

Multi-period forecasting and scenario generation with limited data

Ignacio Rios
University of Chile
Santiago, Chile

Roger J-B Wets and David L. Woodruff *
University of California, Davis
Davis, CA 95616, USA

October 28, 2013

Abstract

Data for optimization problems often comes from (deterministic) forecasts, but it is naïve to consider a forecast as the only future possibility. A more sophisticated approach uses data to generate alternative future scenarios, each with an attached probability. The basic idea is to estimate the distribution of forecast errors and use that to construct the scenarios. Although sampling from the distribution of errors comes immediately to mind, we propose instead to approximate rather than sample. Benchmark studies show that the method we propose works well.

Keywords: Stochastic Programming; Scenarios; Scenario Generation; Scenario Trees; Forecast Error Distributions

1 Introduction

Data for optimization problems often comes from (deterministic) forecasts, but it is naïve to consider a forecast as the only future possibility. A more sophisticated approach uses data to generate alternative future scenarios, each with an attached probability. So, instead of coming up with a specific forecast, one needs to come up with a “stochastic” description of a number of possible *scenarios*, each of which specifies the data for future sub-periods in the model. In fact, these could be a discretization of an extensive range of future possibilities. Sometimes enough historical data and accumulated model-building experience might be available to model the uncertainty as a standard stochastic processes

*Corresponding Author: DLWoodruff@UCDavis.edu

relying, for example, on a time series model or a stochastic differential system and then derive scenarios from a discretization of this stochastic process. But, it's not possible to proceed in this manner when the data are much too scarce to even consider following such a classical approach. As will be laid out later, it's exactly a problem of this nature that we faced in the context of predicting electricity loads (demand) for day D from essentially deterministic weather forecasts provided on day $D - 1$.

We will use the symbol l for the vector of the length T to be forecast using a vector of predictors w . A single forecast ($T = 1$) is special case, but we are primarily interested in situations where a forecast is made for some number of time steps in the future. We refer to the set of time steps $\{1, \dots, T\}$ as a *study period*. We denote the set of historic study periods as \mathcal{D} . Examples of demand forecasts based on leading indicators abound. A classic example is demand for spare parts based a vector of product sales [13]. Our interest in this paper is in situations where there is limited data for model fitting. The mechanisms for forming the forecasting function using historical values l^d and w^d for $d \in \mathcal{D}$ are described in the next section.

Example 1.1. Throughout the paper, we use as a motivating example the forecast of hourly electricity demand for a day (l of length $T = 24$ for a study period of one day) based on forecasts of the evolution of temperature a day (given as w). Appending additional predictors, such as forecasts for dewpoint, to w is a straightforward extension. \square

The basic idea is to estimate the distribution of forecast errors and use that to construct the scenarios. Although sampling from the distribution of errors comes immediately to mind, we propose instead to approximate rather than sample looking for the following properties. Smoothness: we assume that there are no discontinuities from l_t to l_{t+1} . Intertemporal error correlation: we consider contiguous subsets (perhaps not proper) of the study period.

There are two phases to scenario generation. In the first phase, historical data are used to estimate coefficients, using the following steps:

1. Perform a functional regression to obtain a forecast and a distribution of errors
2. Categorize and partition historic study periods
3. Regression curves computation for the categories
4. Estimation of error distributions

In practice, these phase one steps are executed “every so often” but presumably not every time scenarios are needed. The estimates provided are used to generate scenarios using forecasts as they become available. When scenarios for a future study period are needed and predictor data are available, scenarios are generated using these steps:

1. Scenario skeleton points computation
2. Scenario paths construction

The rest of the paper proceeds as follows. In Section 2 we describe the methods used for functional regression. Methods for partitioning the time steps within a study period and for categorizing study periods base on forecast errors are described in Section 3. The points that form the so-called “skeletons” of scenario are created using methods described in Section 4. The section goes on to describe the way these points are connected to form scenarios. In Section 5 we provide some computational benchmarks to illustrate that the method can be effectively deployed; this provides some contrast with sampling-based methods. The paper closes with conclusions and directions for further research.

2 Functional Regression and Error Densities

As outlined in [5], the idea is to find a function that maps w^d to l^d with minimal errors for $d \in \mathcal{D}$ using epi-splines [10]. To discretize, we divide the study period into N sub-intervals. Let $\delta = T/N$. If each study period is T time units long, then we can think of each $h \in 1, \dots, T$ as a “time” and δ as a fraction of a time unit when $N > T$, which is typical. Thus, we think of each of these sub-intervals as ending at time $t_k = k\delta$ for $k = 1, \dots, N$. Implicitly, our construction assumes that the regression curves are twice differentiable (not necessary C^2).

An epi-spline $s(\cdot)$ will approximate the regression function. We assume that the second derivative, although not necessarily a continuous function, is bounded. The epi-spline approximation is based on replacing this term by a , a piece-wise constant function with a constant value a_k , $k = 1, \dots, N$ on each of the N sub-intervals. Consequently, for time $t \in (t_{k-1}, t_k]$, one has,

$$s'(t) = v_0 + \int_0^t a(\eta) d\eta = v_0 + \delta \sum_{i=1}^{k-1} a_i + (t - t_{k-1})a_k$$

and

$$\begin{aligned} s(t) &= s_0 + \int_0^t s'(\eta) d\eta \\ &= s_0 + \sum_{i=1}^{k-1} \int_{t_{i-1}}^{t_i} s'(\eta) d\eta + \int_{t_{k-1}}^t s'(\eta) d\eta \\ &= s_0 + v_0 \cdot t + \delta \sum_{i=1}^{k-1} (t - t_i + \delta/2)a_i + \frac{1}{2}(t - t_{k-1})^2 a_k \end{aligned} \tag{1}$$

where s_0, v_0 and \vec{a} are constants to be determined to as to minimize the error with respect to the historic values l . Given a bound κ on the curvature of the regression function, this

leads to the following optimization problem equation:

$$\min ||(e_h^d, h \in \{1, \dots, T\}, d \in \mathcal{D})|| \quad (2)$$

$$\text{st. } e_h^d = l_h^d - s_h \cdot w_h^d, \quad (3)$$

$$s_h = s_0 + v_0 h + \delta \sum_{i=1}^{m-1} (h - i\delta + \delta/2) a_i + \frac{1}{2} (h - \delta(m-1))^2 a_m, \quad m = \lceil h/\delta \rceil \quad (4)$$

$$a_k \in [-\kappa, \kappa], \quad k = 1, \dots, N \quad (5)$$

Note that s_h is (1) evaluated at time h . In particular, when $h = k\delta$ for $k = 1, \dots, N$, the expression (4) is equivalent to:

$$s_h = s_0 + v_0 \cdot h + \delta^2 \sum_{i=1}^{\lceil \frac{h}{\delta} \rceil} (h - i + \frac{1}{2}) a_i.$$

Defining our *regression function* $r(\cdot)$ as

$$r_h(w^d) = s_h \cdot w_h^d, \quad (6)$$

the constraint (3) can be divided into two constraints:

$$\begin{aligned} e_h^d &= l_h^d - r_h^d, \\ r_h^d &= s_h \cdot w_h^d. \end{aligned} \quad (7)$$

If the norm chosen in (2) is the one or ∞ norm, then the resulting optimization problem is linear. The two norm results in a quadratic problem.

By solving this optimization problem we obtain s_0^*, v_0^*, \vec{a}^* and $s^*(\mathcal{D}) = s(s_0^*, v_0^*, \vec{a}^*)$, and with this we can define the function

$$\hat{r}(w; \mathcal{D}) = s^*(\mathcal{D}) \cdot w,$$

i.e., a vector-valued forecasting function created using the historic w and l vectors for the set of study periods \mathcal{D} and evaluated using predictor vector w . We refer to the resulting forecast for a sub-period $h \in 1, \dots, T$ of a particular study period, d , using

$$\hat{r}_h(w^d; \mathcal{D}) = s_h^*(\mathcal{D}) \cdot w_h^d.$$

Example 2.1. In our example, for each day $d \in \mathcal{D}$ the hourly load (l_h^d) and the hourly temperature prediction for day d on day $d-1$ (w_h^d) are available, so we use this data to obtain our forecast function by solving the previous optimization problem. One variant of this is described in [5].

A particular variation of this is to include a baseline load. This assumes that the electricity's demand can be represented as the sum of two components:

- a non-weather component which is related to the normal consumption habits of the people,

- a weather component, which we model depending on the forecast.

So, for each sub-period h we estimate the baseline load b_h as the average consumption over the study period,

$$b_h = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} l_h^d, \quad h = 1, \dots, 24,$$

and to incorporate it to our optimization problem we replace the constraint (7) by:

$$e_h^d = l_h^d - b_h - r_h^d.$$

In this case our forecasting function is given by,

$$\hat{r}_h(w^d; \mathcal{D}) = b_h + s_h^*(\mathcal{D}) \cdot w_h^d, \quad \forall d \in \mathcal{D}, \forall h \in 1, \dots, T,$$

so our problem is a particular case of the second by considering $b_h = 0, \forall h \in T$.

Another extension is to include a second predictor, as could be the dew point or the wind speed forecasts. This can be accomplished by estimating a regression function for each one of the predictors considered, add them in the expression (6) and include curvature constraints in the optimization problem for each one of them. We have also experimented with two dimensional epi-splines; however, the results were not significantly better than a simple linear combination of one-dimensional splines. \square

For a particular historical study period, d , and sub-period, h , we define the forecast error for a forecast function computed using historical study periods \mathcal{D} :

$$\varepsilon_h^d(\mathcal{D}) = l_h^d - \hat{r}_h(w^d; \mathcal{D}).$$

The errors for a particular sub-period can be used to compute a density of the errors for that sub-period,

$$f_{\varepsilon,h}(x; \mathcal{D}),$$

using epi-splines as described by Royset and Wets [9]. Finally, we integrate it numerically to obtain the CDF,

$$\mathbb{F}_{\varepsilon,h}(x; \mathcal{D}).$$

3 Categories and Partitions

The goal is to divide the historic study periods, \mathcal{D} , into subsets so that the errors in each subset occupy the same quantile of the error distribution; we refer to these subsets as *error categories*. It may be necessary, or desirable, to divide the times $t \in 1, \dots, T$ into contiguous partitions so that this is sensible. I.e., categories are formed for each cell of the partition.

The set H defines a partition of the study period, i.e.,

$$H = \{H_i\}_{i=1}^{|H|}, H_1 = 1, H_{|H|} = T, H_i < H_{i+1}.$$

In addition to this, we can define the parts of the study period (\mathcal{I}) by considering subsets of the times, where the i -th part of the study period is given by the set $[H_i, H_{i+1})$.

We consider K error categories of the error distribution; to ease exposition, they are assumed to be equally-sized slices of the distribution, but this is not required. To do this, let $\mathcal{K} = \{k_0, \dots, k_K\} = \{0, 1/K, 2/K, \dots, (K-1)/K, 1\}$ be the set of cutting points. Then, using the CDF for all of \mathcal{D} , $\mathbb{F}_{\varepsilon, h}(x; \mathcal{D})$, we can compute the inverse distribution function at each point $k \in \mathcal{K}$.

We categorize the study periods according to the rule: for $j \in \mathcal{J} = \{0, \dots, K-1\}$ and $h \in H$,

$$d \in \mathcal{D}_h^j \Leftrightarrow l_h^d - \hat{r}_h(w^d; \mathcal{D}) \in [\mathbb{F}_{\varepsilon, h}^{-1}(j/K; \mathcal{D}), \mathbb{F}_{\varepsilon, h}^{-1}((j+1)/K; \mathcal{D})],$$

where each category has a probability equal to $1/K$.

Note that each part of the study period has the same number of categories, i.e., \mathcal{J} is independent of the sub-period $h \in H$. Also, for each $h \in H$,

$$\bigcup_{j \in \mathcal{J}} \mathcal{D}_h^j = \mathcal{D}.$$

Having partitioned the study periods we can estimate, for each error category $j \in \mathcal{J}$ and each sub-period $h \in [H_i, H_{i+1})$, $i \in \mathcal{I}$, our predictor function,

$$\hat{r}(w; \mathcal{D}_{H_i}^j).$$

Then, for each $h \in H$ and each error category $j \in \mathcal{J}$, we calculate the errors related to a conditional regression curve and the CDF of the conditional errors,

$$\mathbb{F}_{\varepsilon, h, j}^{-1}(x; \mathcal{D}_h^j).$$

For a given period $d \in \mathcal{D}$, sub-period $h \in \{1, \dots, T\}$ and category $j \in \mathcal{J}$ we have a set of values,

$$\hat{r}_h(w^d; \mathcal{D}_{H_1}^j), \dots, \hat{r}_h(w^d; \mathcal{D}_{H_{|H|}}^j).$$

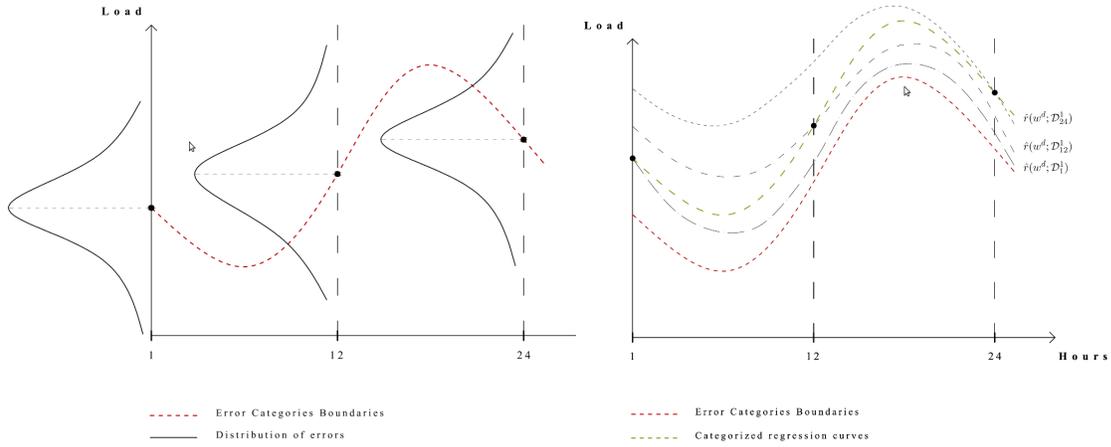
Each cell of the partition has a different regression and hence, for all middle partition boundaries, there could be two points (one from the regression for the cell to the left of the boundary and one from the right). We avoid discontinuities deriving a single regression curve per category, \hat{r}^j , by merging the ones obtained at the limit hours. In particular, for each boundary, H_i for $i \in \{1, \dots, |H|\}$ we have $|H|$ values. So, in order to avoid discontinuities and obtain a single regression curve per error category, $\hat{r}^j(w)$, we consider a weighted combination of the regression curves at the limit hours. Formally, for $h \in [H_i, H_{i+1})$,

$$\hat{r}_h^j(w^d) = \left(\frac{H_{i+1} - h}{H_{i+1} - H_i}\right) \cdot \hat{r}_h(w^d; \mathcal{D}_{H_i}^j) + \left(\frac{h - H_i}{H_{i+1} - H_i}\right) \cdot \hat{r}_h(w^d; \mathcal{D}_{H_{i+1}}^j).$$

Example 3.1. Figure 1 shows an example of the process described above considering:

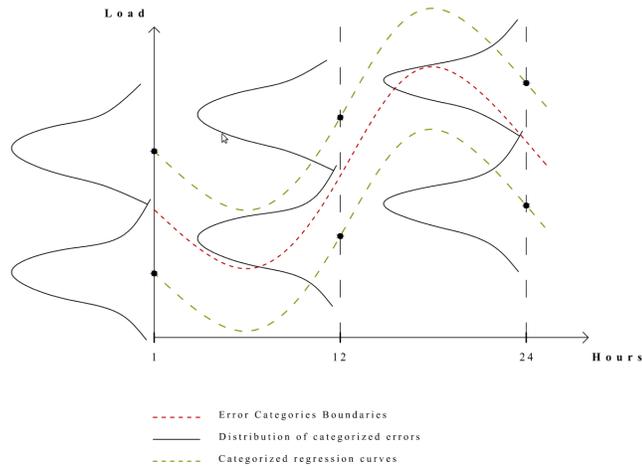
- $H = \{1, 12, 24\}$, which defines two parts of the study period, $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2\} = \{(1, 12), (12, 24)\}$. This allow us to model separately the errors in the morning and afternoon, which are certainly different,
- $\mathcal{K} = \{0, 0.5, 1\}$, delimiting two error categories, ie, $\mathcal{J} = \{0, 1\}$. Using this we are able to cover days where the demand was above and others where it was below the average.

Figure 1a depicts how the error categories are built. The black lines represent the distribution of the errors, whereas the black dots symbolize $\mathbb{F}_{\varepsilon, h}^{-1}(0.5; \mathcal{D})$ for each $h \in H$. Figure 1b presents how the regression curve for the category 1 is obtained based on the forecasting functions estimated for this category and for each day part, $\mathcal{D}_{H_i}^1$, with $i \in \{0, 1\}$. Finally, Figure 1c gives a representation of the error distributions conditioned on the error category and sub-period. In this figure each black curve represents the distribution of the errors for a given category $j \in \mathcal{J}$ and cell of the study period partition $H_i \in H$, $f_{\varepsilon, H_i, j}(x; \mathcal{D}_{H_i}^j)$, and the points depict their expected value.



(a) Definition of the error categories

(b) Regression curves computation



(c) Estimation of error distributions

Figure 1: Construction of the categories and the error distributions

□

4 Scenario construction

The preceding sections describe methods of obtaining an error distribution $f_{\varepsilon, H_i, j}(x; \mathcal{D}_{H_i}^j)$ for each error category $j \in J$ and sub-period $H_i \in H$ and a set of forecasting functions $\hat{r}^j(w)$, one for each category $j \in \mathcal{J}$. The basic idea to generate the scenarios is to discretize the error distributions and then use these points, known as *skeleton points*, as a structure for the full scenario process.

4.1 Skeleton points computation

Given w^d for a future study period, d , the forecasting function and error distributions can be used to construct discretized outcomes with associated probabilities at the points in time given by H . Conditional on being in an error category, $j \in \mathcal{J}$ at some particular H_i , we use this somewhat simplified notation

$$\mathbb{F}_{\tilde{l}_{H_i}^j}(x) = \mathbb{F}_{\varepsilon, H_i, j} \left(x - \hat{r}_{H_i}^j(w^d) \right)$$

to represent the CDF of $\tilde{l}_{H_i}^j$, which is the random variable for the future value of l_h . Let $f_{\tilde{l}_{H_i}^j}(x)$ be the corresponding PDF. Given a set of cutting points of increasing value in $[0, 1]$, $c = \{0, c_2, c_3, \dots, 1\}$, we form this set for the values of \tilde{l} :

$$\begin{aligned} C_{\tilde{l}_{h,d}^j} &= \left\{ \mathbb{F}_{\tilde{l}_{h,d}^j}^{-1}(0), \mathbb{F}_{\tilde{l}_{h,d}^j}^{-1}(c_2), \mathbb{F}_{\tilde{l}_{h,d}^j}^{-1}(c_3), \dots, \mathbb{F}_{\tilde{l}_{h,d}^j}^{-1}(1) \right\} \\ &= \left\{ 0, C_{\tilde{l}_{h,d}^j}^2, C_{\tilde{l}_{h,d}^j}^3, \dots, \mathbb{F}_{\tilde{l}_{h,d}^j}^{-1}(1) \right\}. \end{aligned}$$

To reduce the notation for the moment, we re-index and rename the set to be simply $C = \{C_1, \dots, C_{|C|}\}$. Note that with this indexing $C_1 = 0$ and $C_{|C|} = \mathbb{F}_{\tilde{l}_{h,d}^j}^{-1}(1)$.

For the $j \in \mathcal{J}$ and H_i of interest, we now form intervals that exhaust all possibilities for \tilde{l} and compute their probabilities:

$$\Pr \left(\tilde{l} \in [C_k, C_{k+1}] \right) = \mathbb{F}_{\tilde{l}}(C_{k+1}) - \mathbb{F}_{\tilde{l}}(C_k),$$

which we write later as $p_{H_i, k}^{d, j}$. In order to summarize the values on each interval with a discrete point we compute, for each one of them, the conditional expectation,

$$\mathbb{E} \left[\tilde{l} \mid \tilde{l} \in [C_k, C_{k+1}] \right] = \frac{\int_{C_k}^{C_{k+1}} x \cdot f_{\tilde{l}}(x) dx}{\int_{C_k}^{C_{k+1}} f_{\tilde{l}}(x) dx}.$$

As a result of repeating this process for the predictor w^d we obtain a set of *scenario skeleton points*, which we write as $\tilde{l}_{H_i, k}^{d, j}$ for $k = 1, \dots, |C| - 1$, for all $j \in \mathcal{J}$ and all $i \in \mathcal{I}$ each with an associated probability, $p_{H_i, k}^{d, j}$.

Example 4.1. Figure 2 shows the scenario skeleton points generated for our example. In this case, the set of points in $[0, 1]$ selected were $\{0, 0.5, 1\}$, and this implies that the number of scenarios generated for each category $j \in \{0, 1\}$ and hour $h \in H \setminus \{1\}$ is 2, each one with probability 0.5.

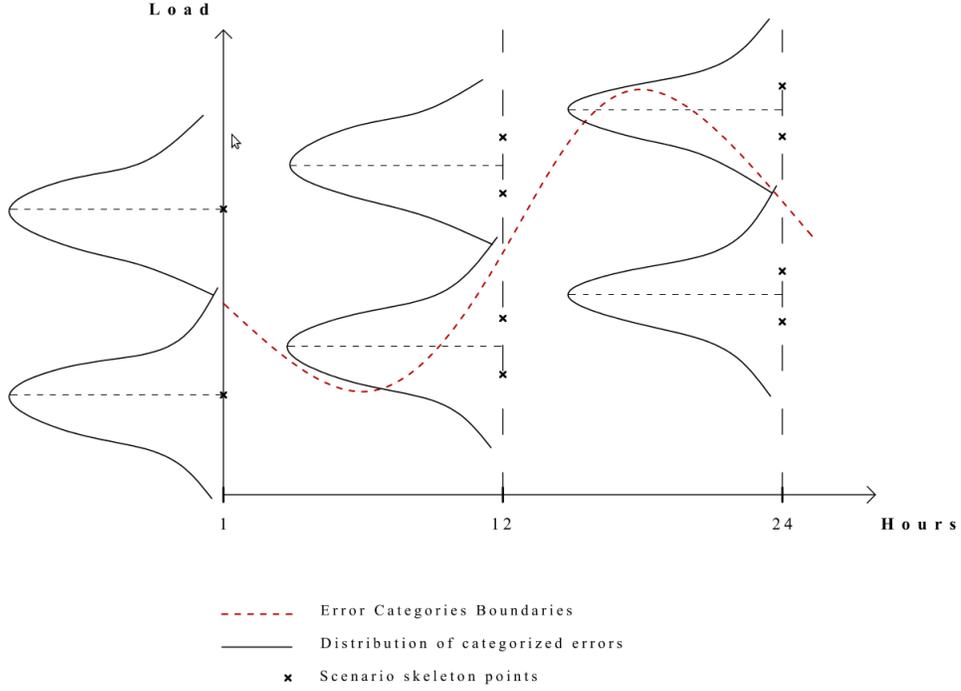


Figure 2: Computation of scenario points

□

4.2 Scenario paths generation

From the previous step we obtain, for each $d \in \mathcal{D}$ and $j \in \mathcal{J}$, a set of scenario skeleton points

$$\bar{l}_{H_1,1}^{d,j}, \left\{ \bar{l}_{H_2,k}^{d,j} \right\}_{k=1}^{|C|-1}, \dots, \left\{ \bar{l}_{H_{|H|},k}^{d,j} \right\}_{k=1}^{|C|-1},$$

and their corresponding probabilities. The last thing is to connect them in order to generate a stochastic process path for a full study period, which is a scenario.

Noting that a scenario path can be obtained by picking a single scenario skeleton point at each $h \in H$ and connecting them, the main idea to build our scenarios is to cover all the possible combinations of scenario skeleton points at different sub-periods $h \in H$, for each error category $j \in \mathcal{J}$.

We refer to a *scenario skeleton* as the sequence of scenario skeleton points that support a given scenario. Noting that at each sub-period H_i each scenario skeleton point $\bar{l}_{H_i,k}^{d,j}$ can be described by a pair of indexes (j, k) , where $j \in \mathcal{J}$ and $k \in \{1, \dots, |C| - 1\}$, a scenario skeleton can be defined by choosing a pair (j, k) at each sub-period $H_i \in H$. Then, a given

scenario skeleton a can be represented as

$$a = (a_1, \dots, a_{|H|}) = \{(j_1, k_1), \dots, (j_{|H|}, k_{|H|})\},$$

where $a_h = (j_h, k_h)$ is the pair chosen for the sub-period $h \in H$ to build the scenario supported by the skeleton a .

Algorithm 1 provides a method to generate the set of scenario skeletons (A) that supports all the scenarios we want to generate. In line 3 we initialize a new scenario skeleton a which starts in the error category j . In the next line we apply the routine SKELETONS to the initialized skeleton a .

Algorithm 1 Scenario skeletons generation

INPUT: $H = \{H_1, \dots, H_{|H|}\}$
 $C =$ cutting points
 $\mathbb{F}_\varepsilon = \{\mathbb{F}_{\varepsilon, h}(x; \mathcal{D})\}_{h \in H}$
 $\bar{l} = \{\bar{l}_{h, k}^{d, j}\}_{j \in \mathcal{J}, h \in H, k \in \{1, \dots, |C|-1\}}$
 $\hat{r} = \{\hat{r}_h^j(w^d)\}_{j \in \mathcal{J}, h \in H}$

- 1: $A \leftarrow \{\emptyset\}$
- 2: **for** $j \in \mathcal{J}$ **do**
- 3: $a \leftarrow ((j, 1))$
- 4: $A \leftarrow \text{SKELETONS}(A, H, C, a, \bar{l}, \hat{r}, \mathbb{F}_\varepsilon)$
- 5: **return** A

The routine SKELETONS($A, H, C, a, \bar{l}, \hat{r}, \mathbb{F}_\varepsilon$) enables us to generate all the scenario skeletons starting at a partial skeleton a and save them in the set of skeletons A . In this context, a partial skeleton a is a sequence of pairs $((j_1, k_1), \dots, (j_n, k_n))$ where $n < |H|$, i.e. it defines a scenario path only up to the sub-period $H_n \in H$.

Algorithm 2 presents how this routine works. A given skeleton a is completed when it contains a pair of coordinates (j, k) for each sub-period $h \in H$. Then, as shown in line 3 it is stored in the set of skeletons A . If the skeleton a is not ready, to add the next coordinate we first check to which error category the last entry of the skeleton a belongs. To accomplish this we take the last element of a (lines 5 and 6); compute the error associated with the corresponding skeleton point and obtain the value of the CDF for that error (line 7); and find the error category with the function FEC, which returns the value j' such that $x \in \left[\frac{j'}{K}, \frac{j'+1}{K}\right]$.

Then, having found j' we append to a the skeleton points belonging to that error category in the next sub-period (line 10) and we apply the routine SKELETONS. As a result of applying this routine recursively we obtain the set A , which contains all the scenario skeletons that will generate the scenarios for the full process.

Algorithm 2 Skeletons starting from a partial skeleton a

```

1: procedure SKELETONS( $A, H, C, a, l, r, \mathbb{F}_\varepsilon$ )
2:   if  $|a| = |H|$  then
3:      $A \leftarrow A \cup \{a\}$ 
4:   else
5:      $i \leftarrow |a|$ 
6:      $(j, k) \leftarrow a_i$   $\triangleright a_i$  is the last element of  $a$ 
7:      $x \leftarrow \mathbb{F}_{\varepsilon, H_i} \left( \bar{l}_{H_i, k}^{d, j} - \hat{r}_{H_i}(w^d; \mathcal{D}) \right)$ 
8:      $j' \leftarrow \text{FEC}(x)$ 
9:     for  $k := 1$  to  $|C| - 1$  do
10:       $a \leftarrow a \cup ((j', k))$ 
11:       $a, A \leftarrow \text{SKELETONS}(A, H, C, a)$ 
12:       $a \leftarrow a \setminus ((j', k))$ 
13:   return  $a, A$ 
14: end procedure

```

It's important to remark that a scenario skeleton a starting in $j \in J$, i.e. $a_1 = (j, 1)$, doesn't necessarily belong to the same error category over all the sub-periods. So updating the error category with the function FEC permits the possibility of scenario paths crossing.

Having the set A we just have to connect the scenario skeleton points belonging to each skeleton $a \in A$ in order to generate our scenario paths. A full description of our approach to do it is formally outlined as Algorithm 3. This algorithm loops over all the scenario skeletons in the set A , generating a total of $|A|$ scenario paths.

For a particular scenario skeleton $\vec{a} = ((j_1, k_1), \dots, (j_{|H|}, k_{|H|}))$, the first step is to initialize the probability of the scenario (s) under construction p_s^d (line 4). Then, for each sub-period $H_i \in H$ the distance (ε_i) between the skeleton point $\bar{l}_{H_i}^{d, j_i}$ and the regression curve of the corresponding error category $\hat{r}_{H_i}^{j'}(w^d)$ is computed, as shown in line 8. To find the error category related to each skeleton point we used the same approach as the one described above (lines 6 and 7).

To compute the load for a given sub-period $h \in [H_i, H_{i+1}]$ we consider the regression curve of the corresponding error category and the weighted sum of the deviations at the limit sub-periods (ε_i and ε_{i+1}), as shown in line 11. Finally, in line 12 the probability of the scenario s is updated.

Algorithm 3 Scenario paths computation

INPUT: $\bar{l}_{H_1,1}^{d,j}$ and $p_{H_1}^{d,j}$, $\forall j \in \mathcal{J}$,
 $\bar{l}_{H_i,k}^{d,j}$ and $p_{H_i,k}^{d,j}$, $\forall j \in \mathcal{J}, i \in \{2, \dots, |H|\}, k \in \{1, \dots, |C| - 1\}$,
 $\hat{r}_h^j(w^d)$, $\forall j \in \mathcal{J}, h \in \{1, \dots, T\}$,
 $A =$ set of scenario skeletons obtained applying Algorithm 1.

- 1: $s \leftarrow 1$
- 2: $S = \{\emptyset\}$
- 3: **for** $\bar{a} \in A$ **do** $\triangleright \bar{a} = ((j_1, k_1), \dots, (j_{|H|}, k_{|H|}))$
- 4: $p_s^d \leftarrow |\mathcal{J}|^{-1}$
- 5: **for** $i := 1$ **to** $|H|$ **do**
- 6: $x \leftarrow \mathbb{F}_{\varepsilon, H_i} \left(\bar{l}_{H_i,k}^{d,j_i} - \hat{r}_{H_i}(w^d; \mathcal{D}); \mathcal{D} \right)$
- 7: $j' \leftarrow \text{FEC}(x)$
- 8: $\varepsilon_i \leftarrow \bar{l}_{H_i}^{d,j_i} - \hat{r}_{H_i}^{j'}(w^d)$
- 9: **for** $i := 1$ **to** $|H| - 1$ **do**
- 10: **for** $h := H_i$ **to** H_{i+1} **do**
- 11: $\hat{l}_{h,s}^d \leftarrow \hat{r}_h^{j_{i+1}}(w^d) + \left(\frac{H_{i+1}-h}{H_{i+1}-H_i} \right) \cdot \varepsilon_i + \left(\frac{h-H_i}{H_{i+1}-H_i} \right) \cdot \varepsilon_{i+1}$
- 12: $p_s^d \leftarrow p_s^d \cdot p_{H_{i+1},k_{i+1}}^{d,j_{i+1}}$
- 13: $S \leftarrow S \cup \{s\}$
- 14: $s \leftarrow s + 1$
- 15: **return** $\hat{l}_{h,s}^d$ and p_s^d , $h \in \{1, \dots, T\}, s \in S$

Note that the expression in line 11 ensures a smooth transition between scenario skeleton points in consecutive sub-periods $\{H_i, H_{i+1}\}, i = 1, \dots, |H| - 1$. Moreover, the expression in line 12 shows that the probability related to a scenario s can be computed as the product of the scenario skeleton point's probabilities that support it.

Example 4.2. Figure 3 shows the scenario path construction for our first category.

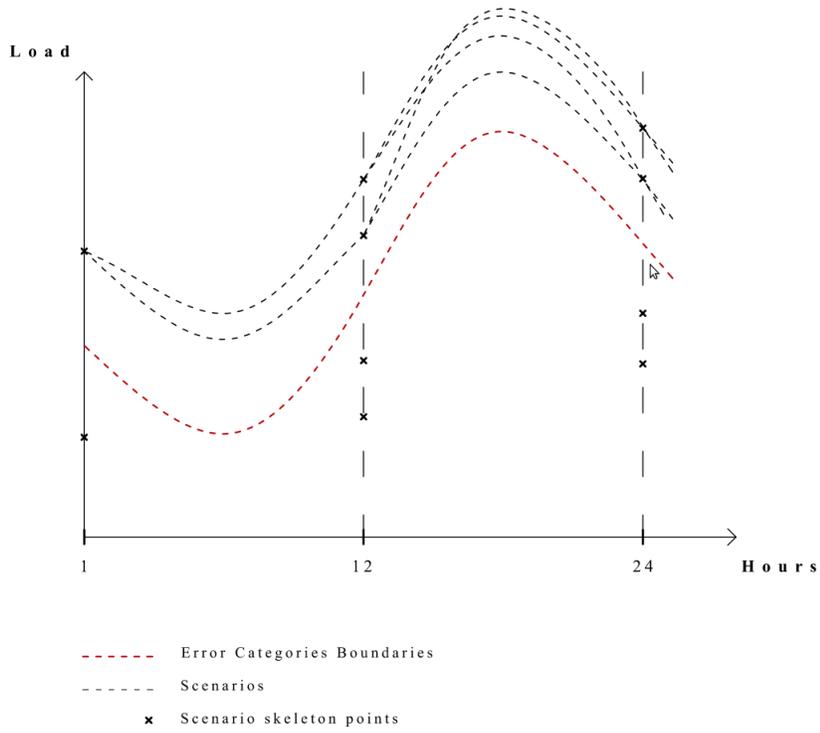


Figure 3: Scenario path construction

Since we consider two categories and $\{0, 0.5, 1\}$ are the cutting points, the probability related to each scenario is given by:

$$p = \frac{1}{2} \cdot 0.5 \cdot 0.5 = 0.125$$

□

Following this methodology we build a number of paths equal to

$$|\mathcal{J}| \cdot (|C| - 1)^{|H|-1}$$

where $|\mathcal{J}|$ is the number of categories considered, $|C|$ is the number of cutting points and $|H|$ is the number of cells of the partition.

Example 4.3. Figure 4 shows the 8 scenarios generated by our approach for our previous example.

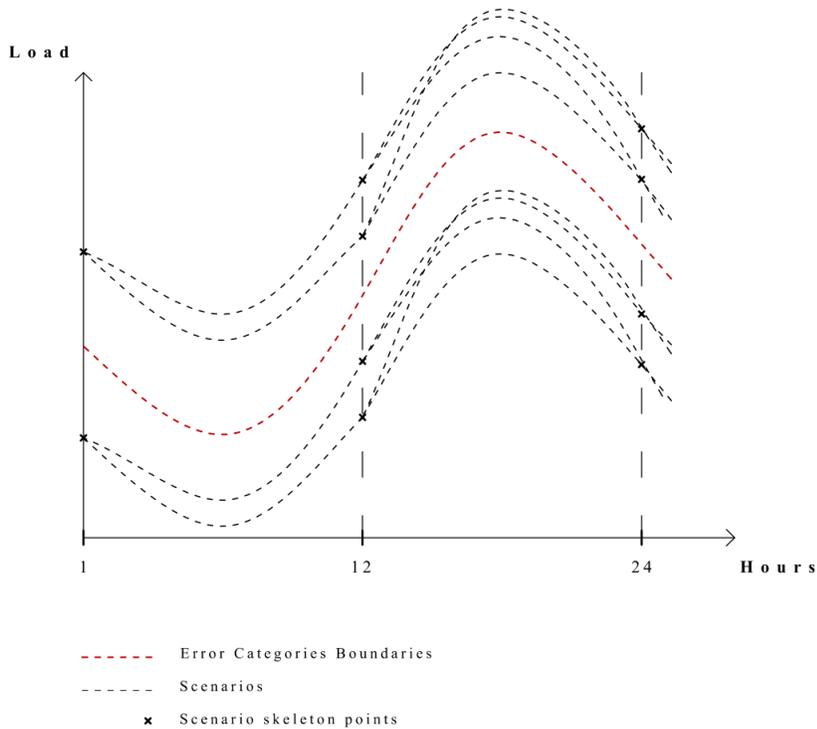


Figure 4: Full set of scenarios

It's important to remark that in this example there are no crossings between the scenarios across the error categories because it's a small example just to illustrate our methodology to generate the scenarios so there are very few cutting points. \square

5 Benchmark results

In this section we report on some computational experiments to benchmark the scenario generation method just described. To provide a concrete problem domain, we continue with the electricity demand application. No work has been done on generating scenarios for 24-hour random variable based on day ahead information so directly comparing our results with the methods used in the literature is not possible. However, we adapted the methodology described in [3] to our problem and we used this to generate scenarios and compare them with the ones obtained with our approach. This allows us to benchmark the last steps in the process (the first steps are used by both methods).

In the remainder of this section we give some background and then explain how we adapted the *forward construction* method introduced by [3]. Then, we present the approach we used to compare both scenario generation methods, and finally we show the results of this comparison.

5.1 Alternative scenario generation methods

A very common approach to generate scenarios from historical data is the use of clustering methods ([2], [8], [15]). In this case, recalling that each of the observations available for l is a vector of length T , clusters can be generated to summarize all the historical data in a set of representative centroids, which can be considered as equally likely scenarios.

Another accepted approach are Monte Carlo sampling methods ([12]). The problem to implement it in our case is that with limited data one cannot estimate a distribution for a T -dimensional problem (e.g., when $T = 24$), so one cannot sample from any distribution. However, for each sub-period $h \in T$ one can obtain a distribution of the observations, and from these distributions samples can be generated. Finally, by connecting the samples generated at each sub-period h a full process for a scenario can be generated.

In summary, a Monte Carlo approach can be summarized as follows:

1. Generate the distribution of l_h for each $h \in T$, $\mathbb{F}_h(l)$,
2. Generate a set of random numbers $\xi = \{\xi_1, \xi_2, \dots\}$,
3. For each random number ξ_i generate a scenario path by considering,

$$l_h^i = \mathbb{F}_h^{-1}(\xi_i), \quad \forall h \in T.$$

It's important to remark that the scenario paths generated by these steps are perfectly correlated, which might not be desirable.

Example 5.1. Considering the same data used in our previous example we first generate the hourly distribution of the loads. Figure 5 shows the functions we obtained.

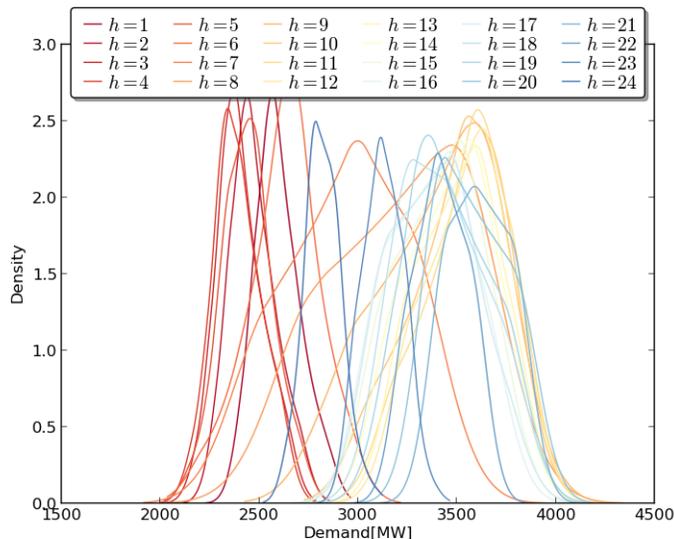


Figure 5: Hourly distributions

Then, considering these distribution and a given a list of random numbers in $[0, 1]$ ($|\xi| = 100$) we generate paths, which are shown in the Figure 6.

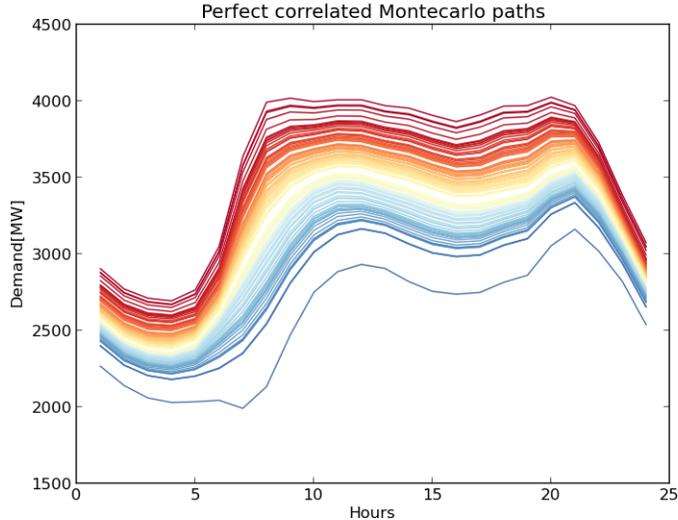


Figure 6: Monte Carlo paths

□

The two methods just described use observations as an input in order to generate the scenarios. However, we would like to take into account our day ahead predictors w in our scenario generation method. Then, an alternative is to generate samples from the distribution of the errors for each sub-period $h = 1, \dots, T$, and add them to the expected load curve generated using our epi-splines approach to generate samples of the full process (of dimension T). Finally, having our set of samples $\xi = \{\xi_1, \dots, \xi_N\}$ we apply a scenario reduction method to build our scenarios starting from these samples. In our case, we implemented the methodology described in [3] to accomplish this.

5.1.1 Sample generation

As in the model we propose, we implement an extension of the Eichhorn et al. methodology to reduce scenarios by relying on the forecasting function $\hat{r}(w; \mathcal{D})$, generating samples of the hourly errors and adding them to the function in order to generate samples of the process. By using this approach one can generate as many scenarios as desired, and also we take into account the values of the predictors w when generating the samples.

The first step is to generate the distribution of the errors, which can be accomplish by using the exponential epi-spline technology introduced by [10]. The next step is to generate samples from these distributions. To do this we proceed as described in Algorithm 4. The main idea is to generate random uniform numbers at each time $h \in T$ (line 4) and obtain

a sample of the errors from their inverse cumulative distribution function, $\mathbb{F}_{\varepsilon,h}^{-1}(\omega; \mathcal{D})$, as shown in line 6. To capture correlation between consecutive periods we consider as random number a linear combination of the previous (ω_1) and the actual (ω_2) random number considered (line 5). Finally, line 7 shows how the sample for the period h , $l_h^{d,s}$, is computed as the sum between the main regression curve $\hat{r}_h^d(w; \mathcal{D})$ and the error sample ε_h .

Algorithm 4 Samples generation

INPUT: N = number of samples to generate,
 ρ = measure of the correlation between the errors of consecutive hours,
 $\mathbb{F} = \{\mathbb{F}_{\varepsilon,h}(x; \mathcal{D})\}_{h \in \{1, \dots, T\}}$,
 $\hat{r} = \{\hat{r}_h^d(w; \mathcal{D})\}_{h \in \{1, \dots, T\}}$.

- 1: **for** $s := 1$ **to** N **do**
- 2: $\omega_1 \leftarrow \mathcal{U}[0, 1]$
- 3: **for** $h := 1$ **to** T **do**
- 4: $\omega_2 \leftarrow \mathcal{U}[0, 1]$
- 5: $\omega \leftarrow \rho \cdot \omega_1 + (1 - \rho) \cdot \omega_2$
- 6: $\varepsilon_h \leftarrow \mathbb{F}_{\varepsilon,h}^{-1}(\omega; \mathcal{D})$
- 7: $l_h^{d,s} \leftarrow \hat{r}_h^d(w; \mathcal{D}) + \varepsilon_h$
- 8: $\omega_1 \leftarrow \omega$
- 9: **return** $l_h^{d,s}, s = 1, \dots, N, h \in \{1, \dots, T\}$

Note that by using $\rho = 1$ we obtain perfectly correlated samples.

Example 5.2. Figure 7 shows an example of the probability density functions obtained for Connecticut, USA in the Autumn of 2011.

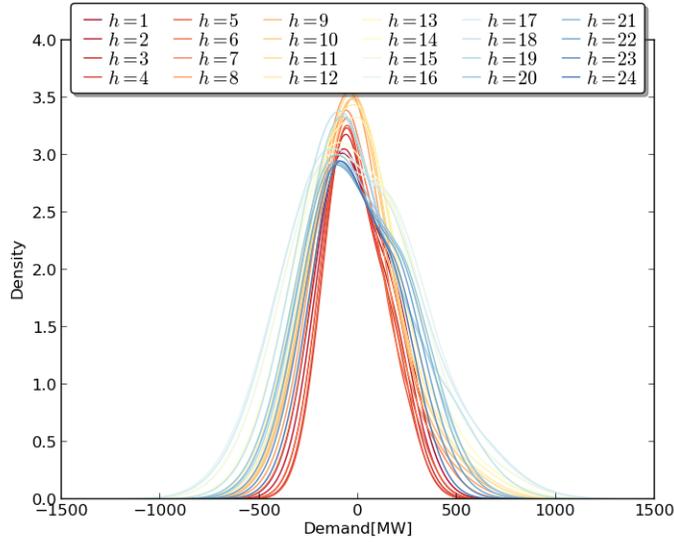


Figure 7: Error distributions at different hours

Using these distributions we generate samples for each hour. Then, connecting these hourly samples we generate paths for the full process. Figure 8 presents an example of $N = 500$ perfectly correlated ($\rho = 1$) samples using the distributions depicted in Figure 7.

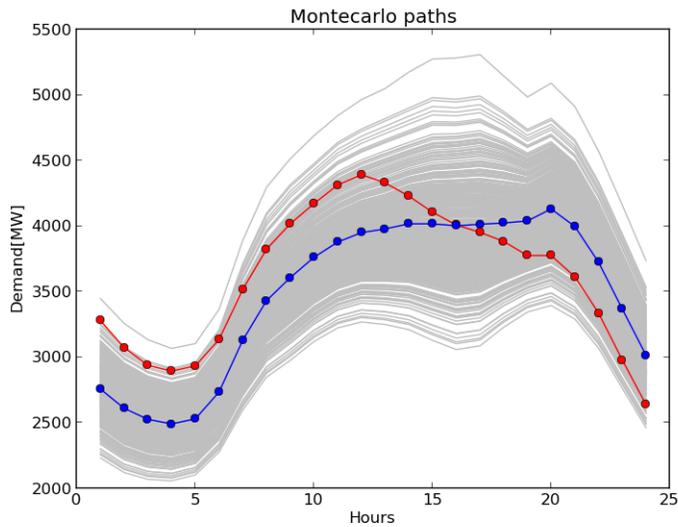


Figure 8: Monte Carlo paths generated for September 15th 2011, considering $\rho = 1$. The blue line depicts the expected loads; the red line shows the actual loads; the gray lines represent the samples generated.

□

5.1.2 Forward scenario construction

Based on N samples of the full process we can build a scenario tree using the scenario reduction methodology described in [3]. In particular, we implement a variation of the *forward construction* in which we allow the construction of clusters only in a subset of sub-periods, $H \subset \{1, \dots, T\}$. This is because the number of scenarios grows exponentially with the number of periods T .

The *forward selection* method described in Algorithm 5 enables summarizing a set of observations C in k clusters chosen such that the weighted distance from their centroids to all the observations is minimized. Then, the C points are classified in the k clusters, assigning each observation to the cluster whose centroid is the closest to that point.

Algorithm 5 Forward Selection

- 1: **procedure** FORWARD SELECTION(C, k)
- 2: $J \leftarrow C$
- 3: $I \leftarrow \{\emptyset\}$
- 4: **while** $|I| \neq k$ **do**
- 5: Find g such that

$$g \leftarrow \operatorname{argmin}_{u \in J} \sum_{j \in J \setminus \{u\}} p_j \min_{i \in J \setminus \{u\}} |l_t^j - l_t^i|$$

- 6: $J \leftarrow J \setminus \{g\}$
 - 7: $I \leftarrow I \cup \{g\}$
 - 8: **return** I
 - 9: **end procedure**
-

The *forward construction* approach is based on building clusters recursively at each period $h \in H$ with the forward selection algorithm. This is described by the Algorithm 6.

Algorithm 6 Forward Construction

INPUT: l = observations available,
 k = number of clusters per sub-period considered,
 H = sub-periods on which we generate the clusters,

- 1: **for** $h \in H$ **do**
- 2: $C_h \leftarrow \{\emptyset\}$
- 3: $\gamma \leftarrow$ FORWARD SELECTION(l, k, H_1)
- 4: $C_1 \leftarrow C_1 \cup \{\gamma\}$
- 5: **for** $h \in H \setminus \{H_1\}$ **do**
- 6: **for** $c \in C_h$ **do**
- 7: $\gamma \leftarrow$ FORWARD SELECTION(c, k, h)
- 8: $C_h \leftarrow C_h \cup \{\gamma\}$
- 9: **return** $l_h^{d,s}, s = 1, \dots, N, h \in \{1, \dots, T\}$

As a result of applying the forward construction algorithm we obtain, for each $h \in H$, a set of clusters C_h with their corresponding centroids. Then, each observation $l^{d,s}$ can be represented with a set of centroids

$$C(l^{d,s}) = \left\{ C_1 \left(l_1^{d,s} \right), \dots, C_T \left(l_T^{d,s} \right) \right\},$$

where $C_t(\cdot)$ is a function that return the centroid of the cluster to which the observation $l_t^{d,s}$ belongs.

In order to connect the centroids that represent each observation and obtain a scenario for the full process we compute a weighted sum of the values related to each centroid. For example, given $h \in [H_i, H_i + 1]$, the value of the scenario corresponding to the observation $l^{d,s}$ is

$$\hat{l}_h^d = \left(\frac{H_{i+1} - h}{H_{i+1} - H_i} \right) \cdot l_h^{d,s_i} + \left(\frac{h - H_i}{H_{i+1} - H_i} \right) \cdot l_h^{d,s_{i+1}}$$

where s_i and s_{i+1} are the indexes of the centroids $C_{H_i} \left(l_i^{d,s} \right)$ and $C_{H_{i+1}} \left(l_{i+1}^{d,s} \right)$ respectively. Finally, the probability related to each scenario is equal to the sum of the probabilities of the observations belonging to that cluster and the last period, i.e.,

$$q_k = \sum_{i \in C_T^k} p_i$$

for each $k = 1, \dots, k_T$.

Example 5.3. The following figures show examples of the scenarios for September 15th 2011 in Connecticut generated by the version of forward construction described here. We also considered as parameters $k = 2$ and $T = [1, 8, 16, 24]$. The difference between these plots is the data used and the correlation coefficient ρ .

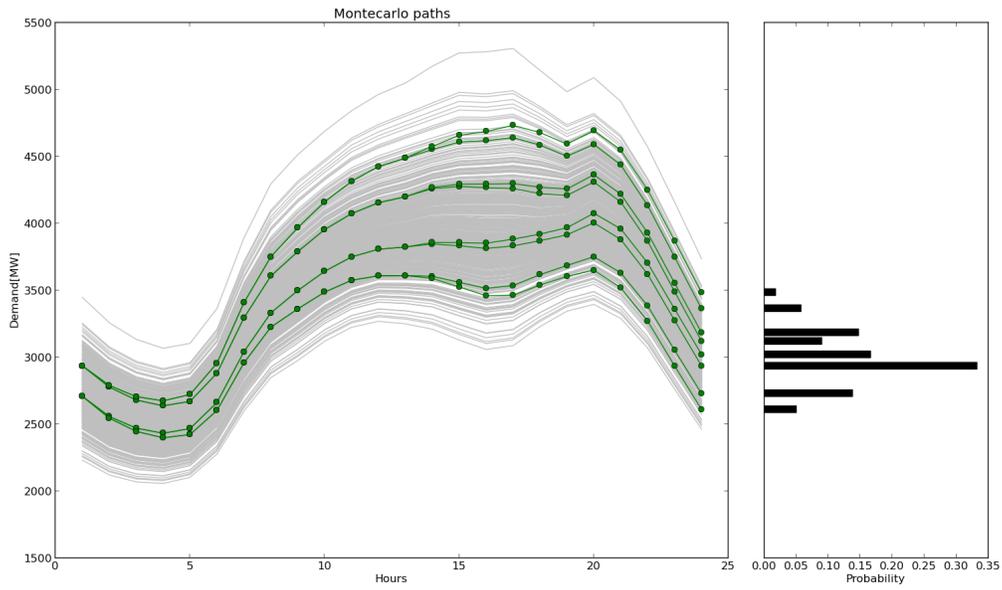


Figure 9: Scenarios generated by a version of forward construction with $\rho = 1$

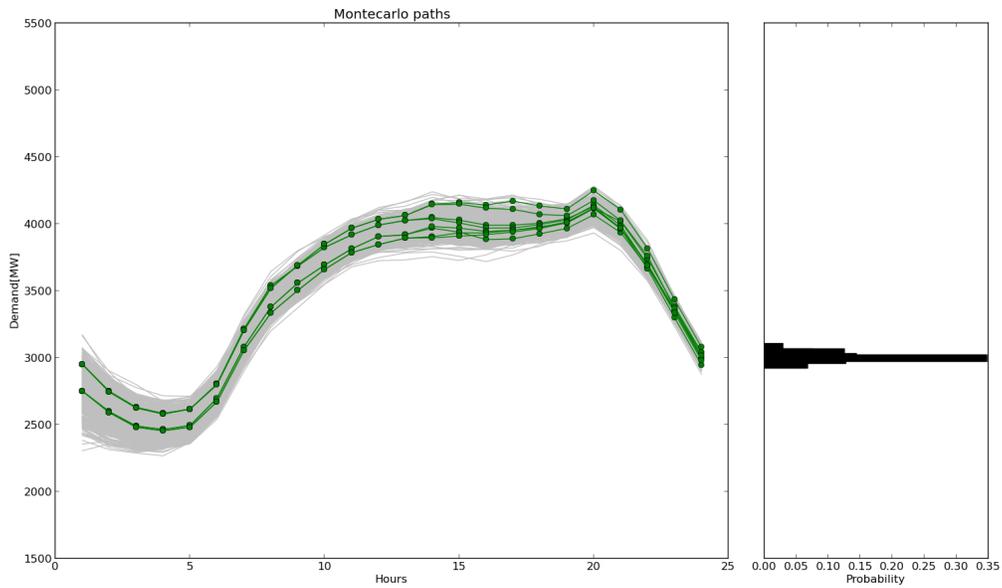


Figure 10: Scenarios generated by a version of forward construction with $\rho = 0.9$

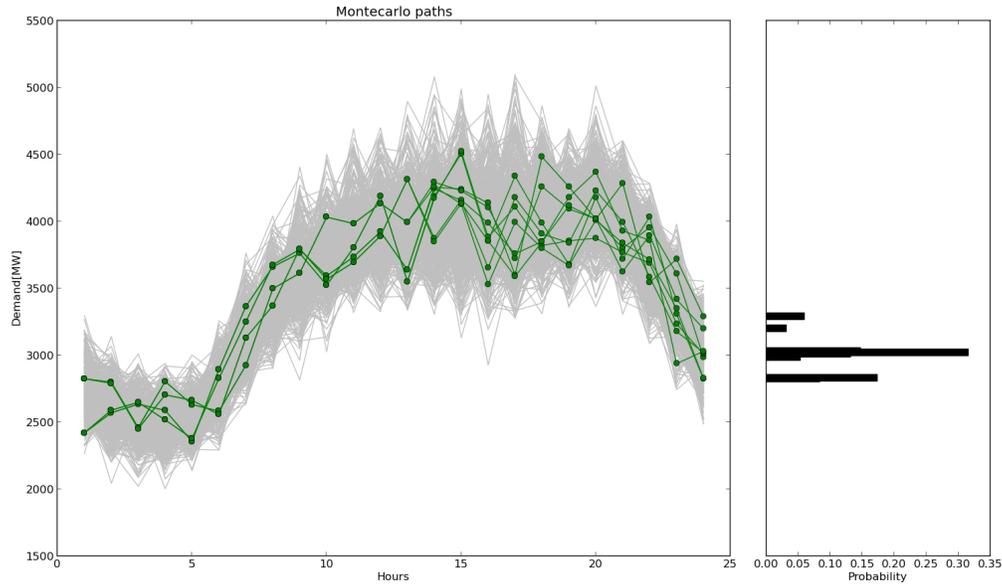


Figure 11: Scenarios generated by a version of forward construction with $\rho = 0$

5.2 Results

As noted in [6], the ultimate goal of incorporating stochasticity in optimization problems is to make better decisions. So a straightforward way to measure the quality of different scenario generation methods is to solve the underlying optimization problem and see which scenario tree leads to better decisions. The optimization problem we use for the benchmark is the stochastic unit commitment (UC) problem (see [14]).

We use the two-stage formulation of Carrion and Arroyo [1], extended slightly as presented in [11]. Our approach to compare the quality of different scenario trees can be summarized as follows: given two scenario trees, for each one of them,

1. Solve the extensive form of the optimization problem using scenarios,
2. Solve the optimization problem considering the real observations and fixing the first stage variables to be the ones obtained in the step 1.

For these experiments we considered the historical weather forecasts and electricity demands of the autumns of 2009 and 2010 in Connecticut, USA, to estimate scenarios for the Autumn of 2011. In addition to this, we generated a varying number of scenarios, $\{4, 8, 16, 32\}$. For each number of scenarios and for each date, we generated two scenario trees: one considering the approach presented in this paper (*RWW*) and the parameters described in Table 1; and the other one using the Forward Selection method introduced by [3] and implemented as described in the previous subsections (*FS*) using 1000 samples.

Number of scenarios	\mathcal{K}	H	c
4	$\{0, 0.5, 1\}$	$\{1, 24\}$	$\{0, 0.5, 1\}$
8	$\{0, 0.5, 1\}$	$\{1, 12, 24\}$	$\{0, 0.5, 1\}$
16	$\{0, 0.5, 1\}$	$\{1, 10, 15, 24\}$	$\{0, 0.5, 1\}$
32	$\{0, 0.5, 1\}$	$\{1, 12, 24\}$	$\{0, 0.01, 0.5, 0.99, 1\}$

Table 1: Parameters involved in the scenario generation with the *RWW* method.

To compare the results obtained with the different scenarios we compute, for each day, the optimality gap ($G(RWW)$ and $G(FS)$) in relation to the actual loads (see [7]) of the SCUC problem considering the data used in [4].

In Table 2 we present the percentage of days on which the solution obtained with the *RWW* scenario generation method leads to a smaller, equal and bigger optimality gap than the one obtained with the *FS* scenario generation method. In particular we can see that, independent of the number of scenarios considered, the number of days on which the methodology presented in this paper results in a lower optimality gap is more than the number of days on which the *FS* method does.

	Number of scenarios			
	4	8	16	32
% of days where $G(RWW) < G(FS)$	20.89	25.37	23.88	29.85
% of days where $G(RWW) = G(FS)$	73.13	68.65	73.13	65.67
% of days where $G(RWW) > G(FS)$	5.97	5.97	2.98	4.47

Table 2: Percentage of days on which the optimality gap obtained using the *RWW* scenario generation method is smaller, equal and bigger than the one obtained with the *FS* scenario generation method.

Complementing the previous results, in Table 3 we present the average optimality gap obtained with each method. From this table we can see that for each number of scenarios the average optimality gap obtained with the *RWW* scenario generation method is smaller, which means that the first stage decisions taken with these scenarios are better. The percentage differences are not large, in part because the two methods share their first steps. In monetary terms, the differences would be significant even just for the US state of Connecticut. However, our main goal is to demonstrate that the methods given here are effective.

	Number of scenarios			
	4	8	16	32
Average optimality gap with <i>RWW</i>	22.84	22.76	23.27	22.71
Average optimality gap with <i>FS</i>	23.37	24.15	24.43	24.26

Table 3: Optimality gap obtained with each scenario generation method.

6 Conclusions

We have described a method for creating scenarios using leading predictors for multiple time steps when the historic data is limited. We propose a deterministic approach to generating scenarios and probability weights that relies on an approximation of the empirical distribution using splines. Our motivating example is the creation of scenarios for 24 hourly electricity demands with regressions and error estimates based on data from only hundreds of observations.

A key step is the use of a functional regression to obtain a forecast based on leading indicators. We then make use of epi-splines to fit an error distribution. We partition the study periods and categorize the errors with each cell of the partition. This enables fitting regression curves that are conditional on the error category and corresponding error distributions. In contrast to sampling based methods, when scenarios are needed we use these distributions to create the skeletons of scenarios in a systematic way.

Benchmark studies show that the method works well. We could not benchmark the entire process, but we are able to extend the forward generation method to allow some comparisons. The methods shared the technology for forecasting and error distribution estimation. The method of scenario generation given here performed better than forward generation for the benchmark study that we conducted.

One area for further research is the partition of the time steps. In the electricity demand example, the study periods had fairly natural partition breakpoints at various hours of the day but for some applications, the categories might be less obvious. It may be desirable to form the partitions so as to be the biggest contiguous groups with fewer than some number of non-conforming errors. Note that for multi-stage formulations, the time steps are natural boundaries for creating multi-stage scenario trees.

Error categories could also be analyzed in greater depth. In our work each part of the study period has the same number of categories but we could have more or fewer for some cells of the partition of the study period by $i \in \mathcal{I}$. For example, one could imagine that in the electricity demand application, perhaps more categories in the afternoon than in the night would be desirable.

The work we have done creates a benchmark itself. The methods given here enable creation of scenarios for multiple time steps that incorporate forward predictors. The models can

be based on relatively small amounts of historic data. This provides an important tool for applying stochastic programming to problems such as unit commitment where there are forward predictors, multiple sub periods, but relatively little historic data.

References

- [1] M. Carrion and J.M. Arroyo. A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. *Power Systems, IEEE Transactions on*, 21(3):1371–1378, 2006.
- [2] Jitka Dupaov, Giorgio Consigli, and SteinW. Wallace. Scenarios for multistage stochastic programs. *Annals of Operations Research*, 100:25 – 53, 2000.
- [3] Andreas Eichhorn, Holger Heitsch, and Werner Römisch. Stochastic optimization of electricity portfolios: Scenario tree modeling and risk management. In Steffen Rebennack, Panos M. Pardalos, Mario V. F. Pereira, and Niko A. Iliadis, editors, *Handbook of Power Systems II*, Energy Systems, pages 405–432. Springer Berlin Heidelberg, 2010.
- [4] E. Ela and M. Milligan. A flexible power system operations simulation model for assessing wind integration. In *Power and Energy Society General Meeting, 2011 IEEE*, pages 1 – 8, 2011.
- [5] Y. Feng, D. Gade, S. M. Ryan, J-P Watson, R. J-B Wets, and D. L. Woodruff. A New Approximation Method for Generating Day-Ahead Load Scenarios. In *Proceedings of the 2013 IEEE Power and Energy Society General Meeting*, 2013.
- [6] Michal Kaut and Stein W. Wallace. Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, 3(2):257–271, 2007.
- [7] Wai-Kei Mak, David P. Morton, and R. Kevin Wood. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters*, 24:47 – 56, 1999.
- [8] Henrikas Pranevicius and Kristina Sutiene. Scenario tree generation by clustering the simulated data paths. In *Proceedings 21st European Conference on Modelling and Simulation*, 2007.
- [9] J.O. Royset and R. J-B Wets. Fusion of soft and hard information in nonparametric density estimation. Technical report, Naval Postgraduate School, 2013.
- [10] Johanness Royset and Roger J-B. Wets. Epi-splines and exponential epi-splines: Pliable approximation tools. Technical report, Naval Postgraduate School, 2013.
- [11] S. M. Ryan, C. Silvia-Monroy, J-P Watson, R. J-B Wets, and D. L. Woodruff. Toward Scalable, Parallel Progressive Hedging for Stochastic Unit Commitment,. In *Proceedings of the 2013 IEEE Power and Energy Society General Meeting*, 2013.

- [12] Alexander Shapiro. Monte carlo sampling methods. In A. Ruszczyński and A. Shapiro, editors, *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*, pages 353 – 425. Elsevier, 2003.
- [13] S. Snapp. Leading indicator forecasting in spp.
- [14] S. Takriti, J.R. Birge, and E. Long. A stochastic model for the unit commitment problem. *Power Systems, IEEE Transactions on*, 11(3):1497–1508, 1996.
- [15] Daobao Xu, Zhiping Chen, and Li Yang. Scenario tree generation approaches using k-means and $\{LP\}$ moment matching methods. *Journal of Computational and Applied Mathematics*, 236(17):4561 – 4579, 2012.