



**HAL**  
open science

# Controlling Microgrids Without External Data: A Benchmark of Stochastic Programming Methods

Alban Puech, Tristan Rigaut, Adrien Le Franc, William Templier,  
Jean-Christophe Alais, Maud Tournoud, Victor Bossard, Alejandro Yousef,  
Elena Stolyarova

► **To cite this version:**

Alban Puech, Tristan Rigaut, Adrien Le Franc, William Templier, Jean-Christophe Alais, et al..  
Controlling Microgrids Without External Data: A Benchmark of Stochastic Programming Methods.  
2023. hal-04083666v1

**HAL Id: hal-04083666**

**<https://hal.science/hal-04083666v1>**

Preprint submitted on 27 Apr 2023 (v1), last revised 4 May 2023 (v3)

**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.

# Controlling Microgrids Without External Data: A Benchmark of Stochastic Programming Methods

Alban Puech<sup>1</sup>, Tristan Rigaut<sup>1</sup>, Adrien Le Franc<sup>2</sup>, William Templier<sup>1</sup>, Jean-Christophe Alais<sup>1</sup>, Maud Tournoud<sup>1</sup>,  
Victor Bossard<sup>1</sup>, Alejandro Yousef<sup>1</sup>, Elena Stolyarova<sup>1</sup>

<sup>1</sup> *Schneider Digital - AI Hub, Schneider Electric*

<sup>2</sup> *LAAS, CNRS*

France

{alban.puech, tristan.rigaut, maud.tournoud}@se.com

**Abstract**—Microgrids are local energy systems that integrate energy production, demand, and storage units. They are generally connected to the regional grid to import electricity when local production and storage do not meet the demand. In this context, Energy Management Systems (EMS) are used to ensure the balance between supply and demand, while minimizing the electricity bill, or an environmental criterion. The main implementation challenges for an EMS come from the uncertainties in the consumption, the local renewable energy production, and in the price and the carbon intensity of electricity. Model Predictive Control (MPC) is widely used to implement EMS but is particularly sensitive to the forecast quality, and often requires a subscription to expensive third-party forecast services. We introduce four Multistage Stochastic Control Algorithms relying only on historical data obtained from on-site measurements. We formulate them under the shared framework of Multistage Stochastic Programming and benchmark them against two baselines in 61 different microgrid setups using the EMSx dataset. We report yearly average cost savings of 7413 euros using our best-performing algorithm over an MPC algorithm that uses the same forecast model, and 21012 euros over our heuristic.

**Index Terms**—microgrid, model predictive control, stochastic programming, stochastic optimization

## I. INTRODUCTION

The US Department of Energy defines a microgrid as a “group of interconnected loads and distributed energy resources with clearly defined boundaries that act as a single, controllable entity and can connect and disconnect from the grid to operate in both grid-connected and island mode” [1]. Microgrids may play a prominent role in increasing climate resilience, with the opportunity to deploy more renewable energy and help to reduce energy costs.

Energy Management Systems (EMS) automatically control microgrids, optimizing an objective function (usually minimizing the cost), while respecting constraints under uncertainties in the energy production/demand/price. Model Predictive Control (MPC) is the most common method to implement an EMS [2], and simply computes controls (*i.e.* actions to take) for a given time interval, using forecasts over a rolling horizon as proxies of future uncertainties. Since MPC is intuitive and scales well w.r.t the number of state variables, it has a wide history of industrial applications. Moreover, it relies on mature mathematical programming techniques and software. However, it is particularly sensitive to the quality of forecasts

so implementing a robust MPC for an industrial application can require paying third-party services to obtain them. These services are often expensive and increase the complexity of EMS development and maintenance.

Instead of being addressed by forecasts, uncertainties can be modelled as stochastic processes conditioned by past observations. This approach to EMS design has proved successful for various microgrid settings, e.g. in [3], [4], [5]. In all three references, stochastic programming methods [6], [7] outperform MPC controllers in most test scenarios. We find these results interesting from the perspective of removing third-party forecasts from the EMS components.

In this paper, we benchmark several EMS solutions that are only allowed to make decisions based on past microgrid data observations. We use a probabilistic modeling of future uncertainties fitted on these data, which relies on the Darts toolbox [8]. We compare four different algorithms, namely stochastic programming (SP), two-stage stochastic programming (2S-SP), two-stage stochastic programming with scenario clustering (2S-SP-C) and MPC. We further benchmark them against a heuristic (HEU) algorithm and an MPC algorithm relying on perfect forecasts of the uncertainties (P-MPC). We assess them on a simple grid-connected microgrid consisting of a single energy storage, a local renewable production and a local load. Our main contributions are:

- A comprehensive mathematical formulation of four different control methods that do not use any external data, under the shared framework of Multistage Stochastic Programming.
- A large-scale benchmark of the algorithms for the control of 61 anonymized industrial sites from the EMSx dataset made publicly available by Schneider Electric<sup>1</sup> [4].
- A quantitative analysis and discussion of the performance of the algorithms on the different sites of the dataset.

## II. MICROGRID MODEL AND SIMULATOR

In this paper, we design an EMS to control a simple grid-connected microgrid. The microgrid includes a building consuming an uncertain load; solar panels producing an uncertain amount of energy; and a battery with an effective

<sup>1</sup><https://zenodo.org/record/5510400#.YUizGls69hE>

capacity  $\bar{S} \in \mathbb{R}_+^*$ , a maximum charge power  $\bar{B} \in \mathbb{R}_+^*$  and a minimum discharge power  $\underline{B} \in \mathbb{R}_-^*$ .

### A. Microgrid model

We model the microgrid as a discrete-time dynamical system. Decisions are made at every time-step of the sequence  $\mathcal{T} = \{0, \dots, T\}$  of length  $T \in \mathbb{N}^*$  and time increment  $\Delta t$ . We introduce the system dynamics during the time interval  $[t, t+1)$  of length  $\Delta t$ .

*a) Battery operation:* let  $s_t$  denote the amount of energy stored in the battery at the beginning of the time interval; which is bounded by  $0 \leq s_t \leq \bar{S}$ . The EMS can charge/discharge an amount  $p_t^b$  of energy in the battery, as expressed in the discrete-time dynamic equation:

$$s_{t+1} = s_t + \rho_c \max\{0, p_t^b\} + \rho_d^{-1} \min\{0, p_t^b\}. \quad (1)$$

$p_t^b$  may be zero, positive (charge), or negative (discharge) and is bounded by  $\underline{B}\Delta t \leq p_t^b \leq \bar{B}\Delta t$ . Each microgrid in the benchmark has specific charge/discharge efficiency coefficients  $\rho_c \in (0, 1]$  and  $\rho_d \in (0, 1]$ , and an initial energy stock  $s_0$ .

*b) Energy production and demand:* we denote by  $p_t$  the energy produced by solar panels and by  $d_t$  the energy demand.

*c) Energy exchanges with the grid:* the EMS can import an energy amount

$$e_t = \max\{0, d_t - p_t + p_t^b\} \quad (2)$$

from the power grid. These importations occur when the *net load*  $d_t - p_t$  plus energy in/outflows  $p_t^b$  from the battery are nonnegative.

*d) Grid import fee:* electricity imports induce a cost that equates to the price of the energy volume  $e_t$ , paid at a cost  $c_t$  in €/kWh, plus a fixed penalty of  $\bar{C}$  in € if this volume exceeds the subscribed power  $\bar{E}$  of the installation. It is expressed as:

$$c_t \times e_t + \bar{C} \times \delta_{\geq \bar{E}}(e_t), \quad (3)$$

where  $\delta_{\geq \bar{E}}$  is the indicator function equal to 1 when the import exceeds the maximum power. We assume that when the microgrid produces more energy than it consumes, the exported energy is not remunerated by the grid operator.

### B. Optimal control framework

*a) Standard control notations:* we adopt standard control notations as used in [6]. State, control, and exogenous random variables are respectively noted as:

$$x_t = s_t, \quad u_t = p_t^b, \quad w_t = d_t - p_t. \quad (4)$$

The aforementioned dynamics (1) and the constraints on  $(s_t, p_t^b)$  are summarized as follows:

$$x_{t+1} = f(x_t, u_t, w_t), \quad g_t(x_t, u_t, w_t) \leq 0, \quad (5)$$

where  $f_t : \mathbb{R}^3 \rightarrow \mathbb{R}_+$  and  $g_t : \mathbb{R}^3 \rightarrow \mathbb{R}^{n_c}$  denote respectively the dynamics function and the constraints. The stage cost described in (3) is written as a function  $L_t : \mathbb{R}^3 \rightarrow \mathbb{R}_+$  of  $(x_t, u_t, w_t)$ .

*b) Controller definition:* At any time-step  $t \in \mathcal{T}$ , the decision  $u_t$  is made based on the current state  $x_t$ , and on the *history* of the microgrid up to time  $t$ , defined as:

$$h_t = \{w_s\}_{s \leq t}. \quad (6)$$

Then, a microgrid controller is a collection  $\phi = \{\phi_t\}_{t \in \mathcal{T}}$  of mappings

$$\phi_t : (x_t, h_t) \mapsto u_t, \quad \forall t \in \mathcal{T}. \quad (7)$$

By design, this generic definition of a microgrid controller is agnostic to the method employed to control the system. This opens our benchmark to any other method from all fields of discrete-time control theory. The core distinction with the microgrid controller from the EMSx benchmark [4] is that the definition of history in (6) does not include external forecasting data.

*c) Evaluating the controllers:* We evaluate the performance of a controller by measuring the costs under its policy  $\phi$  along the scenario  $h = \{h_t\}_{t \in \mathcal{T}}$ , starting from an initial given state  $x_0$ :

$$J(\phi, h) = \sum_{t \in \mathcal{T}} L_t(x_t, u_t, w_t), \quad (8a)$$

$$\text{where } \begin{cases} x_{t+1} = f(x_t, u_t, w_t), \\ g_t(x_t, u_t, w_t) \leq 0, \\ u_t = \phi_t(x_t, h_t), \end{cases} \quad \forall t \in \mathcal{T}. \quad (8b)$$

## III. PRESENTATION OF THE CHALLENGERS

In this section, we introduce our challengers: the different control methods that we have implemented to benchmark against a perfect forecast MPC and a heuristic controller. We frame them all as scenario tree-based Multistage Stochastic Programming (MSSP) policies that differ by the scenario tree that they use.

### A. Multistage Stochastic Programming policies

At every time-step  $t_0$  an MSSP policy observes the state  $x_{t_0}$  and its history  $h_{t_0}$  and searches a control  $\phi_{t_0}(x_{t_0}, h_{t_0})$  in the argmin of the following stochastic optimization problem:

$$\arg \min_{u_{t_0}} \min_{(\mathbf{U}_{t_0+i})_{1 \leq i \leq R}} \mathbb{E} \sum_{t=t_0}^{t_0+R} L_t(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_t), \quad (9a)$$

$$\text{s.t. } \mathbf{X}_{t+1} = f_t(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_t), \quad (9b)$$

$$g_t(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_t) \leq 0, \quad (9c)$$

$$\mathbf{X}_{t_0} = x_{t_0}, \mathbf{H}_{t_0} = h_{t_0}, \quad (9d)$$

$$\sigma(\mathbf{U}_t) \subset \sigma(\mathbf{H}_{t_0}, \mathbf{W}_{t_0+1}, \dots, \mathbf{W}_t). \quad (9e)$$

where  $R$  is a rolling horizon and the expectation integrates over the conditioned distribution  $\mathbb{P}(\mathbf{W}_{t_0+1}, \dots, \mathbf{W}_{t_0+R} | h_{t_0})$ . The policy only returns a control  $u_{t_0}$  for time-step  $t_0$ , that is an element of the argmin in (9a). The min over the subsequent decisions in (9a) measures the expected impact of our current control on the future costs incurred by these decisions. The measurability constraint (9e) states these future decisions are functions of future uncertainties. If the distribution

$\mathbb{P}(\mathbf{W}_{t_0+1}, \dots, \mathbf{W}_{t_0+R} | h_{t_0})$  has infinite support, the problem is hard to solve because the solution space is infinite-dimensional.

We introduce the concept of scenario tree and how it is used to approximate (9) by a tractable optimization problem.

1) *Scenario tree*: A scenario tree  $G = (\mathcal{N}, \mathcal{P})$  is a tree graph representing the distribution of a discrete stochastic process  $(\Xi_0, \dots, \Xi_R)$  with finite support.  $\mathcal{N}$  is a finite set of nodes, a node  $n$  at a depth  $t$  is a realization  $(\xi_1, \dots, \xi_t)$  of the process up to time  $t$ .  $\Xi_0$  has only 1 realization  $\xi_0$  with probability 1 so there is a single node, the root  $n_0$ , at depth 0. Elements  $(m, n) \in \mathcal{P} \subset \mathcal{N} \times \mathcal{N}$  are edges of the tree. We call  $d_n$  the depth of a given node  $n$ . We define the probability  $\pi(n)$  of a node  $n$  as  $\mathbb{P}(\Xi_0 = \xi_0, \dots, \Xi_{d_n} = \xi_{d_n})$ , or equivalently, the product of probability transitions from the root  $n_0$  to  $n$ .

2) *Construction of a scenario tree*: Many methods exist to generate a scenario tree modeling the distribution of a given stochastic process  $(\mathbf{W}_0, \dots, \mathbf{W}_R)$  from a collection of samples. We refer to [9] and [10] for recent reviews of the methods.

In this paper, we generate scenario trees to approximate the distribution of the netload stochastic process  $(\mathbf{W}_{t+1}, \dots, \mathbf{W}_{t+R})$  conditioned by the netload history  $h_t = (w_0, \dots, w_t)$ . To do so, we train a machine learning model to sample from  $\mathbb{P}(\mathbf{W}_{t+1}, \dots, \mathbf{W}_{t+R} | h_t)$ . Our model uses exclusively past realizations to model future uncertainties: this is an *autoregressive* model.

3) *Scenario tree approximation*: Approximating the stochastic process  $\mathbb{P}(\mathbf{W}_{t_0+1}, \dots, \mathbf{W}_{t_0+R} | h_{t_0})$  by a scenario tree  $G$  in (9) turns the solution space into a finite dimensional one. Indeed constraint (9e) implies that, for each node  $n$  of  $G$ , there is one decision  $u_n$ , and due to constraint (9b) one state  $x_n$ . We call  $w_n = w_{d_n}$  the last netload realization of the node  $n$ . We rewrite (9) into its extensive formulation (10).

$$\arg \min_{u_{n_0}} \min_{u_{n \in \mathcal{N} \setminus \{n_0\}}} \sum_{n \in \mathcal{N}} \pi(n) \times L_{d_n+t_0}(x_n, u_n, w_n), \quad (10a)$$

$$\text{s.t. } x_n = f_{d_n+t_0}(x_m, u_m, w_m), \quad \forall (m, n) \in \mathcal{P}, \quad (10b)$$

$$g_{d_n+t_0}(x_n, u_n, w_n) \leq 0, \quad \forall n \in \mathcal{N}, \quad (10c)$$

$$x_{n_0} = x_{t_0}. \quad (10d)$$

In our case, problem (10) is a so-called Mixed-Integer Linear Program (MILP) that we model with Pyomo [11] and solve using HiGHS [12]. The number of variables (resp. constraints) in (10) grows linearly with the number of nodes (resp. edges) in the tree, while the latter grows exponentially with the number of time steps in the rolling horizon. This significantly deteriorates the performance of a MILP solver.

We present hereunder four methods to compute controls using scenario trees that are small enough to solve problem (10) in a reasonable amount of time.

### B. Challengers

Without loss of generality, we fix  $R = 23$  (control horizon of 24 hours). We formulate each of our challengers as an MSSP method and use a LightGBM model implemented using Darts [8] to draw  $K$  samples  $(\tilde{w}_0, \tilde{w}_1^i, \tilde{w}_2^i, \dots, \tilde{w}_{23}^i), 1 \leq i \leq K$  from  $\mathbb{P}(\mathbf{W}_{t+0}, \dots, \mathbf{W}_{t+R} | h_t)$ , that are later used to generate a

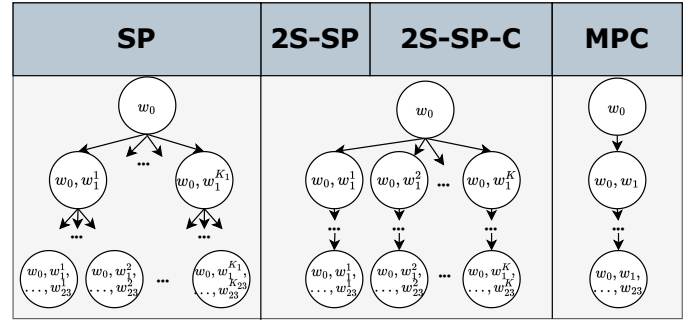


Fig. 1. Diagram of the construction of the trees for each algorithm. We use  $R = 23$  (control horizon of 24h)

scenario tree. A challenger is identified by the kind of scenario tree it generates, as shown in Fig. 1:

1) *SP*: SP (short for Stochastic Programming) takes as input  $(\tilde{w}_0, \tilde{w}_1^i, \tilde{w}_2^i, \dots, \tilde{w}_{23}^i), 1 \leq i \leq K$  and builds a scenario tree with  $K_{23}$  leaves  $(w_0, w_1^{1 \leq i_1 \leq K_1}, w_1^{1 \leq i_2 < K_2}, \dots, w_{23}^{1 \leq i_{23} < K_{23}})$  using scenedr [13]. The reduction parameter defines the number of nodes  $K_1, K_2, \dots, K_{23}$  at each depth of the tree.

2) *2S-SP and 2S-SP-C*: Two-stage stochastic programming, denoted by 2S-SP directly turns each sample into a branch of the tree connected to the root so that  $\forall 1 \leq i \leq K : (w_0^i, w_1^i, \dots, w_{23}^i) = (\tilde{w}_0^i, \tilde{w}_1^i, \dots, \tilde{w}_{23}^i)$ . This type of tree is commonly referred to as a scenario fan. 2S-SP-C is a slightly different version of 2S-SP that uses K-means to separate the  $K$  samples into  $K'$  clusters, whose centres are used instead of the original samples to generate the tree.

3) *MPC*: Finally, Model predictive Control MPC is framed as an MSSP method, where the unique scenario (unique branch of the tree) is obtained by averaging all the samples:  $(w_t)_{t \leq 23} = ((1/K) \sum_{1 \leq i \leq K} \tilde{w}_t^i)_{t \leq 23}$

### C. Two baselines

The challengers are compared with two baseline controllers.

1) *P-MPC*: An ideal MPC with perfect forecasts that permits us to assess the value of perfect forecasts compared to our autoregressive models of uncertainties.

2) *HEU*: A myopic heuristic that stores electricity when the netload is negative and discharges the battery when the netload is positive. This baseline is called HEU and represents the traditional optimization-free industrial method.

## IV. RESULTS AND DISCUSSIONS

### A. Simulation Setup

Our challengers (MPC, SP, 2S-SP and 2S-SP-C) are tested against the two baselines (HEU and P-MPC) through 61 simulations. Each simulation is associated with a time series of solar production and energy demand from a given site<sup>2</sup>, with a 15-minute resolution, resampled into series with 1-hour-long time-steps. The energy demand time series corresponds

<sup>2</sup>Nine sites were not used for our simulations, as they contain extended periods of missing data

TABLE I  
HYPERPARAMETER VALUES AND IMPLEMENTATION SPECIFICS

All challengers (MPC, SP, 2S, 2S-SP, 2S-SP-C)	
# past netload values used for the forecast (time-steps)	48
Forecasting model	LightGBM
SP	
# samples used for tree construction	50
Relative distance during (construction, reduction)	0.2, 0.0
2S-SP	
# samples used for tree construction	20
2S-SP-C	
# samples before clustering	100
# clusters used for tree construction	$\leq 20$

to our exogenous uncertainty. Sites each have a microgrid setup, and we further set the subscribed power limit as  $\bar{E} = \max_t(d_t - p_t) - \bar{B}$ . The penalty for exceeding the power subscription is fixed at 14.31€/h, as per the "Tarif Jaune" rates from France's main electric utility company EDF [14]. The electricity prices (also displayed in Fig. 3) are set to 0.102€ during off-peak hours (i.e. from 00:00 to 06:00, 09:00 to 11:00, 13:00 to 17:00, and 21:00 to 00:00) and to 0.153€ during peak hours. We use 60 % of the dataset for training, and the rest for testing. Table I shows the hyperparameter values and other implementation specifics used for the different algorithms across the 61 simulations. Our control horizon is 24h ( $R = 23$ ). All experiments were run on a computer with an Intel i7-12800H CPU and 32 GB RAM.

### B. Results on the 61 sites

We introduce the average performances, both in terms of savings and computation times, of the challengers and baselines across the whole dataset of 61 sites.

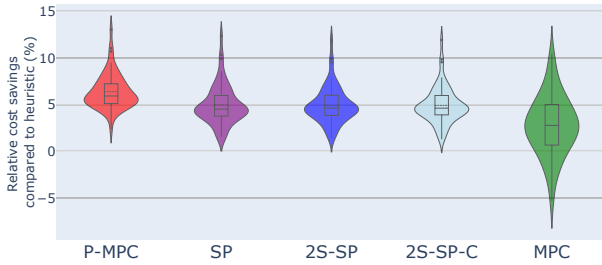


Fig. 2. Relative cost savings from using each of the challenger (SP, 2S-SP, 2S-SP-C) and P-MPC over the heuristic algorithm HEU

1) *Average savings:* We show in Fig. 2 the distribution of the relative cost savings obtained using a given microgrid controller vs HEU. Using SP, 2S-SP, and 2S-SP-C over HEU always yields cost savings, and all three algorithms show similar performances. The savings obtained using MPC are smaller on average, sometimes negative, and show a much larger variance. Note that, in contrast to the other challengers, MPC is performing worse than HEU on almost 25 % of the 61 different simulations. We show in Table II the average results obtained by the different challengers across the 61 simulations. 2S-SP and 2S-SP-C together outperform SP in the majority of the scenarios (71 %), while SP yields higher cost savings on average (4.98 % against 4.94 and 4.93 % for 2S-SP and

2S-SP-C respectively). MPC sometimes performs worse than HEU, and the difference in performance between P-MPC and MPC highlights the importance of having accurate forecasts. Finally, the difference between SP and P-MPC was quite limited (1.51%): using a probabilistic model only with local historical data alleviates the impact of an imperfect forecast.

2) *Computing times:* We additionally report in Table II the mean processing time per control. It corresponds to an average over a 7700-time-step-long test dataset. The training is done on 60 % of the whole dataset and took 2.8s for each challenger. HEU computes a control in the smallest amount of time. P-MPC requires a bit more time as it creates and solves a MILP problem. MPC additionally uses the LGBM model to compute forecasts. 2S-SP also draws samples from the LGBM model but creates a larger MILP problem than MPC, there are more nodes in the scenario tree. 2S-SP-C adds a clustering step to 2S-SP to solve a smaller MILP but still requires more time than 2S-SP. Finally, SP stands between 2S-SP and 2S-SP-C, it adds the scenario tree generation to 2S-SP, which takes less time than the clustering of 2S-SP-C.

### C. Comparison of the control trajectories on a simulation sample

We show in Fig. 3 a sample of the battery state of charge and grid import trajectories under the policies of the different algorithms<sup>3</sup> 2S-SP and 2S-SP-C have the closest trajectories to P-MPC, displayed in light gray. The gains obtained using the challengers over HEU are achieved in two different ways:

a) *Limiting the number of subscribed power limit overruns in high load conditions:* Most of the savings are obtained in the very first (0-25) and last (>200) hours of the simulation slice shown in Fig. 3. The high net load forces all algorithms to exceed  $\bar{E}$  to meet the energy demand, even when not charging the battery. This penalty being fixed, charging the battery on top of importing the energy necessary to satisfy the demand that already exceeds the subscribed power limit does not generate any additional cost (apart from the cost due to a larger volume being bought). The grid import trajectory under the policy of HEU constantly stays over  $\bar{E}$  during those two time-frames, while it is more often below when following the policies of the challengers. On one hand, HEU constantly pays the penalty for exceeding  $\bar{E}$  and imports the electricity necessary to meet the demand without charging the battery. On the other hand, the challengers repeatedly charge and discharge the battery, alternating between time-steps when the battery is charging and when the penalty is paid, and time-steps when discharging the battery allows the grid import to stay below  $\bar{E}$ . In particular, notice how 2S-SP-C succeeds in keeping the grid import below  $\bar{E}$  in the 205-210 time frame.

b) *Arbitrage on the electricity prices and high net load anticipation:* On the rest of the slice, additional savings are obtained thanks to the arbitrage done on the electricity prices. Indeed, they typically charge the battery during off-peak hours, when the price is low, and when the demand is low enough

<sup>3</sup>Note that only a small portion of the simulation on site 19 of the EMSx dataset is displayed and that the full simulation on this site covers 6576 hours.



TABLE II  
SIMULATION RESULTS

Alg.	Mean processing time (ms/it)	Avg savings w.r.t HEU (%)	Extra cost w.r.t P-MPC (%)	Avg yearly savings w.r.t HEU (EUR)	Times ranked 1st challenger algorithm (%)
<b>MPC</b>	36.16	2.77	3.86	13599	0
<b>SP</b>	333.72	<b>4.98</b>	<b>1.51</b>	<b>21012</b>	<b>38</b>
<b>2S-SP</b>	249.35	4.94	1.56	20653	33
<b>2S-SP-C</b>	344.36	4.93	1.57	20584	29
<b>HEU</b>	0.04	.	6.87	.	.
<b>P-MPC</b>	7.01	6.39	.	25945	.

for the grid import (increased by the energy charged into the battery) to stay below  $\bar{E}$ . The battery is then discharged when the electricity prices are higher, or when it allows the grid import that would otherwise exceed  $\bar{E}$  power to stay below, by satisfying a bit of this demand using the energy stored in the battery. In the first case, this saves a cost equal to the amount of energy discharged, multiplied by the price spread. In the second case, this spares the penalty cost associated with the overrun of the subscribed power limit.

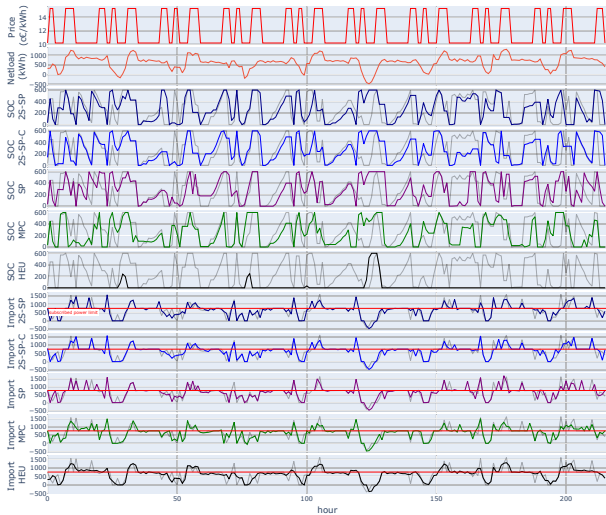


Fig. 3. Sample the control sequences/ trajectories of the challengers and the heuristic algorithm (HEU) running on site 19 of the EMSx dataset. The grey lines correspond to the trajectories under the policy of P-MPC. "Import" refers to the amount of electricity imported from the grid, expressed in KWh.

## V. CONCLUSION

We introduced four different multistage stochastic control algorithms, namely, model predictive control, stochastic programming, two-stage stochastic programming, and two-stage stochastic programming with scenario clustering. We proposed a generic mathematical formulation of those algorithms under the shared framework of Multistage Stochastic Programming. We presented a simple microgrid model and benchmarked our four challengers against two baselines: a simple heuristic, and a model predictive control using perfect forecasts. The benchmark is done through 61 simulations, each having a different microgrid setup and exogenous uncertainty distribution. Our two-stage stochastic programming algorithm shows similar

performances to our stochastic programming one, while at the same time being much easier to implement (as it doesn't require any scenario tree reduction step) and less computationally heavy. Both algorithms outperform the model predictive control approach, reducing the electricity cost by 2.2 % (which amounts to 7413€ of average yearly cost savings) when used over model predictive control, and only generate an extra cost of 1.5 % compared to the perfect baseline. We believe this is an important finding, which questions the financial interest in relying on third-parties services, such as weather/irradiance forecasts, to improve the net load forecast accuracy.

## REFERENCES

- [1] Dan T. Ton and Merrill A. Smith. The u.s. department of energy's microgrid initiative. *The Electricity Journal*, 25(8):84–94, 2012.
- [2] Nor Liza Tumeran, Siti Hajar Yusoff, Teddy Surya Gunawan, Mohd Shahrin Abu Hanifah, Suriza Ahmad Zabidi, Bernardi Pranggono, Muhammad Sharir Fathullah Mohd Yunus, Siti Nadiah Mohd Saphie, and Asmaa Hani Halbouni. Model predictive control based energy management system literature assessment for res integration. *Energies*, 16(8):3362, Apr 2023.
- [3] Tristan Rigaut, Pierre Carpentier, Jean Philippe Chancelier, Michel De Lara, and Julien Waeytens. Stochastic optimization of braking energy storage and ventilation in a subway station. *IEEE Transactions on Power Systems*, 34(2):1256–1263, 2018.
- [4] Adrien Le Franc, Pierre Carpentier, Jean-Philippe Chancelier, and Michel De Lara. EMSx: a numerical benchmark for energy management systems. *Energy Systems*, pages 1–27, 2021.
- [5] François Pacaud, Pierre Carpentier, Jean-Philippe Chancelier, and 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*, pages 1–25, 2022.
- [6] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 1. Athena scientific, 2012.
- [7] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2021.
- [8] Julien Herzen and al. Darts: User-friendly modern machine learning for time series. *CoRR*, abs/2110.03224, 2021.
- [9] Sanjula Kammammettu and Zukui Li. Scenario reduction and scenario tree generation for stochastic programming using sinkhorn distance. *Computers & Chemical Engineering*, 170:108122, 2023.
- [10] Kipgeno Benard Kirui, Georg Ch Pflug, and Alois Pichler. New algorithms and fast implementations to approximate stochastic processes. *arXiv preprint arXiv:2012.01185*, 2020.
- [11] Michael L. Bynum, Gabriel A. Hackebeil, William E. Hart, Carl D. Laird, Bethany L. Nicholson, John D. Sirola, Jean-Paul Watson, and David L. Woodruff. *Pyomo—optimization modeling in python*, volume 67. Springer Science & Business Media, third edition, 2021.
- [12] Qi Huangfu and JA Julian Hall. Parallelizing the dual revised simplex method. *Mathematical Programming Computation*, 10(1):119–142, 2018.
- [13] Holger Heitsch and Werner Römisich. Scenario reduction algorithms in stochastic programming. *Computational Optimization and Applications*, 24:187–206, 2003.
- [14] Arrêté du 30 juillet 2015 relatif aux tarifs réglementés de vente de l'électricité, Jul 2015.