Optimal Management of Storage for Offsetting Solar Power Uncertainty using Multistage Stochastic Programming

Taku Kaneda
Bruno Losseau
Anthony Papavasiliou
Université Catholique de Louvain

Damien Scieur
École Normale Supérieure, Paris

Léopold Cambier
Stanford University

Pierre Henneaux
Niels Leemput
Tractebel Engineering

***Abstract*—Africa has recently engaged in implementing an aggressive renewable energy integration plan. A major challenge in the deployment of renewable power is the management of excess energy. The use of battery storage has been considered as a technically attractive solution. This paper tackles this operational problem using stochastic dual dynamical programming. We present an open-source MATLAB toolbox for multistage stochastic programming which employs stochastic dual dynamic programming. We use the toolbox in order to compare the stochastic solution to a greedy policy which operates batteries without future foresight as a benchmark. We consider a case study of storage management in Burkina Faso. We quantify the benefits of the stochastic solution and test the sensitivity of our results to the optimization horizon of the stochastic program.**

***Index Terms*—Dynamic programming, storage, solar power**

I. Introduction

One of the greatest social challenges presented in Africa is the serious energy access gap. Only 37% of the African population enjoys access to electric power, as compared to a worldwide average value of 85%. African energy demand represents 4% of global energy demand. The continent faces a massive population growth: from 1.2 billion in 2017, African population is expected to grow up to 2.5 billion (25% of worldwide population) by 2050. Poor electricity infrastructure is a key obstacle to economic growth and development in the continent. There is an urgent need for African countries to be electrified, and this need represents an opportunity for Africa to transition towards a renewable energy powered future.

Many initiatives have been launched in recent years in order to tackle the problem of energy access. One of the most significant is the Africa Renewable Energy Initiative (AREI) that was ratified during COP21 on December 7, 2015. The objective of the AREI is to roll out at least 10 GW of new and additional renewable energy generation capacity by 2020, and to mobilize the African potential to install at least 300 GW of renewable capacity by 2030. This would cover the energy access gap of Africa, while ensuring universal access to sufficient amounts of clean, appropriate and affordable energy for all Africans by 2030, and while helping African countries leapfrog to renewable energy systems that support their low-carbon development strategies.

The aggressive renewable energy integration agenda that Africa has embraced is expressed by a strong cooperation between African countries: the West Africa Clean Energy Corridor (WACEC) was launched in 2017, with the objective of accelerating the deployment of utility scale renewable energy into the region. Led by the International Renewable Energy Agency (IRENA) and uniting 14 West African countries, WACEC plans to install 10 GW of renewable capacity into the grid by 2030. Initiatives in this direction are further exemplified by a number of solar generation projects in Senegal, Mali, Burkina Faso and Niger.

In Burkina Faso, numerous projects are underway. The Zagtouli Photovoltaic (PV) power plant announced during COP21 was concretized and started producing 33 MW in September 2017. The construction of a new 20 MW PV facility in Zina was announced on June 2017 and its construction is underway. In 2015, 89% of the electricity that was consumed in Burkina Faso was generated from fossil fuels, whereas 15% was imported from neighboring countries [1]. This configuration of renewable supply is not satisfactory, since it represents a significant cost and is not adequate to fully cover power demand. Renewable energy resources present an interesting alternative to imports and fossil fuel resources for Burkina Faso, as they are clean and perceived as being more affordable [2].

A major challenge in the deployment of solar power, also raised by WACEC, is the management of excess energy during the day, while there is a lack of energy to supply the load during the evening. The large-scale development of battery storage could be a satisfactory complement for the large-scale development of solar PV in countries such as Burkina Faso. Since renewable energy supply and demand cannot be perfectly forecasted, the operational planning of the system at each time period should in principle account for this uncertainty. This presents a problem of decision making under uncertainty, which is treated in this paper as a multi-stage stochastic optimization problem.

In this paper we approach the problem using Stochastic Dual Dynamical Programming (SDDP). The SDDP algorithm has been most successfully applied in the context of medium-term multi-stage hydrothermal scheduling under rainfall uncertainty for handling water levels of hydro reservoirs [3]. We implement the SDDP algorithm on FAST[1], an open-source MATLAB toolbox originally developed at the Université Catholique de Louvain.

Two-stage stochastic unit commitment formulations are commonly used for investigating the operation of storage in short-term operations. The authors in [4] consider a two-stage formulation, with day-ahead unit commitment in the first stage and optimal deployment of storage against realized load forecast errors, wind forecast errors and line and generator outages in the second stage. A similar two-stage stochastic unit commitment model is developed in [5], where commitment and reserve decisions are determined in the day ahead, followed by real-time dispatch against load and wind forecast errors. The stochastic unit commitment model of [6] dispatches conventional units in the first stage, and determines scenario-dependent unit commitment decisions in the second stage. Stochastic programming is used in [7] for obtaining weekly and daily pumped hydro reservoir targets, with reservoir levels being considered as non-anticipative decisions. The hourly dispatch of pumped hydro resources over the duration of a week is solved for by [8], who uses two-stage stochastic unit commitment in order to cope with wind uncertainty and fuel cost uncertainty. Decomposition algorithms for a two-stage stochastic optimal power flow model with an explicit consideration of non-linear power flow constraints have been recently proposed by [9].

Compared to the above literature on two-stage stochastic models, the literature on multi-stage stochastic real-time dispatch is relatively less developed. Two-stage robust optimization is employed in [10], and further extended in [11]. The problem has also recently been investigated by [12], [13] in the context of flexible ramping products, but with emphasis on policy analysis rather than computational challenges. Multi-period stochastic economic dispatch has also been set forth by [14] with an emphasis on efficient sampling methods.

The contributions of this paper can be summarized as follows: (1) From a *policy* standpoint, African power systems present novel technical challenges in terms of managing renewable resources due to their unique capacity mix. In particular, two features of certain African systems are unique, relative to more standard systems that are examined in the literature: (i) the high share of solar power production, and (ii) the lack of sufficient capacity for adequately satisfying demand. These specific features feed into the second contribution of the paper: (2) from a *modeling* standpoint, the paper demonstrates the great potential of multistage stochastic programming in optimizing the management of storage in a system with adequacy challenges, where one faces a significant cost of utilizing domestic thermal resources. Finally, (3) from a *computational* standpoint, the paper presents an open-source software developed by the authors for solving SDDP which can be used for rapid deployment of stochastic programming solutions on MATLAB.

[1]https://web.stanford.edu/∼lcambier/fast/

II. The FAST Toolbox

The optimal operation problem with discrete time steps $t \in T = \{1, 2, \ldots, H\}$, where H is the horizon of the problem, can be expressed as a multi-stage stochastic linear program. We consider a discrete set of realizations of uncertainty following a Markov process. We define Ω_t as the discretized sample space at stage t and $\Omega_{[t]}$ as the set of possible histories up to stage t. Every realization of history $\omega_{[t]} \in \Omega_{[t]}$ has a unique ancestor $A(\omega_{[t]}) \in \Omega_{[t-1]}$. Then the problem can be expressed as follows:

$$\begin{aligned} &\min_x \sum_{t=1}^{H} \sum_{\omega_{[t]} \in \Omega_{[t]}} \pi_{t,\omega_{[t]}} c_{t,\omega_t}^T x_{t,\omega_{[t]}} \\ &W_{t,\omega_t} x_{t,\omega_{[t]}} = h_{t,\omega_t} - T_{t,\omega_t} x_{t-1,A(\omega_{[t]})}, t \in T, \omega_{[t]} \in \Omega_{[t]} \\ &x_{t,\omega_{[t]}} \geq 0, t \in T, \omega_{[t]} \in \Omega_{[t]} \end{aligned} \tag{1}$$

where $\omega_t \in \Omega_t$ is a realization, $\pi_{t,\omega_{[t]}}$ is the probability of history $\omega_{[t]}$ in stage t, c_{t,ω_t} are the cost coefficients, h_{t,ω_t} are the right-hand side parameters, W_{t,ω_t} are the coefficients of the current period decision variables $x_{t,\omega_{[t]}}$, T_{t,ω_t} are the coefficients of the previous period decision variables $x_{t-1,A(\omega_{[t]})}$, and $x_{t,\omega_{[t]}}$ is the set of state and action variables at stage t. The size of this problem grows exponentially in the number of time steps for a fixed number of realizations of uncertainty at each time stage, thus it is impossible to solve since the size of the problem becomes intractable for practical applications.

The SDDP algorithm, which we will use to solve the multi-stage stochastic linear program, decomposes the problem (1) into a collection of subproblems which are called nested L-sharped decomposition subproblem (NLDS)[2]. At each stage $t \in T$ and for each outcome $k \in \Omega_t$:

$$\begin{aligned} NLDS_{t,k} : &\min_x c_{t,k}^T x + V_{t,k}(x) \\ &W_{t,k} x = h_{t,k} - T_{t,k} \hat{x}_{t-1} \\ &x \geq 0 \end{aligned} \tag{2}$$

where $V_{t,k}(x)$ is a piecewise affine convex value function which represents the expected future cost of the remaining stages when the decision is x for stage t and outcome k. Note that $\hat{x}_{t-1}$ is the fixed solution from the previous stage $t-1$, thus it will be treated as a parameter for $NLDS_{t,k}$. Unfortunately, $V_{t,k}(x)$ remains a difficult function, which may involve up to $(H-t)!$ operations to be evaluated. Since $V_{t,k}$ is convex, it can be expressed in the form

$$V_{t,k}(x) \geq V_{t,k}(y) + \delta V_{t,k}(y)^T (x - y), \tag{3}$$

where $\delta V_{t,k}(y)$ is a subgradient of $V_{t,k}$ taken at y. We define $\tilde{V}_{t,k}$ as the lower approximation of $V_{t,k}$ using (3) with a small set of subgradients. Since $\tilde{V}_{t,k}(x)$ is a simple piecewise affine convex function, $NLDS_{t,k}$ can be cast as a tractable linear program.

[2]See http://uclengiechair.be/wp-content/uploads/2017/05/SDDP.pdf

The idea of the SDDP algorithm is to solve the problem by generating promising candidates of decisions, and improving the description of the approximate value function $\tilde{V}_{t,k}(\cdot)$ in the neighborhood of these promising solutions. The idea of the algorithm is to quickly 'zoom in' on the relevant parts of the value functions, and avoid spending unnecessary time on approximating the value function around decisions that are anyways not interesting.

The *forward pass* of the algorithm generates candidate solutions, by drawing Monte Carlo samples of uncertainty over the entire horizon of the problem. For each sample, one steps forward in time by solving problem (2) and generating input for the $NLDS$ of the next time step.

With the candidate decisions of the forward pass at hand, one then generates improved approximations of $\tilde{V}_{t,k}$ in the *backward pass* of the algorithm. For this purpose, one uses the dual multipliers generated from the resolution of $NLDS_{t+1,k}$ for all $k \in \Omega_{t+1}$. The detailed theory supporting the algorithm is not repeated here and can be accessed in e.g. [3], [15], we focus instead here on presenting the aspects of the algorithm that are necessary for using the FAST toolbox.

The fact that the underlying uncertainty is assumed to be Markov implies that the value function for stage $t+1$ can be expressed as $\tilde{V}_{t,k}(x)$. The underlying discrete Markov process can be described by a lattice, which is characterized by the number of time stages, the outcomes at each time stage, and the transition probabilities from a node of a given stage to another node of the following time stage.

More precisely, each stage t consists of a set of nodes $\{1, \dots, |\Omega_t|\}$, and for each of the outcomes $k \in \{1, \dots, |\Omega_t|\}$, $c_{t,k}$ and $h_{t,k}$ are the corresponding realizations of the random vectors c_{t,ω_t} and h_{t,ω_t}, respectively. Similarly, $T_{t,k}$ and $W_{t,k}$ correspond to the realizations of the random matrix T_{t,ω_t} and W_{t,ω_t}, respectively. In addition to describing the realization of these random parameters, it is necessary to describe the transition probability from a node k to $k_{t+1} \in \{1, \dots, |\Omega_{t+1}|\}$ in order to complete the description of the lattice.

The FAST Toolbox provide an implementation of the SDDP algorithm, but also an easy way to model all NLDS (like CVX [16] or YALMIP [17]). Moreover, the toolbox is designed in a way that all NLDS are compiled at the beginning (i.e., the problem is transformed into the format of problem (1)), so that forward and backward passes are performed quickly. The problem can thus be efficiently solved by providing only the description of all NLDS_{t,ω_t} and the transition probabilities $\mathbb{P}[\omega_{t+1}|\omega_t]$. Internally, this data is stored in an object called *Lattice* (denoted by $\mathcal{L}$), which is the core element of the toolbox.

A. Creating the lattice

The lattice is composed by H stages, where each stage contains $|\Omega_t|$ *nodes*, one per realization ω_t. Its main purpose is to summarize the probability of transition from one node to another. The toolbox presents several tools to form the lattice, for example this line forms a lattice modeling the result of H rounds of "head or tails",

```
L = Lattice.latticeEasy(H, N); % N = 2.
```

and each node is stored in `L{t}{j}`, with $j \in \{1, ..., N\}$. The parameter N is equal to the number of scenarios in a stage, in our case $N = 2$ because the possible outcomes are heads or tails. In the node `L{t}{j}` are stored t, j, the NLDS model (which is not yet created) and the transition probabilities for the nodes in the next stage. This basic function set these probabilities to $1/N$, but more complex constructors exist. The most flexible is

```
L = Lattice.latticeEasyMarkovNonConst(H,P)
```

which creates a Lattice with H stages with a variable number of nodes. The argument `P` is a cell array of size $H-1$, where $P\{t\}(i,j)$ represents the probability of transition from realization i at time t to realization j at time $t+1$.

B. Modeling the NLDS

The toolbox provide a way to easily model the NLDS (2) with standard Matlab syntax. We show here the modelization of a simple problem. At each time step t, we receive a quantity of goods Q, and we have to decide to sell a quantity x_t at a random revenue R_{ω_t} or to stock with a cost of C, where the total amount of storage is s_t.

$$\begin{aligned} NLDS_{\omega_t,s_{t-1}} : \min\ & -R_{\omega_t} \cdot x_t + C \cdot s_t \\ & x_t + s_t = Q + s_{t-1} \\ & x_t, s_t \geq 0 \end{aligned} \tag{4}$$

The constraint means that the total amount of goods at time t, $s_{t-1} + Q$, is divided into x_t (the amount we sell) and s_t (the amount of stock at time t). The modelization part is divided into two steps. First, we have to declare the variables x and s in the main script, using the function `sddpVar`,

```
var.x = sddpVar(1,H); var.s = sddpVar(1,H)
```

For simplicity, we put all the variable inside one structure called `var`. Now, we have to write a function which builds the NLDS for a given time t and scenario ω_t. In our example, we assume that $C = 1/2$, and at each time step the price can be *cheap* ($R_{\omega_t} = 1$) or *expensive* ($R_{\omega_t} = 2$).

```
function [constr,obj] = nlds(sce,var)
Q = 10; C=1/2; t = sce.getTime();
R = 1*(sce.index==1)+2*(sce.index==2);
positivity = [var.x(t)>=0, var.s(t)>=0];
if(t == 1)
stock=var.x(t)+var.s(t)==Q;
else
stock=var.x(t)+var.s(t)==Q+var.s(t-1);
end
obj = -R*var.x(t) + C*var.s(t);
constr = [positivity, stock];
```

C. Running the algorithm

It remains to compile the lattice, i.e., the software will explore all nodes of the lattice L and transform the NLDS into a standard form for optimization solver. The main script is written below, which runs the SDDP algorithm on our problem with basic settings.

```
L = Lattice.latticeEasy(H, N); % N = 2.
var.x = sddpVar(H); var.s = sddpVar(H);
L = compileLattice(L,@(sce)nlds(sce,var));
output = sddp(lattice);
```

It is possible to tune many parameters of the algorithm, such as the number of forward passes or the termination criterion.

D. Modeling Alternative Policies

In addition to solving for the stochastic solution, the FAST toolbox allows the user to define alternative policies by specifying linear programs that should be solved at each stage t and for each outcome k, in the same way that the user defines an $NLDS$ for the stochastic program.

For example, we can introduce a greedy policy which optimizes at each stage without foresight. This is done by introducing at each stage t, and for each outcome $k \in \Omega_t$, the following linear program in the toolbox:

$$\begin{aligned} GM_{t,k} : \min_{x} \ & c_{t,k}^T x \\ & W_{t,k} x = h_{t,k} - T_{t,k} \hat{x}_{t-1} \\ & x \geq 0 \end{aligned} \quad (5)$$

Note that, as in (2), the previous stage decision $\hat{x}_{t-1}$ is treated as a parameter, while x, the solution of (5), will be the current decision. What differentiates this policy from the stochastic policy is the fact that the value function approximation $\tilde{V}_{t,k}(x)$ is no longer present. Such policies can be introduced by using the `forwardPass` function of the toolbox, and avoiding using `sddp` itself (to not build the cuts). A perfect foresight policy can also be simulated by using the `waitAndSee` function.

III. Model Description

We now describe the multi-stage stochastic optimal storage operation problem by describing the NLDS (2) for a given stage $t \in T$ and outcome ω_t. We drop the indices t and ω_t in order to lighten notation. In what follows, parameters and functions are denoted by upper case letters and decision variables are denoted by lower case letters.

The objective is to minimize the expected cost caused by power production, imports, and load shedding:

$$\min \left[\sum_{g \in G} C_g(p_g) + CI \cdot pi + VOLL \cdot ls \right] \quad (6)$$

where G is the set of generators, $C_g(\cdot)$ is a piecewise affine convex cost function corresponding to generator $g \in G$ which is producing p_g, CI is the the cost of imports (assumed constant), pi is the amount of imported power, $VOLL$ is the value of lost load, and ls is the amount of load shedding.

The net load, NL, is defined as the difference between the load and the PV power. This is the stochastic input to the system, which appears in the right-hand side $h_{t,k}$ of the constraints in (2). Although we model the load and PV power processes separately (see section IV), we represent net load as a single process in the lattice.

We ignore transmission constraints in this study, although these can be tackled by SDDP [18]. The power balance constraint can then be written as follows:

$$NL + \sum_{j \in J} pd_j + ps = \sum_{j \in J} pb_j + \sum_{g \in G} p_g + pi + ls \quad (7)$$

where J is the set of batteries, pb_j if the power supply when discharging batteries, pd_j is the power demand when charging batteries, and ps corresponds to shedding excess power production.

The dynamics of the batteries is expressed as:

$$s_j = s_{j,t-1} + \left(\eta_j \cdot pd_j - \frac{pb_j}{\mu_j} \right), \ j \in J \quad (8)$$

where s_j represents the storage of battery j at the current stage t, while $s_{j,t-1}$ is a parameter which refers to the storage of battery j at the previous stage $t-1$. η_j (< 1) is the efficiency of charging and μ_j (< 1) is the efficiency of discharging battery j.

The following operating constraints are additionally introduced in the $NLDS$:

$$\begin{aligned} & s_j \leq ST_j, pd_j \leq PD_j, pb_j \leq PB_j, \ j \in J \\ & pi \leq PI \\ & PMin_g \leq p_g \leq PMax_g, \ g \in G \end{aligned} \quad (9)$$

where ST_j is the maximum storage capacity of battery j, PD_j is the charging capacity of the battery, PB_j is the discharge capacity of the battery, PI is the import capacity limit, $PMax_g$ is the production capacity of conventional generator g, and $PMin_g$ is the technical minimum of conventional generator g.

Finally, we impose non-negativity constraints[3]:

$$ls, ps, pi, s_j, pb_j, pd_j, p_g, c_g \geq 0, \ g \in G, j \in J \quad (10)$$

IV. Lattice Generation

We solve the SDDP problem on a lattice in which nodes are associated with net loads and edges are associated with transition probabilities. To build a lattice of net load, we first generate a stochastic model of PV power production and power demand, and we then simulate them (assuming that they are independent) in order to estimate transition probabilities for net load.

[3]There is currently no power market in the West African Power Pool (WAPP) countries, and exchanges are ruled by bilateral agreements. Burkina Faso has a contract with Ivory Coast to import from Ivory Coast at a specific rate, but they do not have any contract to export to Ivory Coast (during the period 2012-2014, exports from Burkina to the Ivory Coast represented a total of 10 MWh while exchanges in the other direction represented 1368 GWh). Because Ivory Coast plans to invest massively in power generation, there is no willingness on their side to conclude import contracts with Burkina Faso. It is thus very difficult to value exports in our model, because there is no guarantee that the energy could effectively be sold to the Ivory Coast.

A. PV Power Production Stochastic Model

We follow the method proposed by [19], which is based on nonparametric kernel density estimation of the conditional probability distribution of solar power production. The implemented model is able to account for the correlation of PV power between adjacent time steps, and model the uncertainty of sunrise and sunset related to random factors such as shading of the PV panels.

We divide the model into two parts, (i) generating joint probability density functions (PDF) and (ii) sampling time series.

(i) Generating the joint PDFs

1) Find the start/end moments of PV generation output in data.[4]
2) Estimate the joint PDF f_{sun} : start/end moments of PV output, t_s and t_e.
3) Estimate the joint PDF of PV power $f_{PV,t-1,t}$ at $t-1$ and t using kernel density estimation.

The profiles are generated in a time-sequential manner, thus the computation time will just increase linearly as the number of time steps is increased.

(ii) Sampling time series

1) Define d the number of hours, and n the number of scenarios.
2) Initialize the output vector $PV_i = [PV_{i1}, PV_{i2}, ..., PV_{id}]$ to be zero for $i = 1, 2, \ldots, n$; let $i = 1$.
3) Generate random samples t_{ss} and t_{es} from f_{sun} by applying inverse transform sampling; let $t = t_{ss}$.
   a) Generate a random sample of PV power production from $f_{PV,t-1,t}$ by applying inverse transform sampling.
   b) If $t > t_{es}$ exit; else let $t = t + 1$ and go back to step 3a.
4) If $i > n$ exit; else let $i = i + 1$ and go back to step 3.

B. Power Demand Stochastic Model

The distribution of power demand is based on univariate kernel density estimation for hourly time data, since the distribution depends strongly on the time of the day. Our analysis of the case study data indicates that there is a significant difference between power demand in weekdays and weekends, thus we separate the data and generate one PDF for weekdays and one for weekends. The sampling method is basically the same as above, except that the first part is replaced by separation of data and estimation by univariate kernel density distribution.

C. Lattice of Net Load

We now build the lattice of net load using PV power and load scenarios generated by the above methods. Since net load is defined as the difference between load and PV power, it can be negative.

Recall that H is the horizon of the model. Denote $|\Omega|$ as the number of nodes at each time step except the first stage, and n as the number of samples that we generate in order to populate our lattice. Further denote $PV_i = [PV_{i1}, \ldots, PV_{iH}]$ and $L_i = [L_{i1}, \ldots, L_{iH}]$ as the vectors of generated samples of PV power and demand for day i respectively. Let $NL_i = [NL_{i1}, \ldots, NL_{iH}]$ denote the net load of day i, with $NL_{ij} = L_{ij} - PV_{ij}$,for $i = 1, \ldots, n$ and $j = 1, \ldots, H$. We now possess n daily samples of net load over H time steps. In order to determine the value of net load associated to each node, we proceed as follows:

1) For each time step ($j = 2, \ldots, H$)
   a) Generate n samples of net load and sort the data of net loads in ascending order.
   b) Discretize the process by dividing the net load values into $|\Omega|$ ranges so as to ensure that each range has the same amount of net load data.
   c) Compute the mean of net load at each range.

Then the transition probabilities can be computed by counting the transition ratio of the n samples from one bin to another over consecutive time steps. This yields an $|\Omega| \times |\Omega| \times H - 1$ tensor which is implemented in the FAST toolbox using the lattice generation function of the toolbox.

V. Case Study

We consider a case study of storage management in Burkina Faso. We use one year of data for calibrating our stochastic model. After describing the system settings, we focus our analysis on (i) the comparison of the SDDP solution with the greedy policy, and (ii) the analysis of the impact of the time horizon of the stochastic program on the value of the stochastic solution and on computation time. This problem is made available[5].

A. System Settings

1) Parameters: Table I represents the data that we employ in the model. We follow the notation introduced in Section II. We consider a single generator with a constant marginal cost, $C(p) = MC \cdot p$. We also set the technical minimum of the generator $PMin_g$ to be zero. We consider a system with five batteries, which are empty at the beginning.

We denote $PVcap$ as the installed PV capacity. We use the toolbox in order to model a greedy policy according to which we charge the battery if the system has an excess of energy (negative net load) and discharge the battery otherwise.

2) Lattice: We build a lattice of net load using 1,000 samples of weekday PV power and load. The starting hour of the day is set to 7 a.m., which is the earliest time of sunrise in samples. We use $|\Omega| = 10$ lattice nodes at each stage.

[4]In [19], the authors mention that "the moments of PV generator to start and stop producing power are affected by sunrise/sunset and other reasons, like sensitivity of the power measurement device, or shading on the PV arrays."

[5]https://github.com/bl-uno/Optimal-Management-of-Storage-in-Burkina-Faso-Case-Study

TABLE I
THE PARAMETER VALUES APPLIED TO THE CASE STUDY

Parameter	Notation	Value	Unit
Storage capacity of battery j	ST_j	1,000	MWh
Capacity of battery charging	PD_j	200	MW
Capacity of battery discharging	PB_j	200	MW
Capacity of imports	PI	200	MW
Production capacity of thermal unit	$PMax_g$	300	MW
Technical minimum of thermal unit	$PMin_g$	0	MW
Cost of imports	CI	100	\$/MWh
Marginal cost of thermal unit	MC	200	\$/MWh
Value of load shedding	$VOLL$	1,000	\$/MWh
Battery charging efficiency	η_j	0.95	
Battery discharging efficiency	μ_j	0.97	
Installed PV capacity	$PVcap$	1,872	MWp

3) Termination Criterion: The number of Monte-Carlo simulations at every forward pass is set to $K = 25$. We set the algorithm to terminate at the 10th iteration, which is sufficiently large to satisfy both Pereira and Pinto [3], and a small standard deviation criterion whereby the algorithm is terminated only when the standard deviation of the mean cost estimated in the forward pass is small enough with respect to the lower bounds[6]. All of the problems are solved using Gurobi.

B. Performance Comparison

1) Results on a 4-day Forecast: We test a 4-day horizon for SDDP and the greedy method, and generate 1,000 samples in order to assess the performance of the different policies. We summarize the computational results in table II.

The average total cost of SDDP is 1,279,115$ and its 95% confidence interval is [1,261,040$, 1,297,191$] The average objective value of our benchmark is 1,849,804$ and its 95% confidence interval is [1,819,220$, 1,880,389$] On average the proposed method results in savings of 570,689$, which amounts to 30.9% of the total cost compared to the greedy strategy. Regarding the SDDP solution, we observe that we hardly need to use the high-cost domestic fossil fuel generators, and that the total cost consists almost entirely of the imports which are the most economical option. The stochastic policy does not shed load. In the greedy policy load shedding occurs, even though its effect on the total cost is small on average.

We present the dispatch under a single sample in Fig. 1 for SDDP (left) and the greedy policy (right). Note that the abscissa axes, time stage begins from 7 a.m. in real time. The stochastic policy uses the generator at the beginning, when the batteries are empty and demand cannot be covered by imports alone. In subsequent periods the oversupply of solar power is stored in the batteries up to ST_j, however this is not sufficient for satisfying the load during the night. The stochastic solution anticipates shortage during the night, and therefore imports electricity as soon as the battery storage capacity allows it, in such a way that the expected total demand needed until sunrise can be covered only by batteries and imports. We also observe

[6]See https://web.stanford.edu/∼lcambier/fast/tuto.php#std

TABLE II
AVERAGE PERFORMANCE FOR 4-DAY HORIZON WITH 1,000 SAMPLES

	SDDP		Greedy	
	Cost (\$)	Percentage	Cost (\$)	Percentage
Total	1,279,115		1,849,804	
Generator	146,442	11.4%	1,018,955	55.1%
Import	1,132,673	88.6%	659,943	35.7%
Load shedding	0	0%	170,906	9.2%

that the stochastic solution uses imports during daytime, even when there is a negative net load. The imported energy is charged to the batteries in order to avoid using the high-cost domestic thermal unit during the night, since the maximum amount of imports PI is limited. However, on the first night, since the PV production is not sufficient to cover the high load, it is not possible to satisfy the positive net load by the batteries and imports alone. Hence, the stochastic solution resorts to the thermal unit during hour 24.

In the case of the greedy strategy, following sunset the batteries are used first, and always run out of energy. Imports are then required. As the imports are not sufficient for fully covering the load, the high-cost domestic thermal generator is also used. Moreover, at hour 89 and 91 some load is shed since imports and domestic power capacity are not sufficient for covering the positive net load.

2) One-day Lattice Vs. Four-Day Lattice: Theoretical analysis and empirical observations demonstrate that extending the optimization horizon increases the computing time of SDDP roughly linearly for a fixed number of iterations. We have also observed that we require more iterations for satisfying the termination criterion of section V-A3 as the horizon increases. In table III we demonstrate that, compared to a 4-day horizon, the 1-day horizon results in a reduction of computation time by 76.3% given our chosen termination criterion.

In order to further analyze the trade-off between solving the problem over a short horizon (one day) versus a longer horizon (four days), we compute the value function $\tilde{V}_{24,k}(x)$ (see (2)), at hour 24 on the 4-day lattice using the 24-hour solutions x_{24} of both the 1-day and the 4-day horizon model. Running 1,000 samples, we obtain the results shown in table III. Solving the problem on a 4-day horizon provides a benefit of 3.7%.

TABLE III
COMPARISON BETWEEN 4-DAY AND 1-DAY HORIZONS

	Mean of $\tilde{V}_{24}(x)$ (\$)	Computation time (s)
4-day	870,231	966
1-day	903,647	229

VI. CONCLUSION

This paper proposes an approach for the short-term operation of storage with renewable energy sources using the SDDP algorithm. In the case study of Burkina Faso on a 4-day horizon model, the algorithm results on average cost savings of 30.9% relative to a greedy storage dispatch policy. It is shown that the SDDP solution is effective in dealing

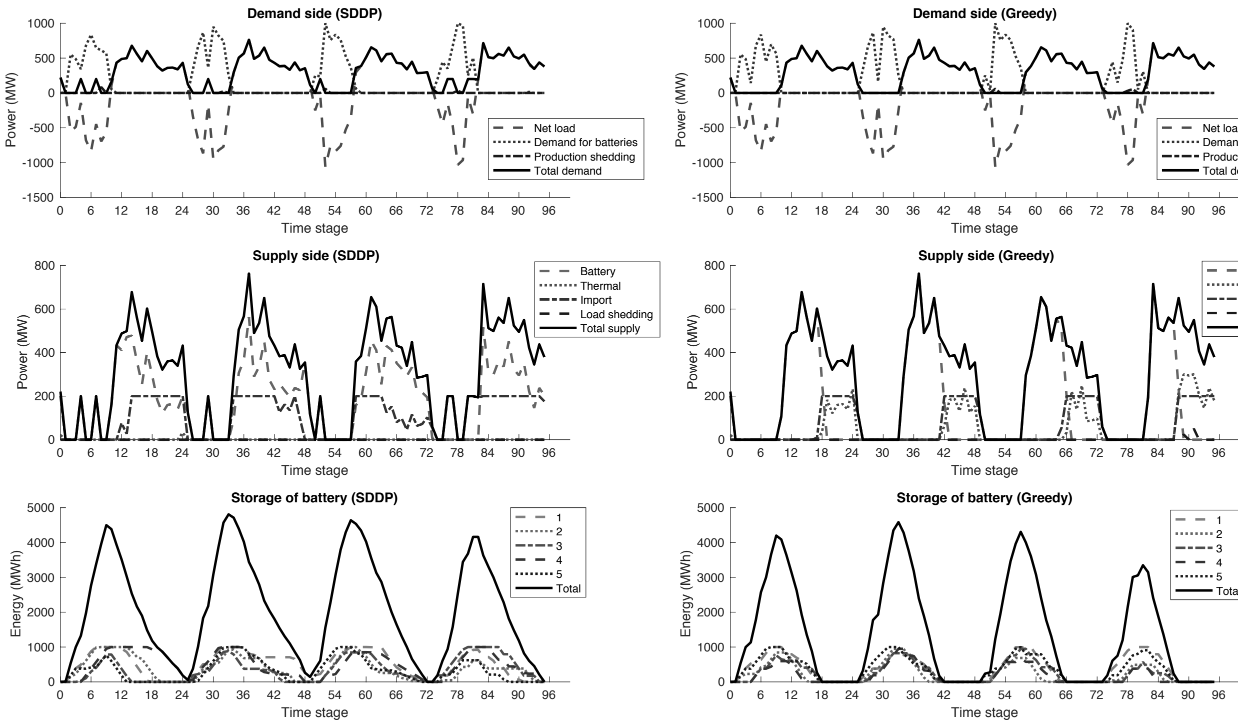

Figure 1. Running a forward pass with a 4 days horizon using SDDP (left) and greedy strategy (right)

with the following trade-off: in the daytime, it is desirable to use the energy generated by PV power, on the other hand it is also important to store power for use at night in order to avoid operating high-cost domestic thermal generators in case the system runs out of import capacity. The proposed problem does not represent some significant properties of the economic dispatch model, such as transmission constraints or ramping constraints, although promising computational results are available in this regard [18].

REFERENCES

[1] "Atlas of Africa energy resources, energy profile: Burkina Faso," United Nations Environment Program, Africa Office, Tech. Rep., 2017. [Online]. Available: https://wedocs.unep.org/bitstream/handle/20.500.11822/20481/Energy_profile_Burkina.pdf?sequence=1&isAllowed=y

[2] M. Taylor and E. Y. So, "Solar PV in Africa: costs and markets," IRENA, Tech. Rep., 2016. [Online]. Available: https://www.irena.org/DocumentDownloads/Publications/IRENA_Solar_PV_Costs_Africa_2016.pdf

[3] M. V. F. Pereira and L. M. V. G. Pinto, "Multi-stage stochastic optimization applied to energy planning," *Mathematical Programming*, vol. 52, pp. 359–375, 1991.

[4] M. Khodayar, M. Shahidehpour, and L. Wu, "Enhancing the dispatchability of variable wind generation by coordination with pumped-storage hydro units in stochastic power systems," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2808–2818, August 2013.

[5] D. Pozo, J. Contreras, and E. Sauma, "Unit commitment with ideal and generic energy storage units," *IEEE Transactions on Power Systems*, vol. 29, no. 6, pp. 2974–2984, November 2014.

[6] N. Li and K. W. Hedman, "Economic assessment of energy storage in systems with high levels of renewable resources," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 3, pp. 1103–1111, July 2015.

[7] J. P. Deane, E. J. McKeogh, and B. P. O. Gallachoir, "Derivation of intertemporal targets for large pumped hydro energy storage with stochastic optimization," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2147–2155, August 2013.

[8] R. Nürnberg and W. Römisch, "A two-stage planning model for power scheduling in a hydro-thermal system under uncertainty," *Optimization and Engineering*, vol. 3, no. 4, pp. 355–378, 2002.

[9] D. Phan and S. Ghosh, "Two-stage stochastic optimization for optimal power flow under renewable generation uncertainty," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 24, no. 1, 2014.

[10] A. Lorca and X. A. Sun, "Adaptive robust optimization with dynamic uncertainty sets for multi-period economic dispatch under significant wind," *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 1702–1713, 2015.

[11] A. Lorca, X. A. Sun, E. Litvinov, and T. Zheng, "Multistage adaptive robust optimization for the unit commitment problem," *Operations Research*, vol. 64, no. 1, pp. 32–51, 2016.

[12] B. Wang and B. F. Hobbs, "A flexible ramping product: Can it help real-time dispatch markets approach the stochastic dispatch ideal?" *Electric Power Systems Research*, vol. 109, pp. 128–140, 2014.

[13] ——, "Real-time markets for flexiramp: A stochastic unit commitment-based analysis," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 846–860, 2016.

[14] C. Safta, R. L.-Y. Chen, H. N. Najm, A. Pinar, and J.-P. Watson, "Efficient uncertainty quantification in stochastic economic dispatch," *IEEE Transactions on Power Systems*, 2016.

[15] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, ser. Springer Series in Operations Research and Financial Engineering. Springer, 2010.

[16] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.0," Stanford University, Tech. Rep., 2013. [Online]. Available: http://cvxr.com/cvx

[17] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *IEEE International Symposium on Computer Aided Control Systems Design*, 2004.

[18] 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*, 2017.

[19] Z. Ren, W. Yan, X. Zhao, W. Li, and J. Yu, "Chronological probability model of photovoltaic generation," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1077–1088, May 2014.