval $[t, t+1)$ and at node $n \in \mathcal{N}$, the cost

$$L_t^n(\boldsymbol{X}_t^n, \boldsymbol{U}_t^n, \boldsymbol{W}_{t+1}^n) = p_t^{el} \boldsymbol{U}_t^{ne,n} \tag{4}$$

depends linearly on the price p_t^{el} to import electricity from the external grid. When we sum over time, we add a final penalization term $K^n(\boldsymbol{X}_T^n)$ to avoid an empty electrical hot water tank at the end of the day.

3.4 Multistage stochastic optimization problem formalization

We write $\boldsymbol{F}^n = (\boldsymbol{F}_0^n, \cdots, \boldsymbol{F}_{T-1}^n)^\top$ the node flow process arriving at each node $n \in \mathcal{N}$ between times 0 and $T-1$. We call optimal nodal cost the expression

$$\begin{aligned}
&J_{\mathcal{N}}^n(\boldsymbol{F}^n, x_0^n) = \\
&\quad \min_{\boldsymbol{X}^n, \boldsymbol{U}^n} \ \mathbb{E}\left[\sum_{t=0}^{T-1} L_t^n(\boldsymbol{X}_t^n, \boldsymbol{U}_t^n, \boldsymbol{W}_{t+1}^n) + K^n(\boldsymbol{X}_T^n) \right] , && \text{(5a)} \\
&\quad \text{s.t.} \quad \forall t \in [\![0, T-1]\!] \\
&\qquad \boldsymbol{X}_{t+1}^n = g_t^n(\boldsymbol{X}_t^n, \boldsymbol{U}_t^n, \boldsymbol{W}_{t+1}^n) \ , \ \ \boldsymbol{X}_0^n = x_0^n \ , && \text{(5b)} \\
&\qquad \Delta_t^n(\boldsymbol{X}_t^n, \boldsymbol{U}_t^n, \boldsymbol{W}_{t+1}^n) = \boldsymbol{F}_t^n \ , && \text{(5c)} \\
&\qquad \sigma(\boldsymbol{U}_t^n) \subset \sigma(\boldsymbol{W}_1, \cdots, \boldsymbol{W}_t, \boldsymbol{W}_{t+1}) \ , && \text{(5d)}
\end{aligned}$$

where $\boldsymbol{X}^n = (\boldsymbol{X}_0^n, \cdots, \boldsymbol{X}_T^n)$, $\boldsymbol{U}^n = (\boldsymbol{U}_0^n, \cdots, \boldsymbol{U}_{T-1}^n)$ and $\boldsymbol{W}^n = (\boldsymbol{W}_0^n, \cdots, \boldsymbol{W}_T^n)$ are respectively the *local* state (stocks), control (production) and uncertainty (consumption) processes. Similarly, we denote by $\boldsymbol{W} = (\boldsymbol{W}^n)_{n \in \mathcal{N}}$ the *global* uncertainty process.

To be able to almost surely satisfy the load balance equations (5c) at each node $n \in \mathcal{N}$, we assume that all decisions follow the *hazard-decision* information structure, that is, decision $\boldsymbol{U}_t^n$ is taken after the global uncertainty $\boldsymbol{W}_{t+1} = (\boldsymbol{W}_{t+1}^n)_{n \in \mathcal{N}}$ has been observed, hence the specific form of the nonanticipativity constraint (5d), where $\sigma(\boldsymbol{U}_t^n)$ denotes the σ-algebra generated by the random variable $\boldsymbol{U}_t^n$.The multistage nature of Problem (5) stems from the nonanticipativity constraint (5d). Indeed, (5d) ensures that the decisions taken at time t depend *only* on the previous global uncertainties or, alternatively, that the solution of (5) is given as a sequence of *policies* $\gamma_1^n, \ldots, \gamma_{T-1}^n$, such that, for all $t = 1, \ldots, T-1$, $\gamma_t^n : \mathbb{W}_1 \times \cdots \times \mathbb{W}_{t+1} \to \mathbb{U}_t$ and $\boldsymbol{U}_t^n = \gamma_t^n(\boldsymbol{W}_1, \ldots, \boldsymbol{W}_{t+1})$.

We have stated a global arc criterion in (1) and local nodal criteria in (5), both depending on node and arc flows coupled by Kirchhoff's current law $C\boldsymbol{Q}_t + \boldsymbol{F}_t = 0$, at each time $t \in [\![0, T-1]\!]$. We rewrite these constraints globally as $\mathcal{C}\boldsymbol{Q} + \boldsymbol{F} = 0$ involving the global node flow and arc flow processes $\boldsymbol{Q}$ and $\boldsymbol{F}$, and where the matrix $\mathcal{C} \in \mathbb{R}^{T \cdot |\mathcal{N}|} \times \mathbb{R}^{T \cdot |\mathcal{A}|}$ is a block-diagonal matrix with matrix C as diagonal element.

We set $\mathbb{X}_0 = \prod_{n\in\mathcal{N}} \mathbb{X}_0^n$ and, for any $x_0 = (x_0^n)_{n\in\mathcal{N}} \in \mathbb{X}_0$, we formulate the *global* optimization problem of the central manager (EMS) as

$$V_0\sharp(x_0) = \min_{\boldsymbol{F},\boldsymbol{Q}} \sum_{n\in\mathcal{N}} J_{\mathcal{N}}^n(\boldsymbol{F}^n, x_0^n) + J_{\mathcal{A}}(\boldsymbol{Q}) \tag{6a}$$

$$\text{s.t. } \mathcal{C}\boldsymbol{Q} + \boldsymbol{F} = 0 \ . \tag{6b}$$

The Problem (6) couples $|\mathcal{N}| + 1$ independent criteria through Constraint (6b). As the resulting criterion is additive and Constraint (6b) is affine, Problem (6) has a nice form to use decomposition-coordination methods.

4 Resolution by distributed optimization

As just detailed in Sect. 3, the global Problem (6) encompasses a family of local multistage stochastic optimization subproblems, coupled together via a transportation problem corresponding to the flows exchanged through the graph. We now detail how to solve (6) in a distributed fashion.

In §4.1, we decouple (6) node by node using either price or resource decomposition schemes. In §4.2, we show how to find the most appropriate deterministic price and resource processes. Thus, we obtain two algorithms, each of them yielding nodal value functions and, from these latter, upper and lower bounds for the optimal cost and online control policies.

4.1 Mixing nodal and time decomposition

In [5], we introduced a generic framework to bound a global problem by decomposing it into smaller local subproblems, easier to solve. Problem (6) lies in the generic framework introduced in [5], and the coupling equation $\mathcal{C}\boldsymbol{Q} + \boldsymbol{F} = 0$ is a special case of the generic coupling constraint of this framework. Thus, to solve Problem (6), we first apply *spatial decoupling* into nodal and arc subproblems, and then apply the *temporal decomposition* induced by Dynamic Programming.

4.1.1 Price decomposition of the global problem

we follow the procedure introduced in [5, §2.2] to solve Problem (6) by price decomposition and to provide a lower bound of its optimal value $V_0\sharp(x_0)$. In the case under study, price decomposition follows from the dualization of Constraint (6b) using a deterministic price coordination process $p = (p^n)_{n\in\mathcal{N}} \in \mathbb{R}^{T\cdot|\mathcal{N}|}$ as multiplier. We define the *global price value function*[2] $\underline{V}[p] : \mathbb{X}_0 \to \mathbb{R}$ associated with Problem (6) by the following expression, for all $x_0 = (x_0^n)_{n\in\mathcal{N}} \in \mathbb{X}_0$,

$$\begin{aligned} r l \underline{V}[p](x_0) = \quad & \min_{\boldsymbol{F},\boldsymbol{Q}} \ \textstyle\sum_{n\in\mathcal{N}} J_{\mathcal{N}}^n(\boldsymbol{F}^n, x_0^n) + J_{\mathcal{A}}(\boldsymbol{Q}) \\ & + \mathbb{E}\big[\big\langle p \,, \mathcal{C}\boldsymbol{Q} + \boldsymbol{F}\big\rangle\big] \ . \end{aligned} \tag{7}$$

We observe in a straightforward manner that

$$\underline{V}[p](x_0) = \sum_{n\in\mathcal{N}} \underline{V}_{\mathcal{N}}^n[p^n](x_0^n) + \underline{V}_{\mathcal{A}}[p] \ , \tag{8}$$

that is, the global price value function $\underline{V}[p](\cdot)$ naturally decomposes into a family of *nodal price value functions* $\underline{V}_{\mathcal{N}}^n[p^n] : \mathbb{X}_0^n \to \mathbb{R}$, $\forall n \in \mathcal{N}$, given by

$$\underline{V}_{\mathcal{N}}^n[p^n](x_0^n) = \min_{\boldsymbol{F}^n} J_{\mathcal{N}}^n(\boldsymbol{F}^n, x_0^n) + \mathbb{E}\big[\big\langle p^n \,, \boldsymbol{F}^n\big\rangle\big] \ , \tag{9a}$$

[2] In the expression $\underline{V}[p](x_0)$, we use brackets $[p]$ to indicate a parametric dependence, whereas we use parenthesis (x_0) to indicate the argument of the function $\underline{V}[p] : \mathbb{X}_0 \to \mathbb{R}$. We also use the notation $\underline{V}[p](\cdot)$ to designate this function.

and an *arc price value function*[3] given by

$$\underline{V}_{\mathcal{A}}[p] = \min_{\boldsymbol{Q}} \; J_{\mathcal{A}}(\boldsymbol{Q}) + \mathbb{E}\big[\big\langle \mathcal{C}^\top p \,, \boldsymbol{Q} \big\rangle\big] \; . \tag{9b}$$

What is more, for all $n \in \mathcal{N}$, the optimal value $\underline{V}^n_{\mathcal{N}}[p^n](x^n_0)$ can be computed by Dynamic Programming under the so-called *white noise assumption*.

Assumption 1 *The global uncertainty process $(\boldsymbol{W}_1, \cdots, \boldsymbol{W}_T)$ consists of stagewise independent random variables.*

For all node $n \in \mathcal{N}$ and for any price $p^n \in \mathbb{R}^T$, taking into account the expression (5) of the nodal cost $J^n_{\mathcal{N}}$, we introduce the sequence $\{\underline{V}^n_{\mathcal{N},t}[p^n](\cdot)\}_{t\in[\![0,T]\!]}$ of local price value functions defined by $\underline{V}^n_{\mathcal{N},T}[p^n](\cdot) = K^n(\cdot)$ (final cost) and then, inductively, for all $t \in [\![0,T]\!]$ and $x^n_t \in \mathbb{X}^n_t$, by

$$\begin{aligned}\underline{V}^n_{\mathcal{N},t}[p^n](x^n_t) = \min_{\boldsymbol{X}^n, \boldsymbol{U}^n, \boldsymbol{F}^n} \mathbb{E}\Big[\sum_{s=t}^{T-1} \Big(L^n_s(\boldsymbol{X}^n_s, \boldsymbol{U}^n_s, \boldsymbol{W}^n_{s+1}) \\ + \big\langle p^n_s \,, \boldsymbol{F}^n_s \big\rangle \Big) + K^n(\boldsymbol{X}^n_T) \Big] \; , \\ \text{s.t. } \; \boldsymbol{X}^n_t = x^n_t \; , \;\; \text{(5b)} - \text{(5c)} - \text{(5d)} \; .\end{aligned}$$

Under Assumption 1, these local price value functions satisfy the following Dynamic Programming equations: for all $n \in \mathcal{N}$, $\underline{V}^n_{\mathcal{N},T}[p^n](x^n_T) = K^n(x^n_T)$, and, for $t = T-1, \ldots, 0$,

$$\begin{aligned}\underline{V}^n_{\mathcal{N},t}[p^n](x^n_t) = \mathbb{E}\Big[\min_{u^n_t} L^n_t(x^n_t, u^n_t, \boldsymbol{W}^n_{t+1}) \\ + \big\langle p^n_t \,, \Delta^n_t(x^n_t, u^n_t, \boldsymbol{W}^n_{t+1}) \big\rangle \\ + \underline{V}^n_{\mathcal{N},t+1}[p^n]\big(g^n_t(x^n_t, u^n_t, \boldsymbol{W}^n_{t+1})\big) \Big] \; .\end{aligned} \tag{10}$$

The nodal price value function $\underline{V}^n_{\mathcal{N}}[p^n](\cdot)$ in (9a) is equal to the local price value function at time $t = 0$: $\underline{V}^n_{\mathcal{N}}[p^n](x^n_0) = \underline{V}^n_{\mathcal{N},0}[p^n](x^n_0)$, for all for $x^n_0 \in \mathbb{X}^n_0$.

Considering the expression (1) of the arc cost $J_{\mathcal{A}}(\boldsymbol{Q})$, the arc price value function $\underline{V}_{\mathcal{A}}[p]$ is additive w.r.t. (with respect to) time and space, and thus can be decomposed at each time t and each arc a. The resulting arc subproblems do not involve any time coupling and can be computed by standard mathematical programming tools or even analytically.

4.1.2 Resource decomposition of the global problem

we now solve Problem (6) by resource decomposition (see [5, §2.2]) using a deterministic resource process $r = (r^n)_{n\in\mathcal{N}} \in \mathbb{R}^{T\cdot|\mathcal{N}|}$, such that $r \in \mathrm{im}(\mathcal{C})$.[4] We decompose the global constraint (6b) w.r.t. nodes and arcs as $\boldsymbol{F} = r, \mathcal{C}\boldsymbol{Q} = -r$. We define the *global resource value function* $\overline{V}[r](\cdot)$ associated with Problem (6) by the following expression, for all $x_0 = (x^n_0)_{n\in\mathcal{N}} \in \mathbb{X}_0$:

$$\overline{V}[r](x_0) = \min_{\boldsymbol{F}, \boldsymbol{Q}} \sum_{n\in\mathcal{N}} J^n_{\mathcal{N}}(\boldsymbol{F}^n, x^n_0) + J_{\mathcal{A}}(\boldsymbol{Q}) \; , \tag{11a}$$

$$\text{s.t.} \quad \boldsymbol{F} - r = 0 \; , \;\; \mathcal{C}\boldsymbol{Q} + r = 0 \; . \tag{11b}$$

We observe in a straightforward manner that

$$\overline{V}[r](x_0) = \sum_{n\in\mathcal{N}} \overline{V}^n_{\mathcal{N}}[r^n](x^n_0) + \overline{V}_{\mathcal{A}}[r] \; , \tag{12}$$

[3] Which, to the difference of the nodal price value function (9a), does not depend on the initial state x_0.

[4] If $r \notin \mathrm{im}(\mathcal{C})$, we have $\overline{V}[r] = +\infty$ in (11) as the constraint $\mathcal{C}\boldsymbol{Q} + r = 0$ cannot be satisfied.

that is, the global resource value function $\overline{V}[r](\cdot)$ naturally decomposes into a family of *nodal resource value functions* $\overline{V}^n_{\mathcal{N}}[r^n](\cdot)$ defined, for all $n \in \mathcal{N}$ and $x^n_0 \in \mathbb{X}^n_0$, by

$$\overline{V}^n_{\mathcal{N}}[r^n](x^n_0) = \min_{\boldsymbol{F}^n} \; J^n_{\mathcal{N}}(\boldsymbol{F}^n, x^n_0) \quad \text{s.t.} \quad \boldsymbol{F}^n - r^n = 0 \tag{13a}$$

and an *arc resource value function* (not depending on x_0)

$$\overline{V}_{\mathcal{A}}[r] = \min_{\boldsymbol{Q}} \; J_{\mathcal{A}}(\boldsymbol{Q}) \quad \text{s.t.} \quad \mathcal{C}\boldsymbol{Q} + r = 0 \; . \tag{13b}$$

For all node $n \in \mathcal{N}$, taking into account the expression (5) of the nodal cost $J^n_{\mathcal{N}}$, we introduce the sequence $\{\overline{V}^n_{\mathcal{N},t}[r^n](\cdot)\}_{t\in[\![0,T]\!]}$ of local resource value functions defined by $\overline{V}^n_{\mathcal{N},T}[r^n](\cdot) = K^n(\cdot)$ (final cost) and then, inductively, for all $t \in [\![0,T]\!]$ and $x^n_t \in \mathbb{X}^n_t$, by

$$\begin{aligned}\overline{V}^n_{\mathcal{N},t}[r^n](x^n_t) = \min_{\substack{\boldsymbol{X}^n \\ \boldsymbol{U}^n \\ \boldsymbol{F}^n}} \mathbb{E}\Big[\sum_{s=t}^{T-1} L^n_s(\boldsymbol{X}^n_s, \boldsymbol{U}^n_s, \boldsymbol{W}^n_{s+1}) + K^n(\boldsymbol{X}^n_T) \Big] \\ \text{s.t.} \;\; \boldsymbol{X}^n_t = x^n_t \;, \;\; \text{(5b)} - \text{(5c)} - \text{(5d)} \;, \\ \boldsymbol{F}^n_s - r^n_s = 0 \;, \;\; \forall s \in [\![t, T-1]\!] \;.\end{aligned}$$

If Assumption 1 holds true, $\overline{V}^n_{\mathcal{N}}[r^n](x^n_0)$ can be computed by Dynamic Programming and the nodal resource value function $\overline{V}^n_{\mathcal{N}}[r^n](\cdot)$ in (13a) is equal to the local resource value function at time $t = 0$: $\overline{V}^n_{\mathcal{N}}[r^n](x^n_0) = \overline{V}^n_{\mathcal{N},0}[r^n](x^n_0)$ for all $x^n_0 \in \mathbb{X}^n_0$. In the case of resource decomposition, arcs are coupled through the constraint $\mathcal{C}\boldsymbol{Q} + r = 0$, so that the arc resource value function $\overline{V}_{\mathcal{A}}[r]$ in (13b) is not additive in space, but remain additive w.r.t. time. As in price decomposition, it can be computed by standard mathematical programming tools or even analytically.

4.1.3 Upper and lower bounds of the global problem

applying [5, Proposition 2] to the global price value function (7) and resource value functions (11), we are able to bound up and down the optimal value $V_0\sharp(x_0)$ of Problem (6) as follows:

$$\underline{V}[p](x_0) \leq V_0\sharp(x_0) \leq \overline{V}[r](x_0) \;, \;\; \forall x_0 \in \prod_{n\in\mathcal{N}} \mathbb{X}^n_0 \;. \tag{14}$$

These inequalities hold true for any price process $p \in \mathbb{R}^{T\cdot|\mathcal{N}|}$ and for any resource process $r \in \mathrm{im}(\mathcal{C}) \subset \mathbb{R}^{T\cdot|\mathcal{N}|}$.

4.2 Algorithmic implementation

In §4.1, we have decomposed Problem (6) spatially and temporally: the global problem is now split into (small) subproblems using price and resource decompositions, and each subproblem is solved by Dynamic Programming. These decompositions yield bounds for the value of the global problem. To obtain tighter bounds for the optimal value in (14), we follow the approach presented in [5, §3.2], that is, we maximize (resp. minimize) the left-hand side (resp. the right-hand side) in (14) w.r.t. the price vector $p \in \mathbb{R}^{T\cdot|\mathcal{N}|}$ (resp. the resource vector $r \in \mathbb{R}^{T\cdot|\mathcal{N}|}$), using a gradient-like algorithm.

4.2.1 Lower bound improvement

we detail how to improve the lower bound given by the price value function in (14). We fix $x_0 = (x^n_0)_{n\in\mathcal{N}} \in \mathbb{X}_0$, and we aim at solving $\max_{p\in\mathbb{R}^{T\cdot|\mathcal{N}|}} \underline{V}[p](x_0)$, that is, written equivalently (see (7))

$$\max_{p\in\mathbb{R}^{T\cdot|\mathcal{N}|}} \; \min_{\boldsymbol{F},\boldsymbol{Q}} \; \sum_{n\in\mathcal{N}} J^n_{\mathcal{N}}(\boldsymbol{F}^n, x^n_0) + J_{\mathcal{A}}(\boldsymbol{Q}) + \big\langle p \,, \mathbb{E}\big[\mathcal{C}\boldsymbol{Q} + \boldsymbol{F}\big] \big\rangle \;. \tag{15}$$

We solve the maximization Problem (15) w.r.t. p using a gradient ascent method (Uzawa algorithm). At iteration k, we suppose given a deterministic price process $p^{(k)}$ and a gradient step $\rho^{(k)}$. The algorithm proceeds as follows, $\forall n \in \mathcal{N}$,

$$\boldsymbol{F}^{n\,(k+1)} \in \arg\min_{\boldsymbol{F}^n} J^n_{\mathcal{N}}(\boldsymbol{F}^n, x^n_0) + \mathbb{E}\big[\big\langle p^{n\,(k)}\,, \boldsymbol{F}^n \big\rangle\big]\,, \tag{16a}$$

$$\boldsymbol{Q}^{(k+1)} \in \arg\min_{\boldsymbol{Q}} J_{\mathcal{A}}(\boldsymbol{Q}) + \mathbb{E}\big[\big\langle \mathcal{C}^\top p^{(k)}\,, \boldsymbol{Q} \big\rangle\big]\,, \tag{16b}$$

$$p^{(k+1)} = p^{(k)} + \rho^{(k)}\ \mathbb{E}\big[\mathcal{C}\boldsymbol{Q}^{(k+1)} + \boldsymbol{F}^{(k+1)}\big]\,. \tag{16c}$$

At each iteration k, updating $p^{(k)}$ requires the computation of the gradient of $\nabla_p \underline{V}[p^{(k)}](x_0)$, that is, the expected value $\mathbb{E}\big[\mathcal{C}\boldsymbol{Q}^{(k+1)} + \boldsymbol{F}^{(k+1)}\big]$, usually estimated by a Monte-Carlo method. The price update formula (16c) — corresponding to the standard gradient algorithm for the maximization w.r.t. p in Problem (15) — can be replaced by more sophisticated algorithms (e.g. quasi-Newton).

4.2.2 Upper bound improvement

we now focus on the improvement of the upper bound given by the global resource value function in (14). We fix $x_0 = (x^n_0)_{n\in\mathcal{N}} \in \prod_{n\in\mathcal{N}} \mathbb{X}^n_0$, and we aim at solving $\min_{r\in\mathrm{im}(\mathcal{C})}\ \overline{V}[r](x_0)$, whose detailed expression is (see (11)):

$$\min_{r\in\mathrm{im}(\mathcal{C})} \Bigg(\sum_{n\in\mathcal{N}} \Big(\min_{\boldsymbol{F}^n}\ J^n_{\mathcal{N}}(\boldsymbol{F}^n, x^n_0) \quad \text{s.t.} \quad \boldsymbol{F}^n - r^n = 0\Big) + \Big(\min_{\boldsymbol{Q}}\ J_{\mathcal{A}}(\boldsymbol{Q}) \quad \text{s.t.} \quad \mathcal{C}\boldsymbol{Q} + r = 0\Big)\Bigg)\,. \tag{17}$$

We now sketch how we solve the minimization problem (17) using a gradient-like method. As we recognize in (17) the minimization problems (13a) and (13b), we obtain that the gradients w.r.t. r in (17) can be expressed as mathematical expectations $\mu^n = \mathbb{E}[\boldsymbol{M}^n]$ and $\xi = \mathbb{E}[\boldsymbol{\Xi}]$, where we have denoted by $\boldsymbol{M}^n$ the optimal multiplier associated with the constraint $\boldsymbol{F}^n - r^n = 0$ in (13a), and by $\boldsymbol{\Xi}$ the optimal multiplier associated with the constraint $\mathcal{C}\boldsymbol{Q} + r = 0$ in (13b).

The algorithm proceeds as follows. At each iteration k, the algorithm updates the resource $r^{(k)}$ and the gradient step $\rho^{(k)}$. We solve the optimization problems (13a) and (13b) with the resource set to $r^{(k)}$ and obtain the optimal solutions together with the associated multipliers $\mu^{n,(k)}$ and $\xi^{(k)}$ as described in the previous paragraph. Then, the resource $r^{(k+1)}$ is updated by

$$r^{(k+1)} = \mathrm{proj}_{\mathrm{im}(\mathcal{C})}\Big(r^{(k)} - \rho^{(k)}\ \big(\mu^{(k+1)} + \xi^{(k+1)}\big)\Big)\,, \tag{18}$$

where $\mathrm{proj}_{\mathrm{im}(\mathcal{C})}$ is the orthogonal projection onto the subspace $\mathrm{im}(\mathcal{C})$ and $\mu^{(k+1)} = \{\mu^{n,(k+1)}\}_{n\in\mathcal{N}}$. The multipliers μ^n and ξ are approximated using a Monte Carlo method. Again, the above projected gradient algorithm, used to update the resource, can be replaced by any gradient-based constrained optimization algorithm.

4.2.3 Upper and lower nodal value functions

The two algorithms (16) and (18) converge respectively to a price process p^∞ and to a resource process $r^\infty \in \mathrm{im}(\mathcal{C})$, parameterized by a *fixed* initial state. However, by applying the two inequalities (14) with both *admissible* processes p^∞ and r^∞, we are able to bound the optimal value function $V_0\sharp(\cdot)$ globally: $\underline{V}[p^\infty](x_0) \le V_0\sharp(x_0) \le \overline{V}[r^\infty](x_0)$ for all $x_0 \in \mathbb{X}_0$.

Problem	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{A}\rvert$	$dim(\mathbb{X}_t)$	$dim(\mathbb{W}_t)$	$supp(\boldsymbol{W}_t)$
3-Nodes	3	3	4	6	10^3
6-Nodes	6	7	8	12	10^6
12-Nodes	12	16	16	24	10^{12}
24-Nodes	24	33	32	48	10^{24}
48-Nodes	48	69	64	96	10^{48}

Table 1: Microgrid management problems (cardinals and dimensions)

5 Numerical simulation results

We will term *Dual Approximate Dynamic Programming* (DADP) the price decomposition algorithm described in §4.2.1 and *Primal Approximate Dynamic Programming* (PADP) the resource decomposition algorithm described in §4.2.2. We compare DADP and PADP with the well-known Stochastic Dual Dynamic Programming (SDDP) algorithm (see [7] and references inside) on the results that they yield on five microgrid optimal management problems with growing sizes: Table 1 displays the features (different sizes and dimensions) of the cases we consider for numerical simulations. For this purpose, we detail the offline computation of value functions in §5.1, the online control policies in §5.2, and we finally summarize the numerical simulation results and compare the three algorithms in §5.3.

For the uncertainties, we generate scenarios for the demands at each node using the generator presented in [2], from which we add the production of the solar panel for the corresponding buildings. Then, we model the process $(\boldsymbol{W}_t^n)_{t,n}$ as nonstationary random variables that are stagewise and node by node probabilistically independent with a finite probability distribution on the set $\mathbb{W}_t^n = \mathbb{R}^2$. For each case, we consider a single initial state x_0, corresponding to standard configurations of the storages (like minimal energy in a battery).

5.1 Computing offline value functions

5.1.1 With the SDDP algorithm

to compute value functions, the SDDP algorithm is not implementable in a straightforward manner. Indeed, even if the cardinality of the support of each local random variable $\boldsymbol{W}_t^n$ remains low, the cardinal of the support of the global uncertainty $\boldsymbol{W}_t$ becomes huge as the number $|\mathcal{N}|$ of nodes grows (see Table 1), so that the exact computation of expectations, as required at each time step during the backward pass of the SDDP algorithm (see [19]), becomes untractable. To overcome this issue, we resample the probability distribution of $(\boldsymbol{W}_t^n)_{n\in\mathcal{N}}$ for each time t, to deal with an uncertainty support of reasonable size, using the k-means clustering method in [17]. As the local problems are convex w.r.t. the uncertainties, by Jensen inequality the optimal quantization yields a new optimization problem whose optimal value is a lower bound for the optimal value of the original problem (see [10] for details). Then, the exact lower bound given by SDDP with resampling remains a lower bound for the exact lower bound given by SDDP without resampling, which itself is by construction a lower bound for the original problem. In the numerical application, we fix the resampling size to 100. We denote by $\{\underline{V}_t^{\text{sddp}}\}_{t\in[\![0,T]\!]}$ the value functions returned by the SDDP algorithm. Notice that, whereas the SDDP algorithm suffers from the cardinality of the global uncertainty support, the DADP and PADP algorithms do not.

We stop SDDP when the gap between its exact lower bound and a statistical upper bound is lower than 1%. That corresponds to the standard SDDP's stopping criterion described in [19], which is reputed to be more consistent than the first stopping criterion introduced in [16]. Our implementation of SDDP uses a level-one cut selection algorithm [8] and keeps only the 100 most relevant cuts. By doing so, we significantly reduce the computation time of SDDP.

5.1.2 With the DADP and PADP algorithms

to optimize the price and resource processes, we use a quasi-Newton method, and more precisely the L-BFGS algorithm (implemented in the nonlinear solver Ipopt 3.12 [23] which allows to explicitly tackle the linear

constraint $r \in \mathrm{im}(\mathcal{C})$ in the resource decomposition algorithm (18)). The algorithms stop at p and r either when a stopping criterion is fulfilled or when no descent direction is found.

5.2 Devising online control policies

Each algorithm (DADP, PADP and SDDP) returns a sequence of value functions indexed by time. Using these value functions, we define a sequence of *surrogate global value functions* $\{\widehat{V}_t\}_{t\in[\![0,T]\!]}$ by[5]

- $\widehat{V}_t = \underline{V}_t^{\mathrm{sddp}}$ for SDDP,
- $\widehat{V}_t = \sum_{n\in\mathcal{N}} \underline{V}_{\mathcal{N},t}^n[p] + \underline{V}_{\mathcal{A},t}[p]$ for DADP,
- $\widehat{V}_t = \sum_{n\in\mathcal{N}} \overline{V}_{\mathcal{N},t}^n[r] + \overline{V}_{\mathcal{A},t}[r]$ for PADP.

With these global value functions, we design online control policies. For any time $t \in [\![0, T-1]\!]$, any global state $x_t \in \mathbb{X}_t$ and global uncertainty $w_{t+1} \in \mathbb{W}_{t+1}$, the control policy is a solution of the following one-step optimization problem:

$$
\begin{aligned}
\gamma_t(x_t, w_{t+1}) \in \arg\min_{u_t} \Big(& \min_{f_t, q_t} \\
& \sum_{n\in\mathcal{N}} L_t^n(x_t^n, u_t^n, w_{t+1}^n) + \sum_{a\in\mathcal{A}} l_t^a(q_t^a) + \widehat{V}_{t+1}(x_{t+1}) \Big) \\
\text{s.t.} \quad & Cq_t + f_t = 0 \,, && \text{(19a)} \\
& x_{t+1}^n = g_t^n(x_t^n, u_t^n, w_{t+1}^n) \,, && \text{(19b)} \\
& \Delta_t^n(x_t^n, u_t^n, w_{t+1}^n) = f_t^n \,, \ \ \forall n \in \mathcal{N} \,. && \text{(19c)}
\end{aligned}
$$

As the policy induced by (19) is admissible for the global problem (6), the expected value of its associated cost is an upper bound of the optimal value $V_0\sharp(x_0)$ of the original minimization problem (6).

5.3 Numerical results

We first compare the three algorithms w.r.t. the convergence and the CPU time needed for computing offline value functions. Second, we compare the values of the theoretical bounds for the optimal expected total cost. Third, we compare online policies simulation results.

5.3.1 Computing offline value functions

we solve Problem (6) by SDDP, price decomposition (DADP) and resource decomposition (PADP). Table 2 details the execution time and number of iterations taken before reaching convergence. For a small-scale

# Nodes $\lvert\mathcal{N}\rvert$	3	6	12	24	48
dim $\mathbb{X}_t$	4	8	16	32	64
SDDP CPU time	1'	3'	10'	79'	453'
SDDP iterations	30	100	180	500	1500
DADP CPU time	6'	14'	29'	41'	128'
DADP iterations	27	34	30	19	29
PADP CPU time	3'	7'	22'	49'	91'
PADP iterations	11	12	20	19	20

Table 2: Convergence results for SDDP, DADP and PADP

problem like 3-Nodes (second column of Table 2), SDDP is faster than DADP and PADP. However, for the 48-Nodes problem (last column of Table 2), *DADP and PADP* are *more than three times faster* than

[5]where the functions $\underline{V}_{\mathcal{A},t}[p]$ (resp. $\overline{V}_{\mathcal{A},t}[r]$) are easily deduced from (9b) (resp. from (13b)).

SDDP. Figure 2 depicts how much CPU time take the different algorithms with respect to the number of state variables of the district. For this case study, we observe that the *CPU time grows almost linearly* w.r.t. the number of nodes for DADP and PADP, whereas it grows exponentially for SDDP. Otherwise stated, decomposition methods scale better than SDDP in terms of CPU time for large microgrids instances.

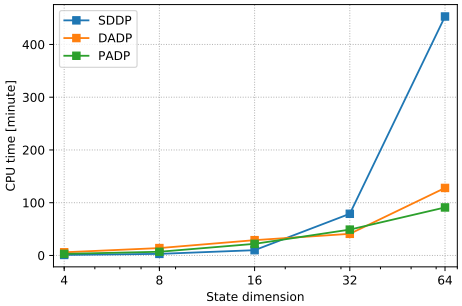
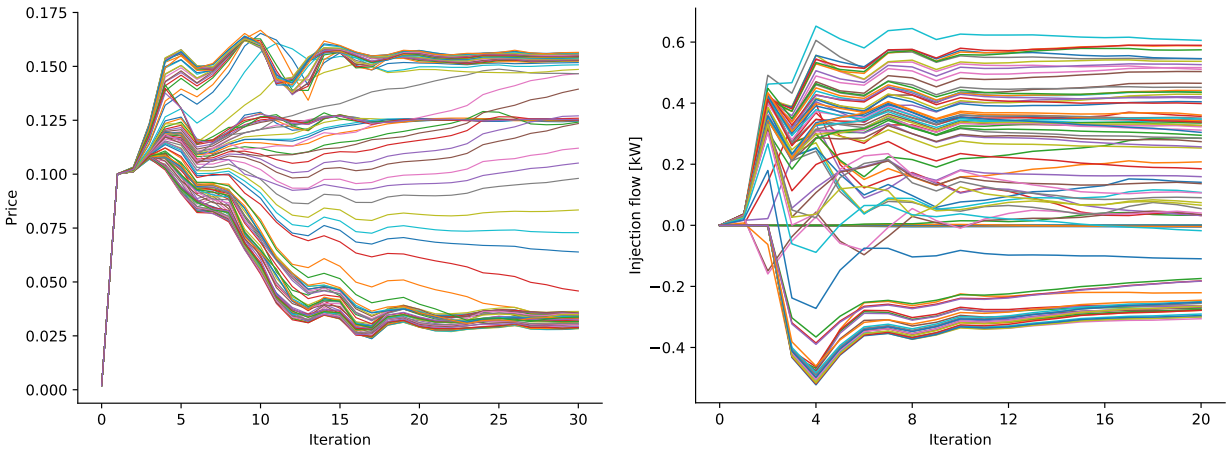


Figure 2: (a) CPU time for the three algorithms as a function of the state dimension

Convergence of the SDDP algorithm on all instances, the approximate upper bound is estimated every 10 iterations, with 1,000 scenarios. On the 12-Nodes problem, we observe that the gap between the upper and lower bounds is below 1% after 180 iterations and that the lower bound remains stable after 250 iterations.

Convergence of the DADP and PADP algorithms Figure 3 shows the evolution of DADP's price process and PADP's resource process over the iterations for the 12-Nodes problem. We depict the convergence only for the first node, the evolution of price process and resource process in other nodes being similar. On the left side of the figure, we plot the evolution of the 96 different values of the price process $p^1 = (p_0^1, \cdots, p_{T-1}^1)$ during the iterations of DADP. We observe that most of the prices start to stabilize after 15 iterations, and do not exhibit significant variation after 20 iterations. On the right side of the figure, we plot the evolution of the 96 different values of the resource process $r^1 = (r_0^1, \cdots, r_{T-1}^1)$ during the iterations of PADP. We observe that the convergence of resources is quicker than for prices, as the evolution of most resources starts to stabilize after only 10 iterations.



(a) (b)

Figure 3: Convergence of DADP prices (a) and PADP resources (b) for the 12-Nodes problem

5.3.2 Theoretical bounds for the optimal expected total cost

we then give the lower and upper bounds obtained by SDDP, DADP, PADP in Table 3. The lower bound of the SDDP algorithm is the value $\underline{V}_0^{sddp}(x_0)$ given by the SDDP method. We recall that SDDP returns a lower bound because it uses a suitable resampling of the global uncertainty distribution instead of the original distribution itself (see the discussion in §5.1.1). DADP and PADP lower and upper bounds are given by (15) and (17) respectively. In Table 3, we observe that SDDP and DADP lower bounds are close to each other, and for problems with more than 12 nodes, DADP's lower bound is up to 2.6% better than SDDP's lower bound. However, the gap between the upper bound given by PADP and the two lower bounds is rather large.

# Nodes $\lvert\mathcal{N}\rvert$	3	6	12	24	48
SDDP LB	225.2	455.9	889.7	1752.8	3310.3
DADP LB	213.7	447.3	896.7	1787.0	3396.4
PADP UB	252.1	528.5	1052.3	2100.7	4016.6

Table 3: Upper and lower bounds, given by SDDP, DADP and PADP, for the optimal expected total cost

To sum up, DADP provides slightly better lower bounds than SDDP, while being less computationally demanding (and a parallel version of DADP would yield even better performances).

5.3.3 Online policies simulation results

we now compare the performances of the different algorithms in simulation. As explained in §5.2, we are able to devise online policies induced by SDDP, DADP and PADP for the global problem, and to compute by Monte Carlo an approximation of the expected cost of each of these policies.

The results obtained in simulation are given in Table 4. SDDP, DADP and PADP values are obtained by simulating the corresponding policies on $5,000$ scenarios. The notation $\pm$ corresponds to the 95% confidence interval. We use the value obtained by the SDDP policy as a reference, and compute the relative difference: a positive percentage means that the associated decomposition-based policy is better than the SDDP policy. Note that all these values correspond to admissible policies for the global problem (6), and thus are *statistical* upper bounds of the optimal cost $V_0\sharp(x_0)$ of Problem (6).

We make the following observations. i) For problems with more than 6 nodes, both the DADP policy and the PADP policy beat the SDDP policy. ii) The DADP policy gives better results than the PADP policy. iii) Comparing with the last line of Table 3, the statistical upper bounds obtained by the three simulation policies are much closer to SDDP's and DADP's lower bounds than PADP's exact upper bound. By assuming that the resource coordination process is deterministic in PADP, we impose constant importation flows for every possible realization of the uncertainties, thus penalizing heavily the PADP algorithm (see also the interpretation of PADP in the case of a decentralized information structure in [5, §3.3]).

6 Conclusion

We have addressed the mathematical problem of optimal management of urban microgrids by using two decomposition algorithms relying on a deterministic price (resp. resource) coordination process: DADP and PADP. Both algorithms work in a distributed manner and are fully parallelizable. We have conducted

# Nodes $\lvert\mathcal{N}\rvert$	3	6	12	24	48
SDDP value	226 ± 0.6	471 ± 0.8	936 ± 1.1	1859 ± 1.6	3550 ± 2.3
DADP value	228 ± 0.6	464 ± 0.8	923 ± 1.2	1839 ± 1.6	3490 ± 2.3
DADP/SDDP	- 0.8 %	+ 1.5 %	+1.4%	+1.1%	+1.7%
PADP value	229 ± 0.6	471 ± 0.8	931 ± 1.1	1856 ± 1.6	3508 ± 2.2
PADP/SDDP	-1.3%	0.0%	+0.5%	+0.2%	+1.2%

Table 4: Simulation results for SDDP, DADP and PADP induced policies

numerical simulations on microgrids of different sizes and topologies, with up to 48 buildings. We have compared the two decomposition algorithms with a state-of-the-art SDDP algorithm. Numerical results have shown the effectiveness of DADP, that gives better results than the reference SDDP algorithm for problems with more than 12 nodes — both in terms of theoretical bounds and of economic performance induced by online policies. On problems with up to 48 nodes (corresponding to 64 state variables), we observed that the performances of DADP and PADP scale well as the number of nodes grew. Numerically, we observe that decomposition-coordination methods are less impacted by the curse of dimensionality than SDDP, as: i) decomposed subproblems have small dimension (1 or 2) and can be solved in parallel; ii) as the size of the problem grows, we empirically observe that the number of iterations of decomposition methods grows slower than with SDDP. Thus, algorithms that mix spatial and temporal decompositions appear to be a promising tool to address large-scale microgrid optimal management problems.

References

[1] Léonard Bacaud, Claude Lemaréchal, Arnaud Renaud, and Claudia Sagastizábal. Bundle methods in stochastic optimal power management: A disaggregated approach using preconditioners. *Computational Optimization and Applications*, 20(3):227–244, 2001.

[2] Ruben Baetens and Dirk Saelens. Modelling uncertainty in district energy simulations by stochastic residential occupant behaviour. *Journal of Building Performance Simulation*, 9(4):431–447, 2016.

[3] Kengy Barty, Pierre Carpentier, and Pierre Girardeau. Decomposition of large-scale stochastic optimal control problems. *RAIRO-Operations Research*, 44(3):167–183, 2010.

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

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

[6] P. Carpentier, J.-Ph. 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.

[7] Pierre Girardeau, Vincent Leclère, and Andrew B. Philpott. On the convergence of decomposition methods for multistage stochastic convex programs. *Mathematics of Operations Research*, 40(1):130–145, 2014.

[8] Vincent Guigues. Dual dynamic programing with cut selection: Convergence proof and numerical experiments. *European Journal of Operational Research*, 258(1):47–57, 2017.

[9] Matt Kraning, Eric Chu, Javad Lavaei, Stephen P Boyd, et al. *Dynamic network energy management via proximal message passing*. Now Publishers, 2014.

[10] Nils Löhndorf and Alexander Shapiro. Modeling time-dependent randomness in stochastic dual dynamic programming. *European Journal of Operational Research*, 273(2):650–671, 2019.

[11] Daniel K Molzahn, Florian Dörfler, Henrik Sandberg, Steven H Low, Sambuddha Chakrabarti, Ross Baldick, and Javad Lavaei. A survey of distributed optimization and control algorithms for electric power systems. *IEEE Transactions on Smart Grid*, 8(6):2941–2962, 2017.

[12] Juan M Morales, Antonio J Conejo, Henrik Madsen, Pierre Pinson, and Marco Zugno. *Integrating renewables in electricity markets: operational problems*, volume 205. Springer Science & Business Media, 2013.

[13] T. Morstyn, B. Hredzak, and V. G. Agelidis. Control strategies for microgrids with distributed energy storage systems: An overview. *IEEE Transactions on Smart Grid*, 9(4):3652–3666, 2018.

[14] François Pacaud. *Decentralized Optimization Methods for Efficient Energy Management under Stochasticity*. Thèse de doctorat, Université Paris-Est, 2018.

[15] Anthony Papavasiliou and Shmuel S Oren. Multiarea stochastic unit commitment for high wind penetration in a transmission constrained network. *Operations Research*, 61(3):578–592, 2013.

[16] Mario VF Pereira and Leontina MVG Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical programming*, 52(1-3):359–375, 1991.

[17] Napat Rujeerapaiboon, Kilian Schindler, Daniel Kuhn, and Wolfram Wiesemann. Scenario reduction revisited: Fundamental limits and guarantees. *Mathematical Programming*, pages 1–36, 2018.

[18] Thomas Schütz, Rita Streblow, and Dirk Müller. A comparison of thermal energy storage models for building energy system optimization. *Energy and Buildings*, 93:23–31, 2015.

[19] A. Shapiro. Analysis of Stochastic Dual Dynamic Programming Method. *European Journal of Operational Research*, 209:63–72, 2011.

[20] Alexander Shapiro, Wajdi Tekaya, Joari P da Costa, and Murilo P Soares. Final report for technical cooperation between Georgia Institute of Technology and ONS – Operador Nacional do Sistema Elétrico. *Georgia Tech ISyE Report*, 2012.

[21] Tiago Sousa, Tiago Soares, Pierre Pinson, Fabio Moret, Thomas Baroche, and Etienne Sorin. Peer-to-peer and community-based markets: A comprehensive review. *Renewable and Sustainable Energy Reviews*, 104:367–378, 2019.

[22] Paul Stadler, Araz Ashouri, and François Maréchal. Distributed model predictive control of energy systems in microgrids. In *2016 Annual IEEE Systems Conference (SysCon)*, pages 1–6. IEEE, 2016.

[23] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.

[24] Qipeng P Zheng, Jianhui Wang, and Andrew L Liu. Stochastic optimization for unit commitment—a review. *IEEE Transactions on Power Systems*, 30(4):1913–1924, 2014.