Improving operating policies in stochastic optimization: An application to the medium-term hydrothermal scheduling problem

Jesús D. Gómez-Pérez[a], Jesus M. Latorre-Canteli[b], Andres Ramos[a], Alejandro Perea[b], Pablo Sanz[b] and Francisco Hernández[b]

[a]*Comillas Pontifical University, Calle Alberto Aguilera, 23, Madrid, 28015, Spain*
[b]*Iberdrola Generación, Tomás Redondo, 1, Madrid, 28033, Spain*

ARTICLE INFO

Keywords:
Time series
Fourier Analysis
Optimization methods
Stochastic programming
SDDP
Sampling methods

ABSTRACT

In decision-making under uncertainty, a robust representation of uncertainty is vital for optimal operational and strategic solutions. We extend existing methods by utilizing Fourier decomposition to create multivariate synthetic time series, capturing stochastic seasonal patterns while preserving correlations. These synthetic time series are transformed into a recombining scenario tree via K-means clustering. To enhance the resulting policy in the Stochastic Dual Dynamic Programming (SDDP) framework, we propose an additional sampling within scenario-tree nodes to consider a better representation of the cost-to-go function. A convergence proof for this sampling technique is provided. Moreover, two new stopping criteria are introduced for better solution accuracy and robustness. The first criterion extends traditional stopping rules to all scenario-tree nodes. The second criterion enforces a minimum count of Benders cuts per node, promoting accurate and robust solutions. Our approach is evaluated on the Spanish hydrothermal system, incorporating synthetic time series with seasonal-trend uncertainty in optimization and simulation. Policies from traditional SDDP and our technique were tested over a thousand realizations, demonstrating that our proposals yield reservoir operation policies closer to the thresholds set by the operator compared to traditional SDDP. Computational efficiency is maintained. The proposed sampling mitigates the impact of discretizing stochastic variables into scenario trees by evaluating more scenarios per node. Our framework offers robust policies under uncertainty through stochastic seasonal patterns by Fourier analysis, novel SDDP sampling, and additional stopping criteria.

1. Introduction

Renewable generation plays a crucial role in the energy-transition policies of many countries worldwide Gielen et al. (2019). However, the decision-making process in systems with high shares of renewable sources exhibits significant challenges due to the inherent uncertainty of these energy sources Fodstad et al. (2022). As a result, there has been an increasing emphasis on researching sequential optimization under uncertainty Roald et al. (2023). This approach yields flexible solutions as it considers the potential outcomes of stochastic variables while ensuring economic efficiency. The rolling-horizon technique is utilized to make decisions at other stages, incorporating the most recent outcomes of uncertain parameters Sethi and Sorger (1991).

In many real-world applications, decision-making extends beyond the immediate moment and involves simulating the system's behavior stage-by-stage. It is a key point that the resulting operating policy can effectively handle realizations of stochastic variables that were not considered during the optimization process and predict the behavior of strategic variables in the future Aranha et al. (2022). Achieving this objective requires striking a balance between accurately representing uncertainty and maintaining the performance of the optimization algorithms. This study proposes a method to enhance the operational policy in stochastic sequential problems by improving the representation of uncertainty without compromising the algorithms' performance.

Multistage Stochastic Programming (MSP) is a framework to model sequential decision problems involving uncertainty. In this framework, uncertainty modeling is essential for achieving precise solutions in a reasonable time. Typically, scenario trees are used to depict uncertainty, enabling us to formulate the system as an extended optimization program Dupačová et al. (2000). The construction of a scenario tree generally follows the outlined methodology below:

1. The first step involves mathematically modeling the historical data. This can be accomplished using precise models such as VARIMA models Morales et al. (2010) or machine learning models like neural networks Vagropoulos et al. (2016).
2. The second step entails discretizing the stochastic process by sampling the density distribution function obtained from the fitting process Roald et al. (2023).
3. Finally, the sample set is transformed into a scenario tree using techniques such as clustering, moment-matching approaches, and others Dupačová et al. (2000), Høyland et al. (2003), Latorre et al. (2007).

This work results from the research project funded partially by the AVANHID project

jdgomez@comillas.edu (J.D. Gómez-Pérez); jesuslc@comillas.edu (J.M. Latorre-Canteli); andres.ramos@comillas.edu (A. Ramos); alejandro.perea@iberdrola.es (A. Perea); psanz@iberdrola.es (P. Sanz); fhernandezg@iberdrola.es (F. Hernández)

ORCID(s): 0000-0002-0510-3442 (J.D. Gómez-Pérez); 0000-0002-1227-5453 (J.M. Latorre-Canteli); 0000-0002-8871-1872 (A. Ramos)

Regarding the first step, several papers and technical books expose exact techniques and machine-learning approaches for modeling univariate time series Deb et al. (2017). However, extending these techniques to multivariate time series is usually computationally expensive due to the need to fit models that not only consider the behavior of a single time series but also account for the influence of other time series simultaneously Wei (2018).

Fitting multivariate time series models tends to be more complex and time-consuming when these sequences show seasonal patterns that depend on many previous samples. For instance, this complexity arises when working with time series with annual patterns and sampled at an hourly or daily frequency. One approach to address this issue involves decomposing the time series into a deterministic component to represent the seasonal behavior and a stochastic component to represent the residual part of the times series (Marulanda et al. (2020), Talbot et al. (2020)).

In Marulanda et al. (2020), the Fourier series technique is utilized to model the seasonal pattern of each time series, while the residuals are modeled using the ARIMA model. The errors of the time series are then modeled using a multivariate Gaussian distribution. This work is mainly focused on introducing correlations among the time series into the errors after the detrending and fitting process. However, this study did not consider the correlations between the seasonal patterns and the residuals. In Talbot et al. (2020), a similar approach is employed, but in this case, the correlations between the time series are considered in the residuals and errors. However, this study does not incorporate the uncertainty and correlations in the seasonal component. Both investigations treat the seasonal part as having a deterministic behavior.

In this work, we build upon the ideas presented in these two papers and propose a methodology to generate scenarios where the generated sequences consider the multivariate uncertainty in the seasonal component, the residual part, and the errors while preserving the correlation between the time series.

When uncertainty is discretized into a scenario tree, it becomes possible to represent the system as an extended optimization program. However, as the number of decision stages and tree nodes increases, the computational complexity grows exponentially, creating new challenges when solving large-scale problems (known as the curse of dimensionality). To address this issue, decomposition techniques have been developed. These techniques divide the problem into smaller sub-problems, where each sub-problem considers the exact cost of that particular sub-problem as well as a cost-to-go function that represents future costs Roald et al. (2023).

Benders decomposition (Benders (1962)) is a widely used technique for solving large optimization models iteratively. It decomposes the original formulation into multiple subproblems, making them computationally manageable. However, due to the curse of dimensionality resulting from the number of stages and scenario tree nodes, reaching convergence with Benders decomposition can be time-consuming.

To address this issue, the authors in Pereira and Pinto (1991) proposed a framework called Stochastic Dual Dynamic Programming (SDDP) based on nested Benders decomposition. SDDP replaces the exhaustive search with a Monte Carlo technique. The forward and backward steps are performed in each iteration of the SDDP algorithm. This approach represents the uncertainty as a stochastic process within the optimization problem. In each iteration, this stochastic process is sampled, allowing for the inclusion of continuous uncertainty modeling. However, despite the improvements offered by SDDP, solving large-scale problems can still be time-consuming.

In order to enhance the solution, there are proposals outlined in the literature. For example, the authors incorporated a sampling approach into the optimization process in Penna et al. (2011). This study employed a periodic autoregressive model (PAR) to represent the uncertainty. This implementation resulted in a more stable solution. Another study by the authors in Homem-de Mello et al. (2011) introduced two sampling strategies: Latin hypercube sampling and randomized quasi-Monte Carlo. These techniques perform better than the traditional Monte Carlo technique, enhancing the algorithm's overall performance. Both research papers aim to enhance the algorithm's efficiency and ensure the stability of the initial stage solution. However, it is important to note that these algorithms remain time-intensive, primarily due to the structure and size of the scenario tree.

In Cerisola et al. (2012), the SDDP algorithm was implemented using a recombining scenario tree. A recombining scenario tree is a structure where certain nodes in each stage can be combined or merged considering their subtrees are identical. This merging of nodes helps prevent the exponential growth of the tree's size with an increasing number of time stages Epe et al. (2009). The Benders cuts were formulated for each node at every stage, known as the multi-cut formulation. This approach considers uncertainty in a discrete form and has shown better performance than sampling from continuous probability distributions. The scenario tree structure also allows for easy implementation of the cut-sharing technique. However, while discretizing uncertainty enhances the algorithm's performance, it may introduce some accuracy loss compared to continuous representation.

Other enhancements to the SDDP approach have been proposed in the literature. For instance, in De Matos et al. (2015), the authors introduce three strategies for selecting a subset of Benders cuts in each iteration to speed up the algorithm. Not all generated Benders cuts are utilized in defining the subproblems. Upon completion of the algorithm, the optimal policy is determined by all computed Benders' cuts. The objective of this research was to enhance the algorithm's execution speed. Nevertheless, as concluded by the authors, these strategies do not affect the policy.

To ensure that the algorithm in this study consistently improves the policy at each decision stage, it is essential to review the stopping criteria. The most common approach is to assess the proximity between the lower and upper bounds of the solution. Convergence is achieved when the lower bound falls within the confidence interval of the upper bound of the first stage Shapiro (2011). The authors in Homem-de Mello et al. (2011) examined this criterion and proposed a novel one that involves controlling the type I error (a classical stopping criterion in the SDDP technique) and the type II error in the hypothesis test for convergence. These criteria evaluate the convergence in the first stage of the problem. However, when designing medium and long-term strategies, it becomes essential to have a policy that has converged at every node and each stage of the scenario tree.

In our research, we introduce two additional stopping measures for the algorithm, with a focus on improving the cost-to-go function at each stage. The first criterion involves extending the classical criteria applied to the initial stage to subsequent stages. The second criterion aims to represent the cost-to-go function at each stage using a minimum number of Benders cuts.

Considering the gaps above, our work introduces three strategies to enhance the policy derived from solving a stochastic optimization program. The first strategy focuses on refining the uncertainty modeling in a multivariate space. The second strategy pertains to the sampling approach used in the SDDP algorithm. Lastly, the third strategy addresses the improvement of the stopping criteria.

Given that the objective of this study is to enhance the policy resulting from solving a stochastic program, our contributions can be categorized into two main blocks:

1. Scenario Trees:
    - When utilizing the Fourier series for modeling seasonal patterns in multivariate time series, we propose a novel approach to represent these patterns while incorporating uncertainty. This enables the generation of new seasonal sequences that preserve the key characteristics of historical data.
    - We present a comprehensive methodology for constructing a recombining scenario tree. This process involves modeling historical data using multivariate techniques and employing clustering approaches to establish the structure of the scenario tree.
2. Optimization Algorithm:
    - We introduce a sampling technique considering a recombining scenario tree within the Stochastic Dual Dynamic Programming (SDDP) framework. This algorithm improves the representation of stochastic parameters during optimization, leading to a more accurate approximation of the cost-to-go function at each node in the scenario tree.
    - We analyze the extension of classical stopping criteria beyond the first stage. This extension enables an enhanced approximation of the cost-to-go function in subsequent stages, thereby improving the ability to simulate the system more effectively in the future.

This paper is organized as follows: Section 2 presents the theoretical framework for uncertainty modeling and introduces our approach to constructing recombining scenario trees in multivariate space. Section 3 outlines the theoretical framework of the Stochastic Dual Dynamic Programming (SDDP) algorithm. We discuss our proposals concerning the sampling technique, provide convergence proof for this technique, and explore the extension of classical criteria to stages beyond the first one. Section 4 presents the application of our recommendations to the hydrothermal scheduling problem using the Spanish system. We also include a discussion of the methods employed in this section. Finally, in Section 5, we present the conclusions derived from our work.

2. Uncertainty modeling

The modeling of uncertainty is a fundamental component of the stochastic optimization framework since it directly impacts the ability to make reliable and cost-effective decisions among the diverse range of potential scenarios that may emerge in the future. Uncertain variables often display temporal and spatial correlations, which can add complexity to the problem and require vector models instead of univariate ones. The hydrothermal scheduling problem exemplifies this, where natural water inflows to reservoirs exhibit seasonal patterns and correlated behavior that depends on their geographical location.

The representation of uncertainty using scenario trees is a widely used approach for modeling stochastic variables, applicable in unidimensional and multidimensional contexts. A scenario tree is a mathematical tool that discretizes the underlying random process and provides a framework for representing possible outcomes. It comprises a set of nodes corresponding to different states of the stochastic variables and a set of edges representing the probability of transitioning from one state to another. The overall process for generating a scenario tree typically involves two main steps: first, generating synthetic time series, and second, clustering or grouping the resulting data in a scenario tree form.

For a visual representation of this process, refer to Figure 1, which illustrates the general steps in generating a recombining scenario tree. Detailed explanations of this process will follow in the subsequent sections. In Section 4.1 we present a step-by-step procedure for modeling hydro inflow uncertainty in the Duero, Sil, and Tajo basins within the Spanish system.

2.1. Synthetic time series generation

Synthetic series generation refers to the process of creating artificial time series data that emulate the statistical properties of real-world time series data. This paper employs the approach outlined in Talbot et al. (2020) and proposes improvements to introduce more variability into the synthetic time series. The methodology is presented below (Appendix A provides comprehensive explanations of the equations utilized in this procedure.)

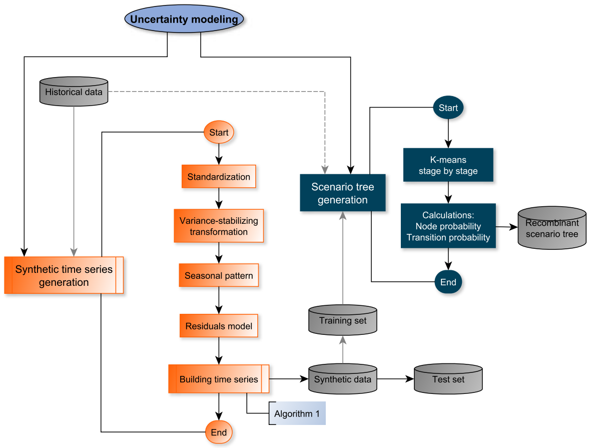

Figure 1: Modeling Uncertainty for Stochastic Optimization

1. Standardization: The initial step in the process is to standardize each time series by subtracting the mean and dividing by the standard deviation. In the hydrothermal scheduling problem, the natural water inflows to each reservoir are influenced by the corresponding feeding river. As a result, data is measured on various scales, highlighting the need for standardization in this specific case.
2. Variance-stabilizing transformation: In contrast to Talbot et al. (2020), this study proposes to transform each non-stationary series into a stationary one as the next step. We employed the probability-distribution transformation to stabilize the time series.
3. Seasonal pattern. Fourier Analysis: Further elaboration on this step can be found in Section 2.1.1, as it constitutes one of the key contributions of this paper.
4. Mathematical model to residuals: The residuals are derived by subtracting the seasonal pattern from the time series. Despite this, the residuals still exhibit autocorrelation and cross-correlations at certain lags. Traditional techniques like VARMA models and neural networks can be employed to capture these correlations. In this investigation, we opted for VAR models for their easy fitting, allowing us to incorporate a broader range of lags.
5. Generation of synthetic series: By considering the mathematical elements arising from the first four steps, we can now proceed with generating synthetic time series, as explained in Section 2.1.2.

2.1.1. Seasonal pattern. Fourier Analysis

In order to create synthetic time series, it is necessary to replicate the seasonal trends found in real-world time series. This can be achieved through deterministic techniques like Fourier decomposition or stochastic methods like SARIMA models. While stochastic methods are commonly used for time series forecasting, their application can become increasingly complex when dealing with multivariate time series with high sampling resolution.

Given that the primary goal of our methodology is scenario generation, we adopted the approach of windowed Fourier decomposition. By utilizing this technique, we are able to model the variability of the most relevant harmonics of the time windows using multivariate probability distributions of their amplitude and phase angle. This model enables us to generate stochastic seasonal trends that accurately preserve the statistical characteristics of the signals. The following is an outline of the steps involved in implementing the proposed analysis:

- **Fourier coefficients computation**: Our first step is to calculate the Fourier coefficients for each selected time window. In this study, we have employed a year as the time series window. This choice aligns with the observation of cyclical patterns, such as monthly or quarterly variations, commonly encountered in hydro-thermal scheduling applications. The formulation for calculating the discrete Fourier coefficients is presented in Equation 1, where $y_i(t)$ represents each dimension i of the multivariate space as a function of time t, k denotes the number of harmonics or Fourier terms, $\omega_0 = \frac{2\pi}{T_0}$ is the fundamental angular frequency, and T_0 is the fundamental period, which coincides with the length of the frame (in our case, corresponding to one year). By decomposing the time series into frames, we can model the amplitude and phase behavior of the common harmonics across these windows.

  $$y_i(t) = \sum_k \left(a_{ik} \cos k\omega_0 t + b_{ik} \sin k\omega_0 t \right) \tag{1a}$$

  $$a_{ik} = \frac{\sum_t y_i(t) \cos k\omega_0 t}{\sum_t \cos^2 k\omega_0 t} \tag{1b}$$

  $$b_{ik} = \frac{\sum_t y_i(t) \sin k\omega_0 t}{\sum_t \sin^2 k\omega_0 t} \tag{1c}$$

  In order to determine which harmonics to use, we calculate the energy of each harmonic throughout one cycle. This measure enables us to rank the harmonics in terms of their impact on the time series, from most significant to the least. The energy of a harmonic over one cycle is calculated using equation 2.

  $$E_{ik} = \frac{\left(a_{ik}^2 + b_{ik}^2\right) T_0}{2k} \tag{2}$$

  The dominant harmonics capture the seasonal trend inherent in each time series. Those harmonics are the ones that exceed a fraction of the most pertinent harmonic. To derive non-seasonal data, we subtract this approximation from each original sequence $y_i(t)$, as depicted in Equation 3, where $\underline{k}$ iterates across the most relevant harmonics. Through this approach, we can eliminate the seasonal patterns within the data, directing our attention to the underlying variations on the residuals $r_i(t)$.

  $$r_i(t) = y_i(t) - \sum_{\underline{k}} \left(a_{i\underline{k}} \cos \underline{k}\omega_0 t + b_{i\underline{k}} \sin \underline{k}\omega_0 t \right) \tag{3}$$

- **Multivariate probability distribution**: We model the amplitude and phase angle of the most significant harmonics obtained in the previous step using a multivariate probability distribution. By employing this model, we can generate stochastic seasonal trends while preserving the spatial correlation of the time series. The spatial correlation is represented in the phase angles. In equation 4, $\mathcal{N}_{\underline{k}}^{A}$ and $\mathcal{N}_{\underline{k}}^{\Phi}$ denote the multivariate Gaussian distribution for the amplitude and phase angle for every representative harmonic.

$$\mathbf{A}_{\underline{k}} = \left[A_{\underline{k}}^{y_1}, \dots A_{\underline{k}}^{y_i}\right] = \mathcal{N}_{\underline{k}}^{A}\left(\mu_{\underline{k}}, \Sigma_{\underline{k}}\right) \tag{4a}$$

$$\Phi_{\underline{k}} = \left[\phi_{\underline{k}}^{y_1}, \dots \phi_{\underline{k}}^{y_i}\right] = \mathcal{N}_{\underline{k}}^{\Phi}\left(\mu_{\underline{k}}, \Sigma_{\underline{k}}\right) \tag{4b}$$

$$A_{\underline{k}}^{y_i} = \left(a_{i\underline{k}}^2 + b_{i\underline{k}}^2\right)^{\frac{1}{2}} \tag{4c}$$

$$\phi_{\underline{k}}^{y^i} = \mathrm{arctan2}\left(\frac{b_{i\underline{k}}}{a_{i\underline{k}}}\right) \tag{4d}$$

2.1.2. *Generation of synthetic series*

By considering the mathematical components arising from the initial four steps (standardization, variance-stabilizing transformation, seasonal pattern, and mathematical modeling of residuals), the process of generating synthetic series involves the following steps:

- Generating a sequence of residuals using the residuals model (in this study, the resulting VAR model). This is done by recursively appending the calculated samples for each time step to the historical series for calculating the next sample.
- Generating a seasonal pattern using equations 1 and 4.
- Aggregating the seasonal and residual parts and applying inverse transformations to recover the original distribution functions, means, and variances.
- Repeating this process for the desired number of synthetic series to be generated.

Algorithm 1 provides an outline of the process for generating synthetic time series by utilizing both the seasonal model and the VAR model for the residuals.

2.2. Generation of scenario tree

The set of synthetic time series is the input for creating a scenario tree. In this study, our focus is on clustering methods to construct the scenario tree. Within this framework, the nodes of the scenario tree are represented by the clusters obtained from the clustering process. The arcs of the scenario tree are represented by the transition probabilities, which are determined based on the number of sequences that simultaneously belong to the initial and final nodes.

Determining the appropriate number of clusters is a common challenge in data clustering, mainly when dealing with high-dimensional data. In real-world applications, it is crucial to consider the trade-off between the level of detail captured by the scenario tree and the effort required to obtain a solution for the stochastic optimization program.

Algorithm 1: Synthetic time series generation

```
// $SS$ is the number of synthetic time series
while $\underline{ss \le SS}$ do
    // Generate residual sequence ($\mathbf{sq}(t)$)
    // $\mathbf{sq}(t) = [sq_1(t), \dots, sq_i(t)]$, where $i$ is every
      dimension
    $\mathbf{sq}(t) \leftarrow []$
    while $\underline{t \le STEPS}$ do
        // Calculate sample of sequence in step $t$
        // VAR model
        // $\mathcal{N}_\epsilon\left(\mu_\epsilon, \Sigma_\epsilon\right)$ is the multivariate Gaussian
          distribution for the errors
        $\hat{\mathbf{r}}(t) = A_1\mathbf{r}(t-1) + \cdots + A_p\mathbf{r}(t-p) + \mathcal{N}_\epsilon\left(\mu_\epsilon, \Sigma_\epsilon\right)$
        // Append sample to history and synthetic
          sequence
        $\mathbf{r}(t) \leftarrow \hat{\mathbf{r}}(t), \mathbf{sq}(t) \leftarrow \hat{\mathbf{r}}(t)$
        // increase step
        $t \leftarrow t + 1$

    // Generate components for each representative
      harmonic
    $\mathbf{A}_{\underline{k}} = \mathcal{N}_A\left(\mu_{\underline{k}}, \Sigma_{\underline{k}}\right)$
    $\Phi_{\underline{k}} = \mathcal{N}_\Phi\left(\mu_{\underline{k}}, \Sigma_{\underline{k}}\right)$
    // Seasonal trend for each time series $i$
    $y_i(t) = \sum_{\underline{k}}\left(a_{\underline{k}} \cos \underline{k}\omega_0 t + b_{\underline{k}} \sin \underline{k}\omega_0 t\right), \forall t$
    // Aggregating residual and seasonal part
    // $\mathbf{y}(t) = [y_1(t), \dots, y_i(t)]$
    $\mathbf{sq}(t) \leftarrow \mathbf{sq}(t) + \mathbf{y}(t), \forall t$
    // Inverse variance-stabilizing transformation
    $T^{-1}\left(\mathbf{sq}(t)\right) = F^{-1}\left(\mathcal{N}\left(\mathbf{sq}(t)\right)\right)$
    // Destandardization
    $\mathbf{sq}(t) = [sq_1(t)\sigma_1 + \mu_1, \dots, sq_i(t)\sigma_i + \mu_i]$
    // increase synthetic series counter
    $ss \leftarrow ss + 1$
```

We employed a recombining scenario tree as a means to model the uncertainty, and in particular, we utilized the k-means algorithm as the clustering technique. In this case, the data is represented in an n-dimensional space, where each time series corresponds to a distinct dimension. At each stage, the number of clusters is determined using the elbow method, although other techniques like silhouette or density methods can also be utilized. The transition probabilities from an initial node in one stage to a final node in the next stage are calculated by dividing the number of time series transitioning from the initial node to the final node by the total number of time series in the dataset.

Each node in the scenario tree is characterized by its centroid, node probability, and the specific series that were grouped. The node probability is calculated as the proportion of the number of time series grouped in that node to the total number of time series.

The series grouped within each node play a crucial role in the optimization process of the SDDP algorithm. By sampling across the scenario tree, which outlines the overall model of the stochastic process, and subsequently sampling within each node to capture more detail of uncertain variables, we can strike a balance between solution accuracy and algorithmic performance. This sampling technique will be analyzed in the next section.

2.3. Highlighting the advantages of our proposal

Regarding the use of windowed Fourier series

Stochastic optimization is an important approach in decision-making under uncertain conditions, where effective uncertainty modeling is crucial. In this context, time series analysis is a valuable tool for such modeling. Traditional methods often employ the Fourier series to uncover underlying temporal patterns. However, these approaches may not consistently address the localized nature of uncertainty in non-stationary time series. Our proposal addresses this limitation by leveraging the benefits of the windowed Fourier series, aiming to enhance models for generating synthetic time series.

By segmenting the time series into localized windows and modeling the dominant harmonics within each segment, we provide an adaptive and dynamic tool for enhanced uncertainty quantification. Below, we outline some benefits of this approach.

- **Improved Interpretability:** Utilizing the windowed Fourier series lets us individually model and interpret important harmonic components.
- **Adaptive Harmonic Modeling:** Our approach adapts to the data by focusing on the most relevant harmonics. This adaptability prevents overfitting, avoiding excessive harmonics and leading to a more efficient representation of the underlying signal structure.
- **Enhanced Probabilistic Modeling:** The ability to probabilistically model the extracted harmonics enhances our capacity to represent uncertainty more effectively. This approach allows for the generation of a broader range of potential scenarios.
- **Scalability:** Our method scales well to long-time series data. Fourier series offer analytical solutions for decomposing complex signals into their constituent harmonic components. These closed-form equations enable efficient and direct calculations regardless of the length of the time series. Using the windowed Fourier series, we can effectively model uncertainty associated with seasonal patterns.
- **Quantification of Temporal Change:** Segmenting the time series allows us to capture temporal changes in the dominant harmonics, facilitating the identification of evolving patterns and trends. This is often hidden when using a static Fourier series approach. This property allows us to generate synthetic series with changes in seasonal patterns.

In Section 4.1, we illustrate the difference between modeling seasonal patterns using static Fourier series and windowed Fourier series approach. Figure 8 shows that static Fourier series may oversimplify seasonal patterns, potentially neglecting the inherent variability and uncertainty within them. Accurate seasonal pattern representation is a key element in scenario generation, which forms the foundation of the stochastic optimization framework.

Regarding the generation of scenario tree

Preserving the series grouped within each cluster after applying the K-means method offers a crucial advantage when constructing a scenario tree for a stochastic optimization model. This approach helps maintain the essential variability and patterns in the original dataset, which can be pivotal for accurately representing uncertain future states in the optimization model. Furthermore, maintaining the integrity of these clusters ensures that the model captures not only the central tendencies but also the diversity of potential outcomes. This, in turn, leads to the development of a more robust and realistic decision-making framework.

This approach not only saves time and computational resources but also ensures that the optimization model comprehensively accounts for the intricacies of uncertainty. The result is a more robust policy generated by the stochastic optimization model.

Section 4.3 further expounds on the advantages of employing this type of scenario tree within the SDDP framework.

3. Stochastic Dual Dynamic Programming (SDDP)

Multistage stochastic programming is a framework for modeling sequential decision-making problems that involve uncertainty. Usually, we model the uncertainty by discretizing the stochastic process in the form of a scenario tree which we will denote as Ω_ω. A general formulation for a multistage stochastic problem is as follows:

$$\begin{aligned} \min_{\mathbf{x}_s^\omega} \quad & \sum_s \sum_{\omega \in \Omega_\omega} p_s^\omega f_s^\omega \left(\mathbf{x}_s^\omega \right) \\ \text{s.t.} \quad & \left(\mathbf{x}_{s-1}^{a(\omega)}, \mathbf{x}_s^\omega \right) \in X_s^\omega \qquad \forall \left(\omega, s \right) \end{aligned} \tag{5}$$

In linear stochastic problems, f_s^ω is a linear function. The constraints X_s^ω can be expressed in the form $A_s^\omega \mathbf{x}_s^{a(\omega)} + B_s^\omega \mathbf{x}_s^\omega = b_s^\omega$, where s denotes the stage, ω is the index for each node on the scenario tree, $a(\omega)$ represents the preceding node to ω, p_s^ω is the probability of each node ω in the stage s, $\mathbf{x}_s^\omega$ stands the decision variables at each node ω in the stage s, and A_s^ω, B_s^ω, and b_s^ω are stochastic parameters. The size of multistage stochastic models depends directly on the scenario-tree size (The more stages and uncertain parameters, the larger the mathematical model).

When adopting a multi-cut approach to Benders cuts within the Benders decomposition framework, Equation 6 characterizes the optimization model at each node of the scenario tree for the stochastic program.

$$
\begin{aligned}
P_s^{\omega} = \min_{\mathbf{x}_s^{\omega},\theta^c} \quad & \mathbf{c}_s^{\omega}\mathbf{x}_s^{\omega} + \sum_{c\in C(\omega)} p^c\theta^c && \text{(6a)}\\
\text{s.t.} \quad & B_s^{\omega}\mathbf{x}_s^{\omega} = \mathbf{b}_s^{\omega} - A_s^{\omega}\mathbf{x}_{s-1}^{a(\omega)} \ : \ \pi_s^{\omega} && \text{(6b)}\\
& \pi_{s+1}^{cl} B_{\omega}\mathbf{x}_s^{\omega} + \theta^c \geq f_{s+1}^{cl} + \pi_{s+1}^{cl} B_{\omega}\mathbf{x}_s^{l} \ : \ \eta_s^{\omega} && \text{(6c)}\\
& \mathbf{x}_s^{\omega} \geq \mathbf{0} && \text{(6d)}
\end{aligned}
$$

In equation 6, c represents the subsequent nodes to ω, l is the number of Benders cuts, θ^c stands the future cost in every subsequent node, f_{s+1}^{cl} represents the future cost for a specific set of decision variable values, and π_{s+1}^{cl} represents the sensitivity of f_{s+1}^{cl} to deviations in $\mathbf{x}_s^{\omega}$ from the specific values $\mathbf{x}_s^{l}$.

The SDDP approach follows Algorithm 2, which involves iterative forward and backward steps. In the forward step, the algorithm updates the state variables in every node of the scenario tree. In the backward step, Benders cuts are generated to improve the approximation of the cost-to-go function used in the forward step.

The algorithm's convergence is evaluated in the first stage. A lower bound (LB) can be obtained by calculating the operational cost in the root node plus the cost-to-go function value. Similarly, a statistical upper bound (UB) can be computed as the average cost across all stages in the forward step. Various stopping criteria have been proposed in the literature to terminate the algorithm. One common criterion is to measure the gap between the upper and lower bounds (equation 7). The algorithm continues iterating until this gap falls below a predefined tolerance value. Alternatively, the algorithm can also stop when the LB falls within the confidence interval of the UB (equation 8) Shapiro (2011).

$$
\left|\frac{UB - LB}{UB}\right| \leq \epsilon \tag{7}
$$

$$
LB \in \left[UB - z_{\frac{\alpha}{2}} \frac{\sigma(UB)}{\sqrt{l}}, UB + z_{\frac{\alpha}{2}} \frac{\sigma(UB)}{\sqrt{l}} \right] \tag{8}
$$

The SDDP framework allows us to make operational decisions considering uncertain parameters. In many real-world applications, it is not only the first-stage solution that is required but also the decisions for subsequent stages. These applications are commonly associated with designing strategies in planning contexts. For example, in the hydrothermal scheduling problem, it is crucial to determine the amount of fuel that thermal agents should purchase for future stages, considering the variability of natural water inflows or gas prices. Therefore, the SDDP framework must incorporate additional considerations to ensure an optimal policy at each problem stage. With this in mind, we propose two improvements to the SDDP algorithm:

Algorithm 2: SDDP algorithm

```
Data: Uncertainty in the form of scenario tree $(\Omega_\omega)$
// INITIAL VALUES
// $l$ is the iteration count (number of Benders cuts to be generated)
$l \leftarrow 1$
$CF_0^l \leftarrow 0$
while (stop == False) and ($l \in \mathcal{L}$) do
    // FORWARD STEP
    for $s \in \{1, \dots, S\}$ do
        Solve problem $P_s^\omega$ (equation 6)
        // Calculate actual cost function in stage $s$
        $CF_s^l \leftarrow CF_{s-1}^l + \mathbf{c}_s^\omega \mathbf{x}_s^\omega$
        // Save $\mathbf{x}_s^\omega$ in every iteration
        $\mathbf{x}_s^l \leftarrow \mathbf{x}_s^\omega$
        if $s == 1$ then
            // Calculate the lower ($LB$) bound as the optimal value of the objective function in the root node ($P_1$)
            $LB \leftarrow CF_1^l + \sum_{c\in C(1)} p^c\theta^c$;
        else
            if $s == S$ then
                // Calculate the upper bound ($UB$) as the average of the total cost of all iterations
                $UB \leftarrow \frac{1}{l}\sum_{l\in\mathcal{L}} CF_s^l$;
        // sample a scenario for the next stage
        $\omega \leftarrow \mathrm{rand}(\Omega_\omega(s+1))$
    // BACKWARD STEP
    for $s \in \{S, \dots, 1\}$ do
        for $\forall \omega \in \Omega_\omega(s)$ do
            solve problem $P_s^\omega$ (equation 6)
            // value of equation 6a
            $f_{s+1}^{cl} \leftarrow \mathbf{c}_s^\omega \mathbf{x}_s^\omega + \sum_{c\in C(n)} p^c\theta^c$
            // dual of equation 6b
            $\pi_{s+1}^{cl} \leftarrow \mathrm{dual}\left(B_s^\omega \mathbf{x}_s^\omega = \mathbf{b}_s^\omega - A_s^\omega \mathbf{x}_{s-1}^{a(\omega)}\right)$
    // STOPPING CRITERION VERIFICATION
    // compute stop criterion. Equation 7 or 8
    stop $\leftarrow LB \in [UB - k, UB + k]$
    // increase iteration
    $l \leftarrow l + 1$
```

- We propose to sample the elements within each node of the scenario tree resulting from the clustering process. This sampling occurs after sampling the scenario tree in the forward step. In the backward step, we also sample the elements within each node to generate the Benders' cuts. This approach enables us to consider a broader range of values for the uncertain variables, going beyond just the centroids resulting from the reduction process during the scenario-tree construction, in fact this additional sampling process improves the algorithm convergence.
- We propose new stopping criteria for stages other than the first one. First, the algorithm should verify the gap between the lower and upper bounds in the non-first-stage nodes of the scenario tree. Second, it should ensure a

minimum number of Benders cuts for the cost-to-go function in each scenario tree node, based on its probability. These additional criteria are introduced to guarantee a minimum level of quality in approximating the cost-to-go function in other nodes of the scenario tree, which cannot be achieved solely with classical stopping criteria. The objective is to prevent premature convergence of the algorithm, which could result in generating a suboptimal operating policy. By incorporating these criteria, the algorithm can extensively explore the solution space, leading to a more accurate and reliable operating policy.

3.1. Proposed nodal sampling technique in SDDP framework

As depicted in Algorithm 2, the uncertainty is represented through a scenario tree. Section II outlines the process of constructing a recombining scenario tree using synthetic data.

Clustering processes used to construct the scenario tree can result in a loss of representativeness in stochastic variables. To mitigate this issue, we propose sampling the elements within each node of the scenario tree during both the forward and backward steps in the SDDP algorithm. This additional sampling process is conducted after the conventional sampling across the scenario tree, where a specific node of the scenario tree is selected. Figure 2 visually illustrates these concepts. In this figure, the blue and orange curves represent the path randomly selected in the forward step. In the classic SDDP algorithm, the system is evaluated at the centroid or medoid of each node or cluster, represented by the blue point. However, in our proposed method, the system is evaluated at any element within each node, represented by the orange point. This approach allows us to explore the overall representation of uncertainty in its discrete form and capture the finer details of the stochastic variables. By randomly selecting elements within each node, we can better exploit the richness of the uncertainty distribution and improve the robustness of the solutions.

In the forward step of the SDDP algorithm, we initially sample over the scenario tree, similar to the classic approach. In the proposed sampling technique, we conduct additional sampling within each node or cluster. This allows us to consider a broader representation of uncertainty by capturing more detailed variations within the node. By incorporating this subsequent sampling process, we aim to mitigate the impact of discretization and enhance the accuracy of the solution to the stochastic program.

In the backward step, we introduce a sampling process where we randomly select an element within each node of the scenario tree. Importantly, the element sampled in the backward step does not necessarily coincide with the element sampled in the forward step. By incorporating these additional samplings, we increase the number of possible system states considered in the forward and backward steps. This enables us to explore a broader range of scenarios and generate Benders cuts in more diverse system states, improving the robustness and accuracy of the solutions.

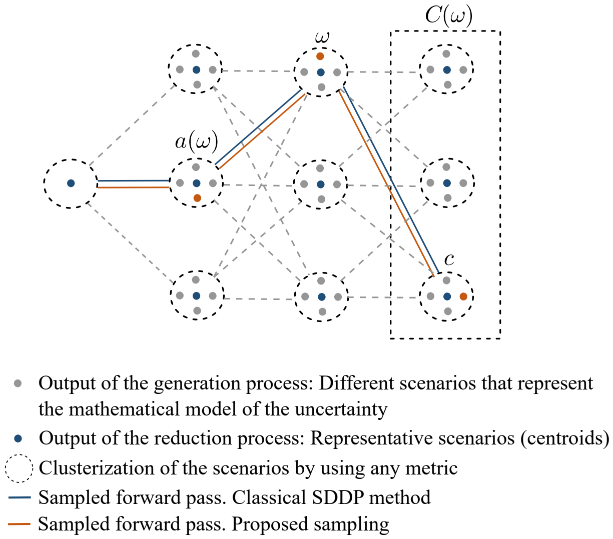
$C(\omega)$
ω
$a(\omega)$
c
● Output of the generation process: Different scenarios that represent the mathematical model of the uncertainty
● Output of the reduction process: Representative scenarios (centroids)
◌ Clusterization of the scenarios by using any metric
— Sampled forward pass. Classical SDDP method
— Sampled forward pass. Proposed sampling

Figure 2: Scenario tree

Appendix B provides a convergence proof for this proposed nodal sampling method.

3.2. Convergence criteria

The classical convergence criteria assess the quality of the solution in the first stage; however, more measures must be implemented to get an optimal policy on the whole horizon. In this way, we state two additional verifications to stop the algorithm:

- Extend the gap verification on the lower and upper bounds (equations 7 or 8) beyond the initial stage to all subsequent stages in every scenario tree node. This approach treats each node similarly to the first node. However, it is important to note that the initial conditions of the problem in nodes beyond the first one may exhibit significant changes during the initial iterations.

$$\left|\frac{UB_s^\omega - LB_s^\omega}{UB_s^\omega}\right| \le \epsilon,\ \forall \omega \in \Omega_\omega \tag{9}$$

- Establish a minimum number of Benders cuts in every node of the scenario tree (l_ω), which depends on its occurrence probability (p_ω). This criterion imposes a minimum representation of the cost-to-go function in every scenario tree node. This representation is more accurate in nodes with higher probability.

$$l^\omega \ge p^\omega \underline{L},\ \forall \omega \in \Omega_\omega \tag{10}$$

3.3. Improvements to SDDP algorithm

The inclusion of proposed nodal sampling and the other two stopping criteria enhance the resulting optimal policy, which will be used to simulate future scenarios in a planning context. Algorithm 3 shows the modifications to the SDDP approach (The numbering in Algorithm 3 is arranged to facilitate the integration of the proposed enhancements within Algorithm 2).

Algorithm 3: Improvements to SDDP algorithm

```
Data: Uncertainty in the form of scenario tree $(\Omega_\omega)$.
      Information about clusterized scenarios
[...]
while (stop == False) and ($l \in \mathcal{L}$) do
    // FORWARD STEP
    for $s \in \{1, \dots, S\}$ do
        [...]
        // sample a scenario tree node for the next stage
        $m \leftarrow \text{rand}(\Omega_\omega (s+1))$
        // sample a scenario inside the node
        $n \leftarrow \text{rand}(m)$
    // BACKWARD STEP
    for $s \in \{S, \dots, 1\}$ do
        for $\forall m \in \Omega_\omega (s)$ do
            // sample a scenario inside the node
            $n \leftarrow \text{rand}(m)$
            [...]
    // STOPPING CRITERIA VERIFICATION
    // first-stage criteria. Equation 7 or 8
    stop1 $\leftarrow LB \in [UB - k, UB + k]$
    // rest of stages criteria. Equation 9
    stop2 $\leftarrow LB_n \in \left[UB_n - k_n, UB_n + k_n\right]$
    // minimum Benders cuts criteria. Equation 10
    stop3 $\leftarrow l_n \geq p_n \underline{L}$
    // global criteria to stop the algorithm
    stop $\leftarrow$ stop1 & stop2 & stop3
    // increase iteration
    $l \leftarrow l + 1$
```

4. Case study and results

We used the hydrothermal scheduling problem as our experiment to test the SDDP algorithm improvements proposed in this work. This model is described in the next section.

4.1. Hydrothermal scheduling problem and uncertainty modeling

Our study employs a model of the Spanish electrical system, developed using information publicly available in sources Iberdrola (2006) and CEDEX. The case study exhibits the following characteristics:

- 4 basins, with 30 reservoirs and 38 hydropower plants,
- 58 thermal plants,
- 24 stages (24 months),
- 18 blocks of demand, wind and solar production per stage (month),
- 22 multidimensional historical time series, which represent natural water inflows into the reservoirs and served as the uncertain parameters in our case studies. Synthetic time series were generated using these historical datasets as a basis.

Basins

In this study, we present two basin models. The first model incorporates the reservoir topology, enabling a detailed representation of the energy interactions between hydro plants. The second model represents the basin as an equivalent hydro plant, which reduces the computational burden. This modeling approach is commonly applied to basins that are not the focus of the study or for which limited information is available. This work adopts the first modeling approach for the larger basins in the Spanish system, namely Duero, Tajo, and Sil (figure 3), while the second approach was applied to the remaining Spanish basins. The hydrothermal model presented in this study focuses on the individual reservoirs rather than the basins themselves. Therefore, there are no specific restrictions associated with the basins.

Reservoirs

In this study, we model the reservoirs of the Duero, Tajo, and Sil basins, considering their hydraulic topology Iberdrola (2006) and natural water inflows in each reservoir from 1995 to 2018 CEDEX. Figure 3 presents a schematic of the basins considered in this work.

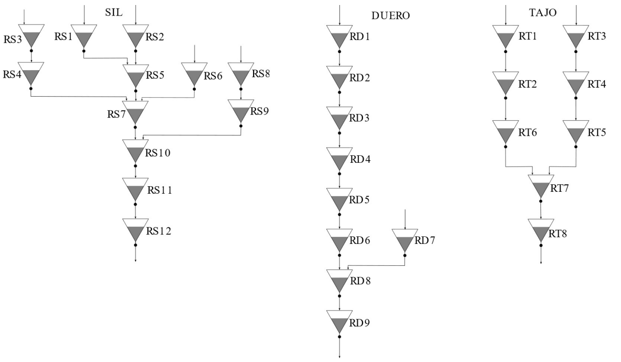

Figure 3: Topology of Sil, Duero, and Tajo basins

Figure 4 shows the interaction between two reservoirs connected in series. The outflows from the upstream reservoir are considered inputs for the downstream reservoir. Equation 11c describes the energy balance stored in each reservoir, considering the topology of the basin and the influence of pumping stations.

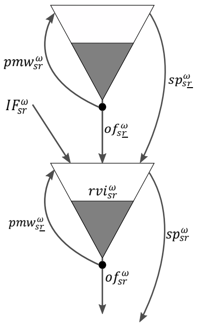

Figure 4: Reservoir modeling

The data structure of the water inflows in CEDEX requires subtracting the outflows from upstream reservoirs from the water input to each reservoir to calculate the natural water inflows. Figure 5 shows the annual uncertainty in natural water inflows in the Duero, Sil, and Tajo basins. The figure highlights that there is significant variability in streamflows between January and June, as well as between November and December. In contrast, natural water inflows during the summer are low, with minimal variability.

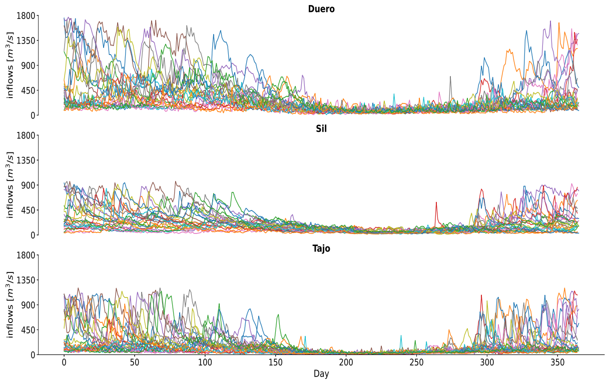

Figure 5: Natural water inflow uncertainty. Historical data

An essential step to obtaining and validating the operational policy is to build a training and testing set using historical data. The training set is used to construct the scenario tree, which is utilized in optimization. On the other hand, the testing set is employed to evaluate the performance of the operational policy through simulation. Simulation holds significant importance within the stochastic optimization framework as it allows for testing the policy's robustness across various potential realizations of stochastic variables. This emphasizes the significance of achieving convergence across all nodes within the scenario tree, resulting in a more accurate policy.

Section 2 describes a methodology for obtaining a scenario tree from historical data. Every step of the methodology is shown below:

- *Standardization:* Each time series is standardized by subtracting the mean and dividing by the corresponding standard deviation. The time series are now comparable in magnitude.
- *Variance-stabilizing transformation:* We employed the probability-distribution transformation for this case study to stabilize the time series. Among the different functions considered for our dataset, the generalized inverse Gaussian distribution yielded the best Akaike Information Criterion (AIC score) for the time series in this study. The positive domain of this function makes it suitable for modeling natural water inflows. Figure 7, blue curve, shows the time series after executing standardization and variance-stabilizing steps. These time series exhibit seasonal components that must be considered during the scenario generation process.
- *Seasonal pattern. Fourier Analysis:* In contrast to previous studies that utilized the Fourier series to represent the seasonal pattern, we decomposed each time series using a moving frame approach. In this case, the frame length was set to one year, with the sine-cosine basis having a period of 365 or 366 days, depending on whether it is a leap year or not. We analyzed the time series with thirty (30) harmonics. Each harmonic corresponds to a time window calculated as the period divided by the harmonic number. To illustrate, harmonic 1 spans 365 days (representing annual behavior), harmonic 2 spans 182.5 days (semester behavior), and harmonic 30 spans 12 days (approximately a week and a half). This array of harmonics spans a timeframe from around a week and a half to a full year.
Although the analysis was performed with 30 harmonics, we filtered out the most representative harmonics based on their energy over one cycle (as shown in equation 2). Figure 6 illustrates the harmonics for each time series and their corresponding energy measured in per unit. We use the harmonic with the highest energy as the basis to convert the values. Figure 7 shows the time series after executing standardization and variance-stabilizing steps and their Fourier approximation.

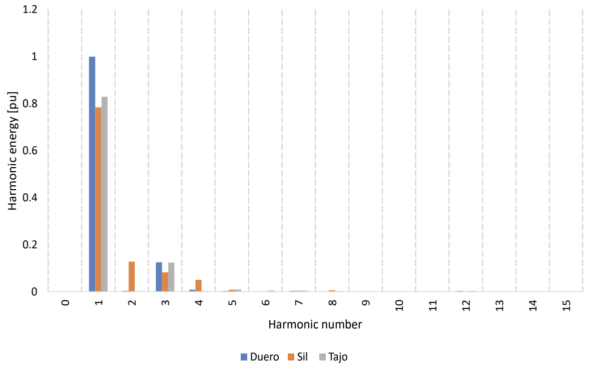

Figure 6: Harmonic energy: This plot illustrates the energy associated with each harmonic. For clarity, only the initial fifteen harmonics are displayed. The remaining harmonics exhibit energy levels close to zero. The constant component is omitted from the figure

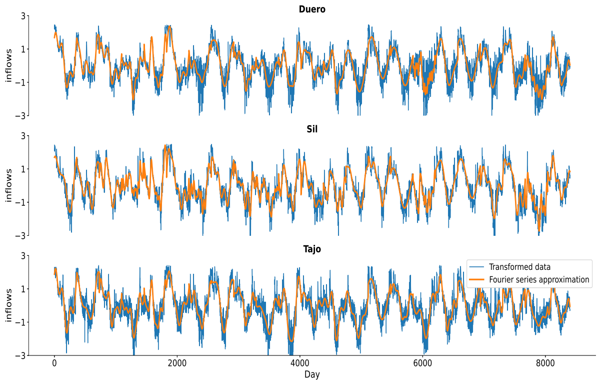

Figure 7: Blue curve: Historical data after standardization and variance-stabilizing steps. Orange curve: Fourier approximation considering 4 harmonics

Each significant harmonic is represented by a multivariate Gaussian distribution, taking into account the mean and covariance matrix of all frames. This modeling approach allows us to generate diverse seasonal trends while considering the correlation between the time series. On average, across all frames, four harmonics are used to represent the seasonal pattern of each time series.
To highlight the advantages of our proposed methodology, Figure 8 compares the annual harmonic of the seasonal pattern using both the static Fourier series and the windowed Fourier series. Using the former approach, we obtain a simplified representation of the seasonal pattern. The variability in the generated scenarios stems solely from the model of the residuals resulting from subtracting the Fourier approximation of the historical data, with the seasonal pattern remaining consistent across all scenarios. In contrast, the windowed Fourier approach allows us to derive a probabilistic model of the seasonal pattern, resulting in more nuanced scenarios. This approach introduces additional variability into the generated scenarios, offering a more realistic representation of the seasonal pattern.

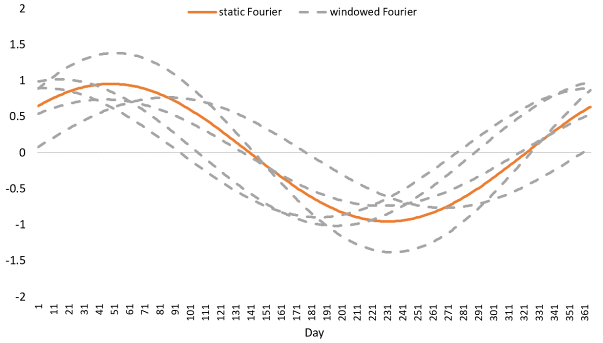

Figure 8: Comparison of annual harmonic in the seasonal pattern: Static Fourier vs. Windowed Fourier

- *Residuals VAR model:* The resulting residuals can be adjusted using a linear model, such as the ARIMA models family, or non-linear models, like neural networks. In this study, we employ a vector autoregressive (VAR) model to capture the correlations among the time series. Figure 9 shows the residuals obtained by subtracting the Fourier approximation from the transformed historical data. Figure 10 and 11 display the autocorrelation and partial autocorrelation plots of the residuals. Upon analysis of these plots, the order of the VAR model is approximately 7. This inference is based on the presence of values exceeding the confidence interval for lags around 7 in the partial autocorrelation plot.
However, in this work, we treat the order of the VAR model as a configurable hyperparameter. To address this, we employ a grid search approach to identify an appropriate value for this hyperparameter Brownlee (2017). The methodology is summarized as follows:

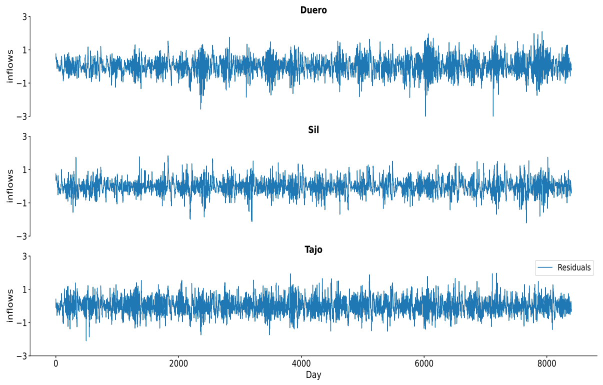

Figure 9: Residuals obtained by subtracting the Fourier approximation from the transformed historical data

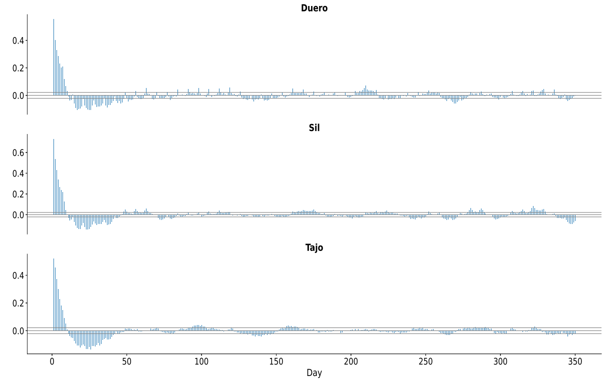

Figure 10: Autocorrelation residuals plot

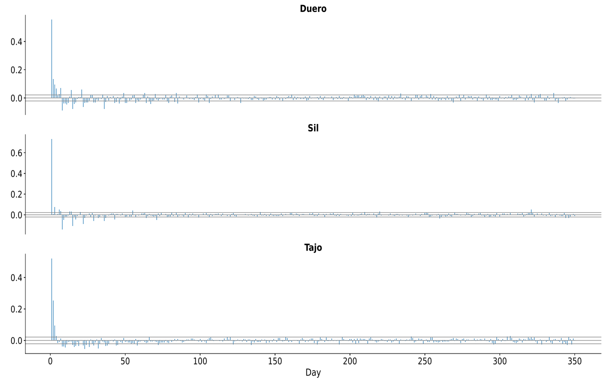

Figure 11: Partial autocorrelation residuals plot

  – Split the dataset into training and test sets. We divided our dataset into 80% for the training set and 20% for the test set.
  – For each value assigned to the hyperparameter, we generated the VAR model using the training set and computed the prediction error on the test set.
  – Selecting the order of the VAR model that results in the lowest prediction error in the test set.

In our study, the grid search for p (representing the order of the VAR model) covers a range from 1 to 20. As described in the methodology, the optimal result is achieved when p is set to 8, as within this range, the error reduction is particularly pronounced. The decrease in error becomes less significant for higher-order values. For additional details, you can refer to Appendix C, which provides matrices corresponding to the first and last lag of the model.

- *Generation of synthetic series:* The stochastic models for the seasonal trend, residuals, and errors allow us to generate synthetic time series by sampling from each model and aggregating the resulting components. After generating the time series, we apply inverse transformations to restore the original distribution and scale. Since each component considers the correlation of the time series, the synthetic time series generated preserve both spatial and temporal correlations. In this study, we generated 2000 synthetic series, of which 1000 were used to construct scenario three for the optimization process. The remaining time series were utilized in the simulation process to evaluate the solution obtained. Figure 12 shows some synthetic time series.
  Evaluating the quality of artificially generated time series compared to historical data is a key step when assessing time series generation models. Several commonly employed approaches for this purpose include:
  - **Visualization:** A well-generated artificial series should visually resemble the actual series.
  - **Descriptive Statistics:** Calculate fundamental descriptive statistics for all series, such as mean and standard deviation, and compare these statistics between the artificial and actual series. Similar statistics indicate that the artificially generated series captures essential characteristics of the real series.
  - **Autocorrelation Functions:** Calculate and compare autocorrelation functions for all series. A well generated artificial series should exhibit autocorrelation patterns that resemble those of the actual series.
  - **Model Fitting:** Fit statistical models to the real and artificially generated series and evaluate how well these models fit the actual and generated data. A good fit suggests that the artificial series shares structural similarities with the real ones. It is important to note that the distribution to be fitted should be multidimensional in space and time, considering the spatial and temporal correlations in the time series. This approach requires fitting a multivariate probability distribution for the actual and synthetic datasets and computing a divergence measured, for example, using the Kullback–Leibler (KL) distance. This method was applied in Yildiran (2019), using the k nearest neighbor (k-NN) approximation of density functions Wang et al. (2009).

  In our present research, we employ the first and second approaches. Figure 12 demonstrates a selection of synthetic time series that exhibit a striking resemblance to the actual data. In Figure 13, we observe that the synthetic data successfully replicate the mean and variation range of the actual series. These synthetic time series can be effectively utilized as training and test sets within the framework of the stochastic optimization problem.

Table 1
Statistics. Actual vs. synthetic series.
act.: actual series
syn.: synthetic series
sd: standard deviation

Parameter	Duero		Sil		Tajo	
	act.	syn.	act.	syn.	act.	syn.
mean $[m^3/s]$	279	266	189	187	160	158
sd $[m^3/s]$	293	246	170	164	207	196
skewness	2.3	1.9	1.9	1.9	2.5	2.5
kurtosis	5.7	4.6	3.5	4.2	6.3	7.7

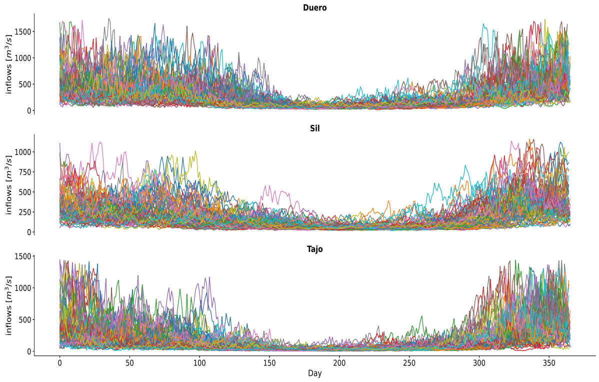

Figure 12: Synthetic natural water inflows

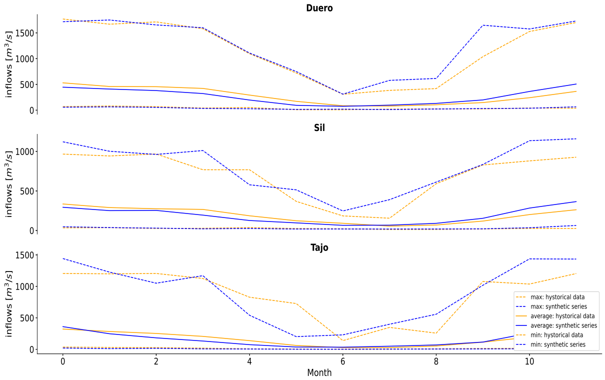

Figure 13: Comparing averages and variability: Actual vs. Synthetic data

Table 4.1 presents the first four moments for the multivariate actual and synthetic data. Although we can observe some differences, the resulting model is well-suited for representing uncertainty within a stochastic optimization problem. It allows us to generate scenarios with diverse characteristics derived from historical data. Consequently, the model can produce wet, average, and dry scenarios, incorporating seasonal changes closely resembling those observed in historical data.

- Generation of scenario tree: We employed the k-means method using n-dimensional points to generate the recombining scenario tree. The 1000 synthetic series were clustered at each stage, where each cluster represents a node in the scenario tree. The transition probability between two nodes was calculated by taking the ratio of the number of series belonging to the initial and final nodes simultaneously to the total number of series. The node probability was computed as the ratio of the number of series belonging to a specific node to the total number of series.

Figure 14 illustrates the recombining scenario tree, considering 6 nodes per stage and 24 stages. This scenario tree represents $6^{23} \approx 10^{18}$ possible scenarios.

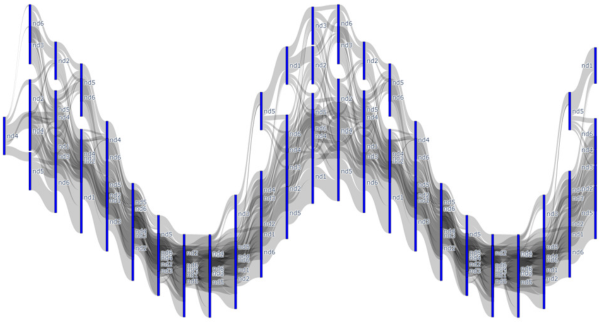

Figure 14: Recombining scenario tree. 6 nodes per stage

Hydro plants

Hydro plants are facilities designed to harness the energy of flowing water, typically from rivers or reservoirs, and convert it into electricity. These power stations utilize the kinetic and potential energy of water to rotate turbines, generating electrical energy through electromechanical conversion. The production of energy is proportional to the water outflows, with the proportionality function depending on the head of the dam. In this work, we consider this function as a constant term calculated based on the maximum water level of the dam. This consideration does not affect the analysis of the impact of our proposals, as we use the same model for both the base case and the case studies with our proposals. For further details on how the dependency of the water head is considered in hydro energy production, refer to Cerisola et al. (2012).

Thermal power plants

The thermal power plants convert heat energy into electric energy. In this study, we utilized data from the publication ESIOS - Red Eléctrica de España (date of access: 2023), associated with thermal plants in Spain. The cost function for thermal production can be modeled using different heat consumption curves (linear, piecewise linear) and multiple fuel types Ramos et al. (1999). However, for simplicity, we used a linear function to represent the variable cost of thermal production. Equation 11a represents the variable cost associated with the production of thermal power plants.

4.2. Mathematical model

We present a stochastic optimization model that minimizes the operational cost within the system. The model incorporates constraints related to demand balance considering renewable resources, water balance in reservoirs (incorporating both physical and operational limits), reservoir boundaries, the conversion of water to electrical energy, and the natural water inflows uncertainty. A detailed mathematical formulation of the hydrothermal scheduling problem is presented in Equations 11.

Nomenclature

Sets (calligraphic)

$\mathcal{B}_s$ Set of demand blocks in stage s. $b \in \mathcal{B}$

$\mathcal{H}$ Set of hydro plants. $h \in \mathcal{H}$

$\mathcal{P}$ Set of pumping plants. $p \in \mathcal{P}$

$\mathcal{R}$ Set of reservoirs. $r \in \mathcal{R}$

$\mathcal{R}_{do}$ Set of reservoirs downstream of reservoir r

$\mathcal{R}_{up}$ Set of reservoirs upstream of reservoir r

$\mathcal{S}$ Set of stages. $s \in \mathcal{S}$

$\mathcal{T}$ Set of thermal plants. $t \in \mathcal{T}$

$\mathcal{H}_r$ Set of tuples (h, r) that relate each hydro plant h to its corresponding reservoir r

$\mathcal{P}_r$ Set of tuples (p, r) that relate each pumping plant p to its corresponding reservoir r (Where the pumping is directed to)

Ω_s Set of scenarios in stage s. $\omega \in \Omega$

Parameters (uppercase)

DE^{ω}_{sb} Demand in block b and scenario ω of stage s. $[MW]$

DU_{sb} Duration of each demand block b in stage s. $[h]$

EC_h Energy coefficient of hydro plant h. $[kWh/m^3]$

IF^{ω}_{sr} Natural water inflows in reservoir r and scenario ω of stage s. $[hm^3]$

PN Penalty cost. $[€/MW]$

PR^{ω}_{s} Probability of scenario ω in stage s. $p.u.$

RVI_r Initial reserve level of reservoir r. $[km^3]$

SP^{ω}_{sb} Solar production. $[MW]$

TC_{sbt} Thermal cost of each thermal plant t in demand block b and stage s. $[€/MW]$

WP^{ω}_{sb} Wind production. $[MW]$

$\underline{GH}_h$, $\overline{GH}_h$ Lo. and Up. bound of each hydro h. $[MW]$

$\underline{GT}_t$, $\overline{GT}_t$ Lo. and Up. bound of each thermal t. $[MW]$

$\underline{PM}_p$, $\overline{PM}_p$ Lo. and Up. bound of each pumping p. $[MW]$

$\underline{RV}_{sr}$, $\overline{RV}_{sr}$ Lo. and Up. bound of each reservoir r. $[hm^3]$

Variables (lowercase)

aif^{ω}_{sr} Slack variable to avoid infeasibilities caused by violations in the lower bound of reservoir r in scenario ω of stage s. $[km^3]$

cur^{ω}_{sb} Curtailment of energy in scenario ω and demand block b of stage s. $[MW]$

gh^{ω}_{sbh} Production of a hydro plant h in scenario ω and demand block b of stage s. $[MW]$

gt^{ω}_{sbt} Production of a thermal plant g in scenario ω and demand block b of stage s. $[MW]$

nse^{ω}_{sb} Non-supplied energy in scenario ω and demand block b of stage s. $[MW]$

of^{ω}_{sr} Water outflows from a reservoir r in scenario ω of stage s. $[km^3]$

pm^{ω}_{sbp} Pumping of a pumping plant p in scenario ω and demand block b of stage s. $[MW]$

pmw^{ω}_{sr} Pumped water from a reservoir r to another reservoir in scenario ω of stage s. $[km^3]$

rv^{ω}_{sr} Final reserve level of a reservoir r in scenario ω of stage s. $[km^3]$

rvi^{ω}_{sr} Initial reserve level of a reservoir r in scenario ω of stage s. $[km^3]$

sp^{ω}_{sr} Spillage from a reservoir r in scenario ω of stage s. $[km^3]$

$$\begin{aligned}\min \quad & \sum_{s}\sum_{\omega\in\Omega_s}\sum_{b\in\mathcal{B}_s} \\ & PR^{\omega}_{s} DU_{sb}\left[\sum_{t} TC_{sbt} gt^{\omega}_{sbt} + PN\left(nse^{\omega}_{sb}\right)\right] \\ & + PR^{\omega}_{s} PN \sum_{r} aif^{\omega}_{sr}\end{aligned} \tag{11a}$$

s.t.

$$\begin{aligned}&\sum_{t} gt^{\omega}_{sbt} + \sum_{h} gh^{\omega}_{sbh} + SP^{\omega}_{sb} + WP^{\omega}_{sb} + nse^{\omega}_{sb} = \\ &\qquad DE^{\omega}_{sb} + cur^{\omega}_{sb} + \sum_{p} pm^{\omega}_{sbp}, \quad \forall(s,\omega,b)\end{aligned} \tag{11b}$$

$$\begin{aligned}&rv^{\omega}_{sr} = rvi^{\omega}_{sr} + IF^{\omega}_{sr} - of^{\omega}_{sr} - sp^{\omega}_{sr} - pmw^{\omega}_{sr} + aif^{\omega}_{sr} + \\ &\sum_{\underline{r}\in\mathcal{R}_{up}(r)}\left(sp^{\omega}_{s\underline{r}} + of^{\omega}_{s\underline{r}}\right) + \sum_{\underline{r}\in\mathcal{R}_{do}(r)} pmw^{\omega}_{s\underline{r}}, \quad \forall(s,\omega,r)\end{aligned} \tag{11c}$$

$$\begin{aligned}&rvi^{\omega}_{sr} = rv^{a(\omega)}_{(s-1)r}, \quad \forall(s\neq\{1\},\omega,r) \\ &rvi^{\omega}_{sr} = RVI_{r}, \quad \forall(s=\{1\},r)\end{aligned} \tag{11d}$$

$$of^{\omega}_{sr} = \sum_{h\in\mathcal{H}_r}\sum_{b\in\mathcal{B}_s} DU_{sb} gh^{\omega}_{sbh}/EC_{h}, \quad \forall(s,\omega,r) \tag{11e}$$

$$pmw^{\omega}_{sr} = \sum_{p\in\mathcal{P}_r}\sum_{b\in\mathcal{B}_s} DU_{sb} pm^{\omega}_{sbp}/EC_{p}, \quad \forall(s,\omega,r) \tag{11f}$$

$$\underline{GT}_{t} \le gt^{\omega}_{sbt} \le \overline{GT}_{t}, \quad \forall(s,\omega,b,t) \tag{11g}$$

$$\underline{GH}_{h} \le gh^{\omega}_{sbh} \le \overline{GH}_{h}, \quad \forall(s,\omega,b,h) \tag{11h}$$

$$\underline{PM}_{p} \le pm^{\omega}_{sbp} \le \overline{PM}_{p}, \quad \forall(s,\omega,b,p) \tag{11i}$$

$$\underline{RV}_{r} \le rv^{\omega}_{sr} \le \overline{RV}_{r}, \quad \forall(s,\omega,r) \tag{11j}$$

$$\underline{RV}_{sr} \le rv^{\omega}_{sr} \le \overline{RV}_{sr}, \quad \forall(s,\omega,r) \tag{11k}$$

$$0 \le cur^{\omega}_{sb} \le SP^{\omega}_{sb} + WP^{\omega}_{sb}, \quad \forall(s,\omega,b) \tag{11l}$$

$$0 \le nse^{\omega}_{sb} \le DE^{\omega}_{sb}, \quad \forall(s,\omega,b) \tag{11m}$$

Equation 11a defines the thermal cost function that needs to be minimized across all scenarios and stages. This function incorporates various components, such as penalties for non-supplied energy and violations to lower bounds of reservoirs. Artificial water inflows occur when the water level of any reservoir falls below the allowed bound. The penalty cost is around four times the average cost of thermal plants. 11b is the demand balance equation; 11c and 11d represents the water balance in every reservoir and the initial reserve in every stage; 11e is the energy conversion in the hydro plants; 11 represents the pumping process; 11g to 11m are the bounds of every variable. In this model, we consider two types of reservoir level limits: operational guiding determined by long-term requirements, specific seasonal irrigation needs, agreements with other stakeholders, and other factors, and physical limits defined by the technical characteristics of the dams.

The model exhibits what is known as *relatively complete recourse*, meaning that for any feasible first-stage solution, the second-stage problem remains feasible Rockafellar and Wets (1976). This property is maintained through slack variables such as non-supplied energy (nse^{ω}_{sb}), production curtailment (cur^{ω}_{sb}), artificial water inflows (aif^{ω}_{sr}), and spillage (sp^{ω}_{sr}).

4.3. Evaluation and Comparative Analysis of SDDP Proposals and Classic Algorithm

We propose the following methodology to evaluate the quality of policies generated by both the standard Stochastic Dual Dynamic Programming (SDDP) algorithm and the proposed algorithms:

- *Performance Metric:* To measure the effectiveness of the generated policies, we assess how well they help address issues within the system. These issues include situations like insufficient energy supply and deviations from the recommended reservoir levels set resulting from other analyses. Additionally, we analyze the impact of increasing thermal energy production, a key indicator of the system's overall cost implications. All measurements are presented consistently as per unit ratios, which are standardized against the baseline results from the cl1 case. The performance index is represented by the ratio of the violation variable to thermal production. Our analysis includes both average values and those at the 95th percentile. The latter is specifically considered to understand how policies perform in challenging situations. This work involves 1,000 distinct natural water inflow simulations, ensuring a thorough evaluation across a broad spectrum of hydrological conditions.
- *Case studies:* In our reference case study, we consider the centroid of each scenario-tree node (cluster) as the representative natural water inflows in the stochastic scheduling problem. For the subsequent case studies, the water inflows are selected from points within each node (cluster). These elements are chosen randomly in each iteration of the forward and backward steps, following the Algorithm 3.

- *Policy Evaluation:* To evaluate the performance of the generated policies, we employ both the standard SDDP algorithm and the proposed algorithms to obtain the operating policy for each test case. Subsequently, we execute simulations using these policies to assess their performance according to the defined metrics. The simulations are performed using the test set comprising 1000 synthetic series.
- *Statistical Analysis:* We focus on the average and 95th percentile of the simulation results to compute our performance metric. Considering the 95th percentile, we capture the high-impact violations and evaluate the policy's performance under worst-case conditions.
- *Sensitivity Analysis:* To evaluate the robustness of the generated policies, we perform sensitivity analyses by introducing changes to various parameters or assumptions. In this context, we propose to repeat the analysis employing a recombining scenario tree with four nodes per stage for the base case (cl1) and the case with our proposals (sm123).

  This sensitivity analysis enables us to investigate the influence of our proposed algorithms under different levels of uncertainty granularity and scenario tree structures, providing valuable insights into the stability and adaptability of our approach.
- *Decision Support:* Finally, we provide decision support and insights based on the evaluation results. We compare the performance of the algorithms and identify their strengths and weaknesses.

Table 2
Case studies definition

Case study	Description
cl1	This case solves the hydrothermal scheduling problem with the classical SDDP method (sampling in the reduced scenario tree). The stopping criterion is related to convergence in the first stage (the algorithm stops when the LB is included in the confidence interval of the UP)
cl12	Similar to the previous case (cl1). A new stopping criterion is added: convergence in every node of the scenario tree (equation 9)
cl123	Similar to the previous case (cl12). A new stopping criterion is added: minimum number of Benders cuts in every node according to its probability in the scenario tree (equation 10)
sm1	This case solves the hydrothermal scheduling problem with the SDDP method and nodal sampling (section 3.1). The stopping criteria are the same to case cl1
sm12	Similar to the previous case (sm1). The stopping criteria are the same to case cl12
sm123	Similar to the previous case (sm12). The stopping criteria are the same to case cl123

4.3.1. *Case studies*

Table 2 summarizes the case studies analyzed in this work. Note that case studies starting with "cl" use the classic SDDP algorithm, and "sm" denote cases using the nodal sampling method. Also, the suffix indicates when the algorithm uses the classic stopping criterion (1), the classic criterion plus convergence at each node (12), or uses all proposed criteria (123)

4.3.2. *Policy Evaluation. Statistical analysis*

Convergence

Figure 15 presents the convergence of the first-stage solution in each case study. The reference case (case cl1) achieved convergence in 51 iterations using the standard stopping criterion. Similarly, the proposed nodal sampling technique (case sm1) also converged in the same number of iterations when employing the same stopping criterion.

The plots corresponding to our sampling proposal in Figure 15 show a crossing point between the lower and upper bounds. This phenomenon can be attributed to the generation of Benders cuts during the backward step in each iteration. Specifically, a distinct point within each cluster is sampled in every backward path to create a Benders cut. These samples can represent either more or less restrictive operations, generating more or less restrictive Benders cuts.

As iterations progress, it becomes possible, within the same node of the scenario tree, to obtain Benders cuts from samples that lead to less restrictive operations, rendering some cuts redundant, or to obtain cuts from samples that produce more restrictive operations, creating dominant cuts. Since the lower bound is calculated based on the current stage variables along with an estimation of the future derived from the dominant Benders cuts, it's possible to overestimate the future compared to the upper bound in a forward path that uses less restrictive samples.

The lower bound is constructed using a future estimation based on the dominant Benders cuts. In contrast, the upper bound is created using actual values, which may take favorable values during sampling. In the classic SDDP algorithm, the same sample is evaluated in every node of the scenario tree across iterations, avoiding the emergence of dominant Benders cuts and, consequently, the crossing between bounds.

In the asymptotic case, our sampling method converges after evaluating all samples within each node, allowing the identification of the most restrictive Benders cuts. Thanks to this property, our sampling method yields a more robust operational policy, maintaining the performance of the classic algorithm.

Table 3 shows that when applying the new stopping criteria to the SDDP algorithm (cases cl12, cl123, sm123), the number of iterations increased within a range of 6 (as observed between case sm123 and sm1) to 28 (as seen between case cl123 and cl1). This increase was anticipated

Table 3
Case studies performance

Case study	Iterations	Time [s]	OF [M€]
cl1	51	4284	10.082068
cl12	71	5964	10.082318
cl123	79	6636	10.082331
sm1	51	4284	10.698785
sm12	51	4284	10.698785
sm123	57	4788	10.707610

since the convergence in scenario-tree nodes other than the first one is being evaluated.

It is worth mentioning that the algorithm incorporates the stopping criteria after the first 50 iterations to ensure an adequate number of samples for calculating reliable confidence intervals. Additionally, the paths selected for the forward step remain consistent for both the standard and proposed nodal sampling algorithms throughout iterations. This implies that we employ the same set of random numbers for both sampling methods (common random numbers). This uniformity guarantees an accurate evaluation of how the choice between selecting the centroid or any value within the scenario-tree node influences the resulting policy, regardless of the specific path chosen.

Each iteration in the nodal sampling proposal incurs a computational burden comparable to that of the classical SDDP algorithm. The primary distinction lies in the additional sampling performed within each node in the proposed algorithm. However, the time spent on this sampling is negligible compared to the overall problem-solving time within the node.

On average, each iteration took 84 seconds to complete. Table 3 provides information on the number of iterations and the corresponding computational time for each case study. We can observe that the new stopping criteria introduce slight changes in the objective function. However, the most significant impacts occur in the operating policy derived from the stochastic optimization program. This is a consequence of the new stopping criteria assessing nodes other than the initial one. As this operating policy serves as input for conducting future simulations, the accuracy of this policy becomes increasingly important.

Performance metrics

Figure 16 presents the water level in the Duero basin, one of the largest basins in Spain. The water reserves obtained with the proposed nodal sampling technique are higher than those obtained with the reference technique. This difference can be attributed to the fact that the proposed technique evaluates natural water inflows that are more critical for the system than the mean (centroid) in each node of the scenario tree. Consequently, the operational policy generated by our proposals prepares the system for more scenarios, resulting in slightly higher water storage.

Figure 17 presents the empirical distribution functions of the bi-annual thermal production of the system for each case study based on the samples in the test dataset. As depicted

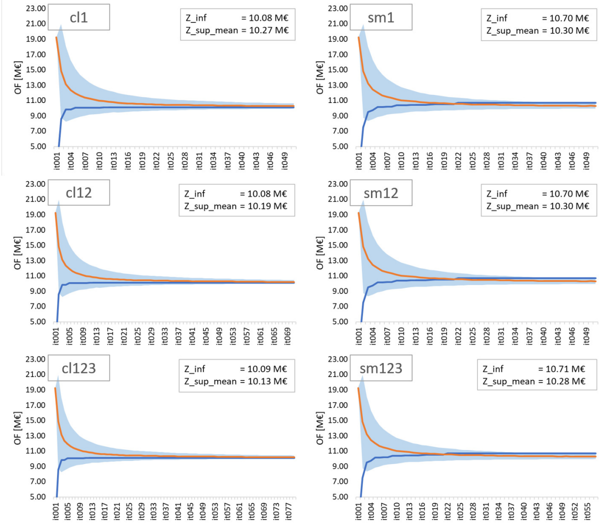

Figure 15: Convergence of the SDDP algorithm in the first stage

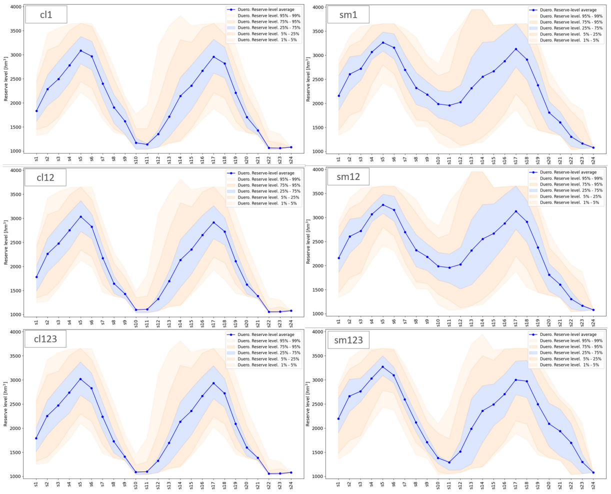

Figure 16: Volume level in Duero basin

in the figure, the thermal production of the system is around 279 TWh with a 95% probability. Figure 18 shows the empirical distribution functions of reservoir level violations in relation to the thresholds set by the operators. These thresholds encompass a combination of factors, including long-term requirements, specific seasonal irrigation needs, agreements with other agents, and additional considerations.

Both Figures 17 and 18 depict minimal variations in thermal production across the different policy scenarios. Nonetheless, a significant reduction in violations of reservoir thresholds is evident, primarily attributed to the sampling method we have introduced.

Table 4 shows a comprehensive overview of the performance of each operational policy within the test dataset. It presents the average and the value with a 95% probability for both bi-annual thermal generation and reservoir-threshold violations. The values in this table represent the ratios of the results for each parameter in the respective case study

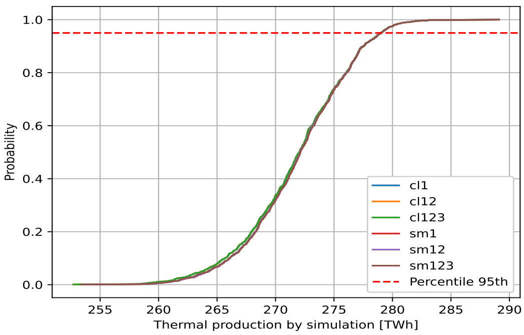

Figure 17: Empirical distribution of thermal production

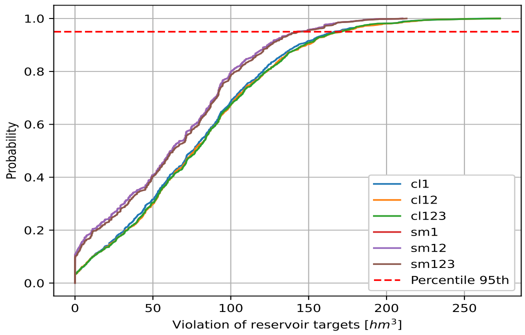

Figure 18: Empirical distribution of violations in reservoir threshold levels

Table 4
Summary of values for variables in the policy assessment

Case study	Thermal generation		Reservoir violation	
	mean	P95th	mean	P95th
cl1	1.0000	1.0000	1.0000	1.0000
cl12	1.0000	1.0000	1.0231	1.0245
cl123	1.0000	1.0000	1.0233	1.0268
sm1	1.0006	1.0001	0.8070	0.8536
sm12	1.0006	1.0001	0.8070	0.8536
sm123	1.0007	1.0001	0.8268	0.8646

compared to those obtained in the case cl1. In other words, each value represents the percentage (expressed per unit) of results in relation to the baseline case, cl1, which utilizes the standard method. For instance, the value 0.8536 located in the row *sm1* and columns *Reservoir violation*, *P95*, indicates that in the quantile 95, the violation in reservoir thresholds is 0.8536 times the violation given in the base case.

Figure 19, presents the behavior of the performance metric defined previously. This metric represents the ratio of the reduction in the violation variables to the increase in thermal generation resulting from implementing the proposals discussed in this paper. This index shows a reduction of roughly 20% in violations of reservoir threshold levels while only incurring minor adjustments in thermal generation production.

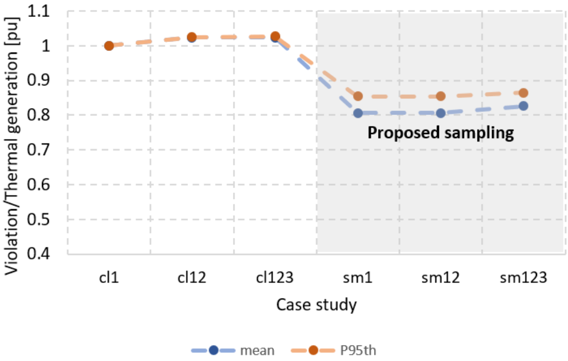

Figure 19: Evolution of the ratio between the reservoir threshold violations to the increase of thermal generation

4.3.3. *Sensitivity analysis*

In this sensitivity analysis, we apply our proposals within a more compact scenario tree. Specifically, we utilize a four-scenario tree, with each cluster encompassing more extreme scenarios. This is a result of a reduction in the number of nodes per stage when compared to previous cases. This evaluation focuses on assessing the operation of resources, computational performance, and, most importantly, the consistency of our proposals, as well as our proposals enhance the operational policy in extreme scenarios.

Convergence

Figure 20 shows the convergence of the standard SDDP algorithm and our proposed methods for the four-node scenario tree. The lower and upper bounds appear to closely mirror those achieved in the case of the six-node scenario tree. In terms of iteration count, in the base case (cl1), the number of iterations remains consistent with the previous section (six-node scenario tree). However, in the case study sm123, the problem was resolved in 52 iterations, five iterations fewer than in the six-node scenario tree. This is attributed to the more compact representation of uncertainty. The time required for each iteration remains in line with the description provided in the preceding section, approximately 84 seconds.

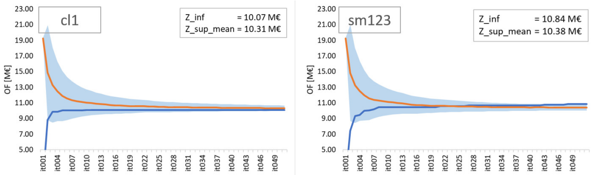

Figure 20: SDDP algorithm convergence in a four-node scenario tree

Performance metrics

Figures 21 and 22 exhibit a similar pattern as in Figures 17 and 18. Thermal production remains nearly identical when comparing the classic algorithm to our proposed approach. However, a noticeable reduction in reservoir threshold violations is evident. Thermal production and reservoir

Table 5
Four-Node vs. Six-Node Scenario Tree Comparison

Parameter	Four-node	Six-node
Objective Function [M€]	10.838	10.707
Iterations	52	57
Time $[s]$	4368	4788
Thermal generation P95th $[TWh]$	279	272
Reservoir violation P95th $[hm^3]$	64	65

violation values demonstrate similarities between the four- and six-node scenario trees.

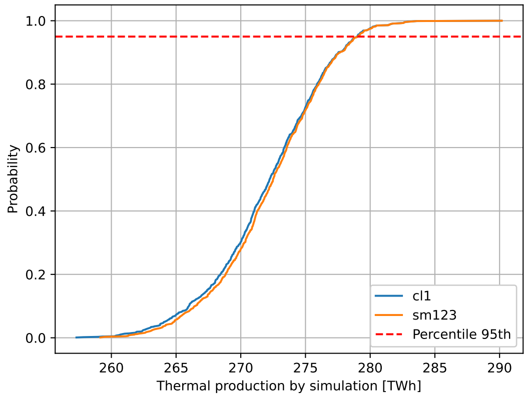

Figure 21: Empirical distribution of thermal production. Four-node scenario tree

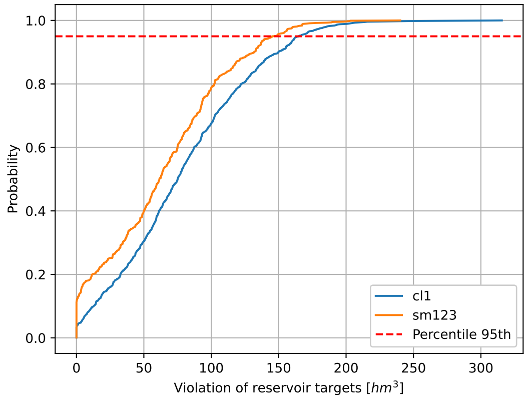

Figure 22: Empirical distribution of violations in reservoir threshold levels. Four-node scenario tree

Four-Node vs. Six-Node Scenario Tree Comparison

Table 5 provides a summary comparing some parameters for the results obtained using our proposals with both four- and six-node scenario trees. These comparisons are performed in the case study sm123.

4.4. Discussion

4.4.1. About the uncertainty modeling

In this work, we have developed a time series model to represent uncertainty, focusing primarily on generating synthetic water inflow scenarios, as illustrated in Figure 12. Through visual inspection and analysis of descriptive statistics (Figure 13 and Table 4.1), the model demonstrates favorable performance, making it well-suited for modeling uncertainty within a stochastic program. In future research, it may be worthwhile to research alternative methods and metrics to assess the model's effectiveness in generating scenarios.

4.4.2. About decision support

The policy enhancement proposed in this study can have several impacts on various aspects of the algorithm. Below are some key elements that may be affected:

Computational burden: The proposed nodal sampling technique in this work introduces two additional sampling processes compared to the standard SDDP method, specifically one in the forward step and another in the backward step. While it is true that these other sampling processes incur in some additional computational burden, it is important to note that this burden is negligible compared to the time required for solving the linear program in each iteration. As a result, the impact of the added computational burden is minimal.

Iteration number: In the proposed nodal sampling technique, it is important to note that the algorithm does not guarantee a specific number of iterations equal to or less than the classical algorithm. However, in our case studies, we observed that the number of iterations in the proposal was never greater than those of the classical SDDP method. This can be attributed to several factors.

Firstly, the nodal sampling method preserves the structure of the scenario tree, including the same number of nodes and the same transition probabilities between nodes. This helps to maintain a similar overall iterative process.

Secondly, the samples of uncertain variables evaluated within each node in the proposed nodal sampling technique can potentially generate more restrictive Benders cuts than those generated by the centroid. These more restrictive cuts can lead to faster convergence of the algorithm.

It is important to mention that including complementary stopping criteria can increase the number of iterations. These stopping criteria are adjustable in real applications to balance the trade-off between policy accuracy and computational burden. For example, the measure of statistical closeness between bounds in stages other than the first one (as defined in equation 9) can be implemented only in the initial stages where the impact of decisions is more significant. Similarly, the criterion of the minimum number of Benders cuts (as defined in equation 10) in each node of the scenario tree can be adjusted accordingly.

Solution quality and robustness: In the proposed nodal sampling technique, evaluating more samples of the uncertain variables enables us to obtain a more robust operating policy. By considering more samples, we can capture a wider range of potential scenarios and make more informed decisions.

The sampling method proposed in this work mitigates the effects of discretizing uncertainty (as reflected in the number of nodes in the recombining scenario tree). This is attributed to the retention of time series data from the training set after the clustering process, and they are considered in both the forward and backward steps of the SDDP algorithm for the recombining scenario tree. This capability enables us to consider a more simplified scenario tree without sacrificing solution accuracy. Using a compact scenario tree results in faster resolution of stochastic problems.

Contrary to algorithm performance concerns, the proposed nodal sampling can lead to faster convergence. This is because evaluating more samples allows us to identify critical scenarios for generating more restrictive Benders cuts. These cuts provide valuable information that helps improve the solution's quality and accelerate the convergence process.

Increasing policy costs: Considering a more detailed representation of uncertainty, the solution to the stochastic problem can result in a cheaper or more expensive policy, depending on the impact of uncertain parameters on the cost of the objective function. In the case studies, we can evidence a reduction between 16% to 20% in violation variables by marginal increases in thermal generation (compared to the base case, cl1). This indicates that the resulting operating policy from the proposed nodal sampling method is profitable.

5. Conclusions

This paper proposes an approach to model seasonal patterns in time series data using Fourier series decomposition by windows. Unlike traditional methods that assume fixed coefficients for the entire time series, our approach allows for the characterization of seasonal trends in different periods of the time series. This enables the generation of stochastic seasonal patterns, preserving the spatiotemporal correlations in the data. We can capture a broader range of possible system states and uncertainties by generating synthetic time series with stochastic seasonal patterns. The artificial time series are integrated into the SDDP framework, allowing for exploring larger solution space to develop more robust operating policies.

The proposed nodal sampling technique in the stochastic dynamic dual programming (SDDP) framework improves the robustness of the operating policy by considering more realizations of uncertain parameters. It allows for a more thorough solution-space exploration without significantly increasing the computational burden. This method maintains the performance of the standard SDDP algorithm while capturing the potential risks associated with more extreme uncertainties, resulting in more robust policies.

This sampling method also mitigates the effects of discretizing uncertainty (as reflected in the number of nodes in the recombining scenario tree). This capability allows us to employ a more streamlined scenario tree without compromising solution accuracy. Using a compact scenario tree results in faster resolution of stochastic problems.

The proposed complementary stopping criteria are useful in achieving more robust policies in stochastic programming optimization. By incorporating additional Benders cuts in all scenario-tree nodes, the representation of the cost-to-go function can be enhanced. Although this may increase the number of iterations, these criteria can be adjusted to balance solution quality and computational efficiency. As a result, these stopping criteria contribute to the generation of more robust policies without significantly impacting the overall performance of the algorithm.

In our case studies, we found that the proposed nodal sampling method required fewer iterations than the classical method. Additionally, we observed an improvement in policy quality, which we evaluated through simulations using the test dataset. To assess the accuracy of the stochastic policies, we conducted a statistical analysis of the violation variables resulting from constraint violations and the thermal production over the planning horizon. As an evaluation metric, we introduced a ratio that considers the change in the violation variables and the thermal production.

In future work, we aim to explore the modeling of the resulting residuals, obtained after subtracting the seasonal trend using Fourier series decomposition, with alternative techniques such as machine learning or other time series analysis methods. This will allow us to evaluate the performance of these techniques in capturing the remaining patterns in the data, particularly in high-dimensional spaces.

In addition, we plan to extend the proposed nodal sampling technique to a multi-horizon stochastic optimization framework, both at the strategic and operational levels. We can obtain more robust and accurate policies for complex systems by applying this sampling method in a multi-horizon context.

A. Appendix. Synthetic time series generation. Mathematical concepts

A.1. Standardization

The initial step in the process is to standardize each time series by subtracting the mean and dividing by the standard deviation. This procedure ensures that every time series has a mean of zero and a standard deviation of one, enabling comparability between all series.

$$\overline{\mathbf{Y}}(t) = \left[\frac{y_1(t) - \mu_1}{\sigma_1}, \dots, \frac{y_i(t) - \mu_i}{\sigma_i}\right] \tag{12}$$

where, $y_i(t)$ represents each dimension i of the multivariate space as a function of time, and μ_i and σ_i stand for the mean and the standard deviation of each time series i respectively.

A.2. Variance-stabilizing transformation

In contrast to Talbot et al. (2020), this study proposes to transform each non-stationary series into a stationary one as the next step. A strictly stationary time series (or stochastic process) is one whose properties or probabilities do not change over time, meaning its cumulative distribution function is independent of time. A time series is considered weakly stationary or of order 2 if only the moments up to order 2 (such as the mean, variance, and covariances) are finite and independent of time. Since time series theory assumes Gaussian processes (i.e., weakly stationary processes), detrending or transformation becomes necessary to make the data conform to this assumption. Below are some techniques to stabilize variance:

- **Delta method**: This method aims to find a functional transformation that ensures constant variance of the data. The underlying assumption is that the variance varies functionally with changes in the mean ($\sigma_i^2 = cf(\mu_i)$) Wei (2006). The functional transformation is obtained by solving the equation 13

$$T\left(y_i(t)\right) = \int \frac{1}{\sqrt{f(y_i(t))}} dy_i(t) \tag{13}$$

- **Box-Cox transformation**: The Box-Cox transformation, introduced by Box and Cox in 1964, involves finding the optimal parameter λ in Equation 14 that maximizes the symmetry and normality of the transformed distribution. Once this parameter is obtained, the data can be transformed by applying Equation 14

$$T\left(y_i(t)\right) = \frac{y_i^{\lambda}(t) - 1}{\lambda} \tag{14}$$

- **Probability distribution transformation**: In this approach, the original data representing a probability distribution function (PDF) are transformed into data with a new distribution. For time series, it is required that the distribution of the data be Gaussian, which ensures stability in mean and variance and satisfies the assumption of normality that is typical in time series theory. Equation 15 is used to transform the data, where $\mathcal{N}$ represents the inverse of the cumulative distribution function of the Gaussian and F represents the cumulative distribution function of the data. The determination of F can be achieved through statistical methods, either parametric or non-parametric.

$$T\left(y_i(t)\right) = \mathcal{N}^{-1}\left(F\left(y_i(t)\right)\right) \tag{15}$$

A.2.1. VAR model to residuals

Removing the seasonal trend (Equation 3) from the time series reduces the number of lags required to fit a parametric model. As a result, we can increase the complexity of a multivariate model while maintaining a low computational burden. Conversely, fitting a model with the original data would require a significantly higher number of lags, resulting in a more time-consuming model.

Conventional methods, such as VARMA models, neural networks, etc., can be applied since the residuals remain a multivariate time series. For this study, VAR models were utilized due to their ease of adjustment, enabling us to incorporate more lags. Equation 16 shows the model for the residuals.

$$\mathbf{r}_t = A_1\mathbf{r}_{t-1} + \cdots + A_p\mathbf{r}_{t-p} + \mathcal{N}_{\epsilon}\left(\mu_{\epsilon}, \Sigma_{\epsilon}\right) \tag{16}$$

where $\mathbf{r}_t$ denotes the residuals, Aj corresponds to matrices that establish correlations between the multivariate signals and samples at lag j of these signals, and $\mathcal{N}\epsilon\left(\mu\epsilon, \Sigma_{\epsilon}\right)$ represents a multivariate Gaussian model that characterizes the errors inherent to the VAR model.

A.2.2. Generation of synthetic series

The generation of synthetic series involves the following steps:

- Generating a sequence of residuals using the VAR model 16. This is done by recursively appending the calculated samples for each time step to the historical series for calculating the next sample.
- Generating a seasonal pattern using equations 1 and 4.
- Aggregating the seasonal and residual parts and applying inverse transformations to recover the original distribution functions, means, and variances.
- Repeating this process for the desired number of synthetic series to be generated.

B. Appendix. Convergence proof

To demonstrate the convergence of the proposed approach, let us consider a two-stage stochastic problem as shown in Equation 17

$$\min_{\mathbf{x}_1, \mathbf{x}_2^{\omega\xi}} \quad \mathbf{c}_1\mathbf{x}_1 + \sum_{\omega}\sum_{\xi} p_2^{\omega} p_{\omega}^{\xi}\left(\mathbf{c}_2\mathbf{x}_2^{\omega\xi}\right) \tag{17a}$$

$$\text{s.t.} \quad A_{11}\mathbf{x}_1 = \mathbf{b}_1 \tag{17b}$$

$$A_{21}\mathbf{x}_1 + A_{22}\mathbf{x}_2^{\omega\xi} = \mathbf{b}_2^{\omega\xi}, \quad \forall(\omega, \xi) \tag{17c}$$

where ξ is the index for each element within the cluster ω, and p_{ω}^{ξ} is the probability associated with each element.

Formulation 17 can be rewritten as the one shown in 18

$$\min_{\mathbf{x}_1, \mathbf{x}_2^{\omega\xi}} \quad \mathbf{c}_1\mathbf{x}_1 + \sum_{\omega} p_2^{\omega} Q^{\omega}(\mathbf{x}_1) \tag{18a}$$

$$\text{s.t.} \quad A_{11}\mathbf{x}_1 = \mathbf{b}_1 \tag{18b}$$

where $Q^{\omega}(\mathbf{x}_1)$ is,

$$Q^{\omega}(\mathbf{x}_1) = \min_{\mathbf{x}_2^{\omega\xi}} \quad p_{\omega}^{\xi}\sum_{\xi}\mathbf{c}_2\mathbf{x}_2^{\omega\xi} \tag{19a}$$

$$\text{s.t.} \quad A_{22}\mathbf{x}_2^{\omega\xi} = \mathbf{b}_2^{\omega\xi} - A_{21}\mathbf{x}_1, \quad \forall(\xi) \tag{19b}$$

As the problem $Q^{\omega}(\mathbf{x}_1)$ is separable, it can be represented as the sum of minimizing the costs of each individual ξ.

$$Q^{\omega}(\mathbf{x}_1) = p^{\xi}_{\omega} \sum_{\xi} \min_{\mathbf{x}_2^{\omega\xi}} Q^{\omega}_{\xi}(\mathbf{x}_1) \tag{20}$$

where $Q^{\omega}_{\xi}(\mathbf{x}_1)$ is

$$\begin{aligned} Q^{\omega}_{\xi}(\mathbf{x}_1) = \quad & \min_{\mathbf{x}_2^{\omega\xi}} \quad \mathbf{c}_2 \mathbf{x}_2^{\omega\xi} && \text{(21a)} \\ & \text{s.t.} \quad A_{22}\mathbf{x}_2^{\omega\xi} = \mathbf{b}_2^{\omega\xi} - A_{21}\mathbf{x}_1 && \text{(21b)} \end{aligned}$$

and its dual formulation is

$$\begin{aligned} Q^{\omega}_{\xi}(\mathbf{x}_1) = \quad & \max_{\pi_2^{\omega\xi}} \quad \left(\mathbf{b}_2^{\omega\xi} - A_{21}\mathbf{x}_1\right)^T \pi_2^{\omega\xi} && \text{(22a)} \\ & \text{s.t.} \quad A_{22}^T \pi_2^{\omega\xi} \leq \mathbf{c}_2 && \text{(22b)} \end{aligned}$$

For each Q^{ω}_{ξ}, the polytope $A_{22}^T \pi_2^{\omega\xi} \leq \mathbf{c}_2$ serves as the feasible region for the optimization problem. It can be noted that this polytope is the same for all the points belonging to the cluster ($A_{22}^T \pi_2^{\omega\xi} \leq \mathbf{c}_2 \rightarrow A_{22}^T \pi_2^{\omega} \leq \mathbf{c}_2$), so each one can be solved by enumeration of solutions, as shown in equation 23. Note that the superscript ξ is eliminated from π since, as mentioned before, the polytope is the same for each Q^{ω}_{ξ}. The superscript i denotes each extreme point of the polytope.

$$\begin{aligned} Q^{\omega}_{\xi}(\mathbf{x}_1) = \quad & \min_{\theta_2^{\omega}} \quad \theta_2 && \text{(23a)} \\ & \text{s.t.} \quad \theta_2^{\omega} \geq \left(\mathbf{b}_2^{\omega\xi} - A_{21}\mathbf{x}_1\right)^T \pi_2^{\omega i} \quad \forall(i) && \text{(23b)} \end{aligned}$$

Based on the above equations, a piecewise linear approximation can replace the future cost function for each node in the scenario tree. These equations are the Benders cuts, expressed as either equation 24 or 25. Equation 26 represents the master problem with the Benders cuts.

$$\theta_2^{\omega} \geq \left(\mathbf{b}_2^{\omega\xi} - A_{21}\mathbf{x}_1\right)^T \pi_2^{\omega i} \tag{24}$$

$$\theta_2^{\omega} \geq f_2^i + \pi_2^{\omega i} A_{21} \left(\mathbf{x}_1^i - \mathbf{x}_1\right) \tag{25}$$

$$\begin{aligned} \min_{\mathbf{x}_1, \mathbf{x}_2} \quad & \mathbf{c}_1\mathbf{x}_1 + \sum_{\omega} PR^{\omega}\theta_2^{\omega} && \text{(26a)} \\ \text{s.t.} \quad & A_{11}\mathbf{x}_1 = \mathbf{b}_1 && \text{(26b)} \\ & \theta_2^{\omega} \geq f_2^i + \pi_2^{\omega i} A_{21} \left(\mathbf{x}_1^i - \mathbf{x}_1\right), \quad \forall(\omega, i) && \text{(26c)} \end{aligned}$$

The SDDP algorithm iteratively computes Benders cuts for each dual variable in the second stage through a forward-backward process. The polytope of the dual formulation of the subproblem is finite and independent of the primal solution proposed by the master problem. In the worst-case scenario, all vertices may need to be calculated to find the Benders cuts representing future cost. This demonstrates that the algorithm proposed in this work is finite and converges to a solution.

It is worth mentioning that feasibility cuts can be avoided by introducing slack variables and penalizing them in the objective function. The penalty values must be chosen appropriately to ensure that the slack variables converge to zero by the end of the optimization process. This method allows only the utilization of optimality Benders cuts, eliminating the need for additional cuts to ensure feasibility in the problem formulation.

C. Appendix. VAR model matrices

In section 4.1, we fitted a VAR model of order 8 to the residuals obtained by subtracting the seasonal pattern from the historical data. The matrices for the first and last lag, as well as the covariance matrix for the errors, are presented below:

$$A_1 = \left[\begin{array}{cccc} & \text{Duero} & \text{Sil} & \text{Tajo} \\ \text{Duero} & 0.4111 & 0.0261 & 0.1067 \\ \text{Sil} & 0.1005 & 0.6942 & 0.0810 \\ \text{Tajo} & 0.1241 & 0.0592 & 0.3279 \end{array}\right] \tag{27}$$

$$A_8 = \left[\begin{array}{cccc} & \text{Duero} & \text{Sil} & \text{Tajo} \\ \text{Duero} & -0.0803 & 0.0354 & -0.0489 \\ \text{Sil} & -0.0549 & -0.1514 & -0.0190 \\ \text{Tajo} & -0.0234 & 0.0017 & -0.0223 \end{array}\right] \tag{28}$$

$$\Sigma_{\epsilon} = \left[\begin{array}{cccc} & \text{Duero} & \text{Sil} & \text{Tajo} \\ \text{Duero} & 0.1202 & 0.0119 & 0.0187 \\ \text{Sil} & 0.0119 & 0.0694 & 0.0148 \\ \text{Tajo} & 0.0187 & 0.0148 & 0.1408 \end{array}\right] \tag{29}$$

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT-3.5 in order to improve readability and language. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

CRediT authorship contribution statement

Jesús D. Gómez-Pérez: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing - Original Draft, Data Curation. **Jesus M. Latorre-Canteli:** Conceptualization, Supervision, Validation, Writing - Review and Editing. **Andres Ramos:** Conceptualization, Supervision, Validation, Writing - Review and Editing, Funding acquisition, Resources. **Alejandro Perea:** Conceptualization, Validation. **Pablo Sanz:** Conceptualization, Validation. **Francisco Hernández:** Conceptualization, Validation.

References

Aranha, A.S., Street, A., Fernandes, C., Granville, S., 2022. Risk-constrained optimal dynamic trading strategies under short-and long-term uncertainties. IEEE Transactions on Power Systems 38, 1474–1486.

Benders, J.F., 1962. Partitioning procedures for solving mixed-variables programming problems. Numerische mathematik 4, 238–252.

Brownlee, J., 2017. Introduction to time series forecasting with python: how to prepare data and develop models to predict the future. Machine Learning Mastery.

CEDEX, . Anuario de aforos 2018 - 2019. modalidades de consulta. URL: https://ceh.cedex.es/anuarioaforos/demarcaciones.asp.

Cerisola, S., Latorre, J.M., Ramos, A., 2012. Stochastic dual dynamic programming applied to nonconvex hydrothermal models. European Journal of Operational Research 218, 687–697.

De Matos, V.L., Philpott, A.B., Finardi, E.C., 2015. Improving the performance of stochastic dual dynamic programming. Journal of Computational and Applied Mathematics 290, 196–208.

Deb, C., Zhang, F., Yang, J., Lee, S.E., Shah, K.W., 2017. A review on time series forecasting techniques for building energy consumption. Renewable and Sustainable Energy Reviews 74, 902–924.

Dupačová, J., Consigli, G., Wallace, S.W., 2000. Scenarios for multistage stochastic programs. Annals of operations research 100, 25–53.

Epe, A., Küchler, C., Römisch, W., Vigerske, S., Wagner, H.J., Weber, C., Woll, O., 2009. Optimization of dispersed energy supply—stochastic programming with recombining scenario trees. Optimization in the Energy Industry , 347–364.

ESIOS - Red Eléctrica de España, date of access: 2023. Unidades de Programación - ESIOS. URL: https://www.esios.ree.es/es/unidades-de-programacion.

Fodstad, M., del Granado, P.C., Hellemo, L., Knudsen, B.R., Pisciella, P., Silvast, A., Bordin, C., Schmidt, S., Straus, J., 2022. Next frontiers in energy system modelling: A review on challenges and the state of the art. Renewable and Sustainable Energy Reviews 160, 112246.

Gielen, D., Boshell, F., Saygin, D., Bazilian, M.D., Wagner, N., Gorini, R., 2019. The role of renewable energy in the global energy transformation. Energy strategy reviews 24, 38–50.

Høyland, K., Kaut, M., Wallace, S.W., 2003. A heuristic for moment-matching scenario generation. Computational optimization and applications 24, 169–185.

Iberdrola, 2006. Grandes presas: 110 años del trabajo de Iberdrola. Iberdrola.

Latorre, J.M., Cerisola, S., Ramos, A., 2007. Clustering algorithms for scenario tree generation: Application to natural hydro inflows. European Journal of Operational Research 181, 1339–1353.

Marulanda, G., Bello, A., Cifuentes, J., Reneses, J., 2020. Wind power long-term scenario generation considering spatial-temporal dependencies in coupled electricity markets. Energies 13, 3427.

Homem-de Mello, T., De Matos, V.L., Finardi, E.C., 2011. Sampling strategies and stopping criteria for stochastic dual dynamic programming: a case study in long-term hydrothermal scheduling. Energy Systems 2, 1–31.

Morales, J.M., Minguez, R., Conejo, A.J., 2010. A methodology to generate statistically dependent wind speed scenarios. Applied Energy 87, 843–855.

Penna, D.D., Maceira, M.E.P., Damázio, J.M., 2011. Selective sampling applied to long-term hydrothermal generation planning, in: Power System Computation Conf.(PSCC), Stockholm, Sweden.

Pereira, M.V., Pinto, L.M., 1991. Multi-stage stochastic optimization applied to energy planning. Mathematical programming 52, 359–375.

Ramos, A., Ventosa, M., Rivier, M., 1999. Modeling competition in electric energy markets by equilibrium constraints. Utilities Policy 7, 233–242.

Roald, L.A., Pozo, D., Papavasiliou, A., Molzahn, D.K., Kazempour, J., Conejo, A., 2023. Power systems optimization under uncertainty: A review of methods and applications. Electric Power Systems Research 214, 108725.

Rockafellar, R.T., Wets, R.J., 1976. Stochastic convex programming: relatively complete recourse and induced feasibility. SIAM Journal on Control and Optimization 14, 574–589.

Sethi, S., Sorger, G., 1991. A theory of rolling horizon decision making. Annals of Operations Research 29, 387–415.

Shapiro, A., 2011. Analysis of stochastic dual dynamic programming method. European Journal of Operational Research 209, 63–72.

Talbot, P.W., Rabiti, C., Alfonsi, A., Krome, C., Kunz, M.R., Epiney, A., Wang, C., Mandelli, D., 2020. Correlated synthetic time series generation for energy system simulations using fourier and arma signal processing. International Journal of Energy Research 44, 8144–8155.

Vagropoulos, S.I., Kardakos, E.G., Simoglou, C.K., Bakirtzis, A.G., Catalao, J.P., 2016. Ann-based scenario generation methodology for stochastic variables of electric power systems. Electric Power Systems Research 134, 9–18.

Wang, Q., Kulkarni, S.R., Verdú, S., 2009. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. IEEE Transactions on Information Theory 55, 2392–2405.

Wei, W.W., 2018. Multivariate time series analysis and applications. John Wiley & Sons.

Wei, W.W.S., 2006. Time Series Analysis Univariate and Multivariate Methods. 2nd ed., Addison-Wesley.

Yildiran, U., 2019. Nonnegative wind speed time series models for sddp and stochastic programming applications .