Paper accepted for presentation at 2009 IEEE Bucharest Power Tech Conference, June 28th - July 2nd, Bucharest, Romania

Supervised learning of intra-daily recourse strategies for generation management under uncertainties

Bertrand Cornélusse, *Student Member, IEEE*, Gérald Vignal, Boris Defourny and Louis Wehenkel, *Member, IEEE*

***Abstract*—The aim of this work is to design intra-daily recourse strategies which may be used by operators to decide in real-time the modifications to bring to planned generation schedules of a set of units in order to respond to deviations from the forecasted operating scenario. Our aim is to design strategies that are interpretable by human operators, that comply with real-time constraints and that cover the major disturbances that may appear during the next day. To this end we propose a new framework using supervised learning to infer such recourse strategies from simulations of the system under a sample of conditions representing possible deviations from the forecast. This framework is validated on a realistic generation system of medium size.**

***Index Terms*—Mixed integer linear programming, generation planning, uncertainty management, machine learning.**

I. Introduction

In the electricity generation management context, the numerous sources of uncertainty imply that the problems related to different time horizons are treated as multistage decision problems with recourses. In this work we focus on the design of very short term (intra-daily) recourse strategies. These strategies are used by operators to decide in real-time the modifications to bring to the generation schedules of a set of units in order to respond in a safe and economically efficient way to deviations from the forecasted operating scenario. To be useful, such recourse strategies must be interpretable by human operators and comply with real-time constraints. At the same time, they should cover all the major likely or unlikely disturbances that may appear during the day, such as the loss of any generating unit and/or significant deviations from forecasted demand conditions. In current practice, a reference schedule (unit commitment) of all power plants is typically computed one day ahead for the next 24 hours, by using a detailed model of the generation system and an appropriate dynamic optimization algorithm (see, e.g., [1] for more details about a formulation and solution method for this problem), while the intra-daily recourse strategies are pre-determined by experts during off-line studies.

Within this context, the purpose of our work is to develop a systematic and essentially automatic approach to (re)design intra-daily recourse strategies compatible with real-time constraints and interpretable by human operators, so as to allow them to respond to an a priori defined set of disturbances in a near-optimal way. The proposed approach consists in pre-computing the day ahead optimal adjustments of the generation schedule for the time slots where these adjustments can be made and for a representative set of disturbance scenarios. The resulting database is then fed to a supervised learning algorithm which interprets the information and computes decision rules for the intra-daily recourses. The decisions are further post-processed in order to comply with real-time feasibility constraints and validated on an independent set of disturbance scenarios, before they are handed over to the operators. An interesting byproduct of supervised learning is to help understanding the influences of the different sources of uncertainty on these operating strategies.

The rest of the paper is organized as follows. Section II describes the different steps of the proposed approach and provides background information about supervised learning and intra-daily generation management tools. Section III reports on a detailed case study with a representative test problem of medium size, and Section IV discusses our proposal with respect to related work in the context of multistage stochastic programming. Finally, section V concludes and discusses directions for further work.

II. Methodology

We suppose that we have an optimization algorithm to compute an *open-loop* generation schedule for a period of time ahead (typically a few hours) from a model of the considered generation system and for a given scenario describing the electric load that has to be served and the availabilities of the generation units over this time-period. In practice, such algorithms are routinely used by generating companies to derive every day the planned operation of their assets for the next 24 hours.[1] In order to handle in real-time deviations from the forecasted scenario (see upper and middle parts of Figure 1), operators use so-called *recourse strategies*, which are decision rules mapping information obtained about the actual realization of the scenario at some predefined time steps (e.g. at times t_1 and t_2 in the lower part Figure 1) towards adjustments of the planned generation schedule of the subsequent time steps.

We propose an approach to compute these recourse strategies from the information available the day ahead, in the same environment where the operation plan of the units is computed for the next day and using the same optimization algorithm

Bertrand Cornélusse, Boris Defourny and Louis Wehenkel are with the Department of Electrical Engineering and Computer Science, University of Liège, B-4000 Liège, Belgium. Gérald Vignal is with the OSIRIS department, EDF R&D, 1 avenue du général de Gaulle, F-92141 Clamart Cedex, France. (E-mail: bertrand.cornelusse@ulg.ac.be, Boris.Defourny@ulg.ac.be, Louis.Wehenkel@ulg.ac.be).

[1]To simplify the presentation, but without loss of generality, we do not consider the case where the decision making process explicitly implies price-forecasts and decision variables representing market operations.

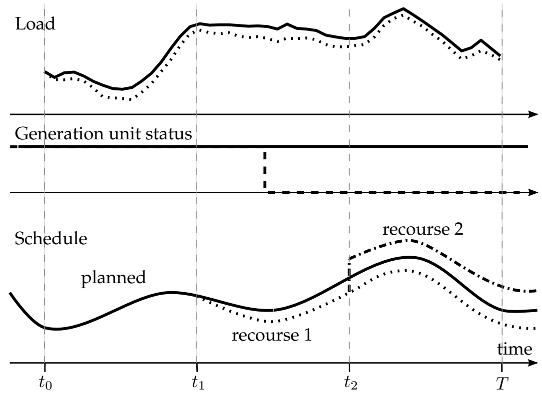


Fig. 1. Top: Forecast (solid) and realization (dotted) of the load curve (part of the scenario). Middle: Forecast (solid) and realization (dashed) of the availability of a generation unit (other part of the scenario). Bottom: illustration of a recourse strategy. The solid curve depicts the planned actions, the other curves represent the recourses taken at different time steps.

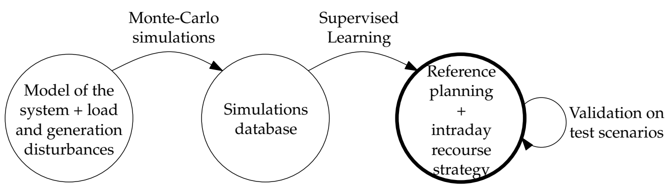


Fig. 2. Schematic overview of the proposed supervised learning-based approach for building intra-daily recourse strategies.

that is used to compute this plan. The proposed approach is depicted schematically on Figure 2. It is composed of the following main ingredients:

1) the generation of a set of perturbed scenarios around a reference scenario,
2) the re-optimization of the generation planning for each perturbed scenario from the instant we take the recourses (t_r) to the end of the optimization period (T),
3) the processing by supervised learning of the information contained in the optimally adjusted plannings to build a near-optimal recourse strategy,
4) the post-processing of the learnt rules to impose feasibility (e.g. dynamic, coupling) constraints,
5) the validation of the learnt and/or post-processed rules on an independent set of scenarios.

These steps are further described in the following subsections.

A. Generation of perturbed scenarios

We need a way to generate a set of demand and availability scenarios representing the range of deviations from the forecast that we want to cover with our recourse strategies. In practice, there are essentially two approaches to gather a set of such scenarios, namely their collection from actual operation of the system or the use of a Monte-Carlo simulation approach exploiting a probabilistic model of possible deviations from forecasts. In our approach, we can use either of these two approaches, or even a combination of them, since the input to the next step is merely a set of time series representing the deviation of load from the forecast and the moments at which a particular generation unit becomes unavailable.

In our case study, we start from the scenario containing the day ahead load forecast and the generation units availabilities used to compute a reference schedule. Then we generate different perturbed scenarios randomly around the reference scenario (Figure 3(a)) by combining a probabilistic model of load-forecast errors and a screening of generation outages.

B. Re-optimization of perturbed scenarios

Consider a set D of load patterns, and a set O of generation units which could become unavailable next day. Let $\pi^\star$ be the optimal planning associated to a reference load scenario $D^\star \in D$. Suppose that we want to compute an optimal recourse strategy $\sigma^*_{t_r}(\xi_{t_r})$ for a single a priori fixed recourse time $t_r \in \{t_0, t_0+1, \ldots, T\}$, i.e. we want to know the modifications to bring to all the units from time t_r+1 to T once the real behavior ξ_{t_r} of the system between time t_0 and t_r is known. Let S be the set of scenarios made of one demand of D and of a unit outage of O imposed at a time in $\{t_0, ..., t_r\}$. First, we compute the plannings π_s for each scenario $s \in S$ by imposing the planning $\pi^\star$ for times t_0 to t_r and by using the given optimization algorithm so as to adjust the planning for time t_r+1 to T.

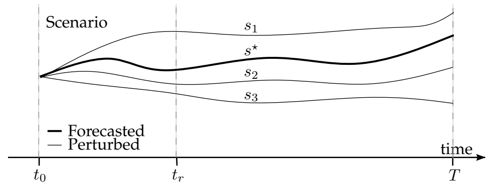
The difference $\pi_s - \pi^\star$ illustrates the impact of the demand variation and the unit outage of scenario s on the generation schedule (Figure 3(b)). This adjustment actually represents the difference between the open loop plan computed the day ahead from the forecasts and the optimally adjusted plan if at time t_r perfect information became available about the realization of the scenario for the whole period t_0 to T. Notice that in real-time operation the information available at time t_r to take the recourse decision is not so strong; while it may perfectly describe the realization up to t_r it will in general only reduce, but not totally remove, the uncertainty about the future realization of the process during (t_r, T).

Therefore, the re-optimization of the perturbed scenarios provides a database of generation adjustments which are optimistically biased because they assume perfect information about the behavior of the system at subsequent time steps. This over-fitting of the sample of perturbed scenarios is however partially countered by the application of supervised learning at the next step of the proposed approach which enforces the projection of this information on a set of non-anticipative decision strategies which are only function of the information available at time t_r.

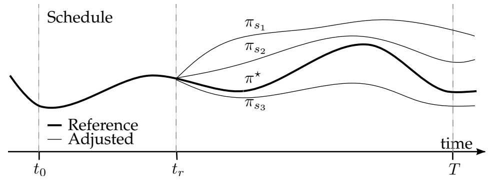
C. Supervised learning application

1) Supervised learning background: Machine learning is a sub-field of artificial intelligence in which algorithms are designed to make a computer able to learn from data. Among the different fields of machine learning stands supervised learning. In the supervised learning (SL) paradigm, the data is organized as a collection of N objects $\{(x_i, y_i)\}_{i=1}^N$. Classically, each object i is described by a vector of input features[2] $x_i = (x_i^1, \ldots, x_i^n) \in \mathcal{X}$ and an output value $y_i \in \mathcal{Y}$. The

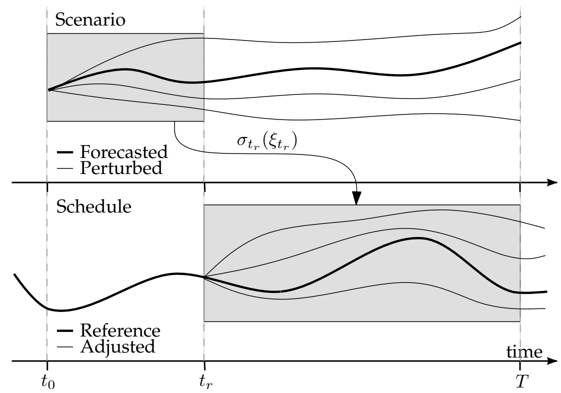
[2]Subsequently in this paper, we will use subscripts $_i$ to denote samples, while we will use superscripts ^{k} to denote specific features, while denoting input and/or output information.



(a) Generation of perturbed scenarios.



(b) Re-optimization on the recourse period.



(c) Illustration of the supervised learning task.

Fig. 3. The main steps to compute an approximation of $\sigma^*_{t_r}(\xi_{t_r})$.

goal is to find a mapping $f : \mathcal{X} \mapsto \mathcal{Y}$ between the features and the output values. Ideally we would like f to minimize the expected loss $L = E_{x,y}\{\ell(f(x), y)\}$ (according to a predefined loss function $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto R_+$) over the whole input-output space. As we usually do not know the distribution $p(x, y)$, most supervised learning algorithms actually search for a function f which minimizes an estimate of L (e.g. the empirical loss $\frac{1}{N}\sum_{i=1}^{N} \ell(f(x_i), y_i)$).

2) Supervised learning problem formulation: We want to exploit the simulations of section II-B to formulate a supervised learning problem in order to derive an approximation of $\hat{\sigma}^*_{t_r}(\xi_{t_r})$ that will serve as a recourse strategy mapping the information available at t_r to generation adjustments at subsequent time steps. The direct formulation of this supervised learning problem would consist in using as output space $\mathcal{Y}$ a set of multi-dimensional time-series representing the possible power level evolutions of all generation units for $t > t_r$, with the goal of approximating these latter as a function of the state of the generation system at time t_r and the information about the realization of the load and generation availability scenario collected until t_r (which are also time series). This is schematically depicted in Figure 3(c).

In order to apply standard supervised regression algorithms, which operate with a scalar (i.e. one-dimensional) output space, we simplified this problem by reducing it to a set of elementary supervised learning problems, one for each generation unit. This yields, for each individual unit, a supervised learning problem, where the output is now a one-dimensional time-series describing the adjustments of this particular unit. Furthermore, in order to tackle the temporal dimension of the outputs, we used a supervised learning formulation where the temporal dimension is explicitly represented in the input features. Thus, if the original problem is formulated over M units, our decomposition yields M elementary regression problems, where the input space is the Cartesian product of the original input space $\mathcal{X}$ and the recourse time interval, and where the output is a real-number representing the generation level of a specific unit as a function of the information gathered at time t_r and the considered prediction time-step $t_p > t_r$.

The overall description of the supervised learning of recourse strategies is summarized in Tables I and II.

TABLE I

AN ITEM OF THE LEARNING SET FOR A GENERATION UNIT RELATED SUB-PROBLEM

Inputs	**output**
• state of the system at time t_r • observed demand deviation from forecasting until time t_r, • unit(s) outage before time t_r, • prediction time t_p,	power of the unit at time t_p.

TABLE II

CONSTRUCTION OF A RECOURSE STRATEGY.

Input: an optimal planning $\pi^\star$ associated to a reference demand scenario $D^\star \in D$.
Output: a recourse strategy $\hat{\sigma}_{t_r}(\xi_{t_r})$ associated to $\pi^\star$ for the recourse time t_r.

1. Let S be the set of scenarios made of one demand of D and of a unit outage event,
2. compute a planning π_s adjusted from $t_r + 1$ to T for each scenario $s \in S$ by imposing the planning $\pi^\star$ for times t_0 to t_r,
3. solve a set of M supervised learning problems in order to derive an approximation $\hat{\sigma}^*_{t_r}(\xi_{t_r})$ of the optimal recourse strategy for time t_r, ξ_{t_r} being the state information available at time t_r.

Once the overall problem has been reduced to a number of standard regression problems, one could in principle apply any available supervised regression algorithm. In our preliminary investigations, we have applied several such algorithms, in particular the so-called $\varepsilon-$support vector machines for regression (ε-SVR) [2], and a variety of methods based on regression trees [3]. In these investigations, we found that the so-called Extra-Trees supervised learning method [4] yielded in general the best compromise between accuracy and computational efficiency, so we decided to stick to this method in our case study reported below. For the sake of completeness, we recall in the next subsection the rationale and the main features of this method. The reader who is already familiar with this material, may skip this section.

3) The Extra-Trees supervised learning method: In this section we recall the main principles of the so-called Extra-Trees supervised regression method by stressing its characteristics of interest in our application. Note that this section is very similar to the section II.B of [5].

Tree-based supervised learning is well known for its computational efficiency, interpretability, robustness to outliers, and its capability to cope with high-dimensional problems with a large number of irrelevant input features. The idea is to recursively split the training set, thanks to tests on the value of the features, in several subsets in order to decrease a measure of impurity until the subsets are composed of object sufficiently similar in the output space. In regression trees, a usual impurity measure of a subset is the variance of the output variable y: the higher the variance of y in a subset of objects, the more heterogeneous its objects.

A classical *single* tree induction algorithm thus works as follows. First the training sample TS is attached to the top node of the tree. Then for each node, three steps are applied:

- evaluation of the necessity to split the node,
- if no, the node becomes a terminal node (a leaf) and a label defining the output value is assigned to this node,
- otherwise, the node becomes an internal node (a test node), and the feature and its cut-off value that define together how to split the node are determined, so as to partition its associated set of objects in two subsets which will correspond to two new nodes of the tree that form the two children of the considered test node.

If we use the mean square error criterion to estimate the accuracy of the tree predictor, it turns out that assigning the mean value of the outputs of the objects constituting a terminal node is optimal with respect to empirical prediction error.

To split a node, a test is defined by a feature x^k ($k \in \{1, \ldots, dim(\mathcal{X})\}$) and a cut-off value ($v^k$). All the objects satisfying the test $x^k > v^k$ are assigned to the right descendant node and the remaining ones are assigned to the left descendant node. To find the best test, a score is computed for every input feature and for every possible cut-off value. For regression trees, a typical score is the decrease of output variance in the two descendant nodes with respect to the output variance of the current node. The test with the highest variance reduction is thus chosen. Notice that, if the mean square error is chosen as error criterion, the split with the highest output variance reduction turns out to be the split that is optimal in terms of the reduction of the empirical prediction error.

It is more complicated to assess if a node should be split. A classical way is to stop splitting when the number of objects in a node is below a threshold value n_{min}, but many other techniques have been developed to identify the tree of optimal complexity (pruning methods). For a more complete description of these pruning algorithms see for example [3].

While the learning of single regression trees is computationally very efficient and often leads to highly interpretable decision rules, is has however been shown that single tree-based methods have a high *learning variance*[3] [6], which implies that they are often suboptimal in terms of accuracy, specially on problems where the information is spread among a large number of equally relevant features.

[3]The *learning variance* of a learning algorithm quantifies the dependence of the models that this algorithm produces with respect to the random nature of the training sample. In practice, high learning variance implies low accuracy. This type of variance should not be confused with the variance of the output variable used in splitting procedure to develop the regression trees.

Therefore, tree-based ensemble methods have been introduced to decrease variance and to allow them to cope with very complex tasks such as image, text and time-series classification. The general idea behind these methods is to avoid giving a single tree the capability of modeling the whole training set. This can be achieved either by perturbing the training set, either by perturbing the construction algorithm, in order to build from a training set an (often) very large set of different trees, and by deriving the prediction h by aggregating in some fashion (e.g. by voting or by averaging) the predictions derived from each tree in the ensemble.

A major cause of learning variance of regression trees is the sensitivity of test nodes cut-off value to the content of the training set. The main aim of the Extra-Trees [4] is to mitigate this behavior by randomly perturbing the structure of the trees, thus decreasing their dependence on the training set. During the construction phase of a single tree in this method, the search for the best feature and the best threshold at each node is somewhat randomized. The level of randomization is related to the size of the subset of input features which are considered in the search of the best split according to a given score measure. This is controlled through the parameter K. In addition, for each feature of the subset the threshold is also randomly chosen in its variation interval. Except for the above, each of the T trees is built on the whole training set using a classical top down induction algorithm, without pruning. Because all the trees are built independently and because the induction procedure is simplified, this algorithm is computationally very efficient. The prediction of the ensemble is obtained by averaging the prediction of the single trees. With respect to classical single trees, the accuracy of this method is in general dramatically increased.

Notice that in addition to producing fast and often very accurate decision rules, the Extra-Trees method produces as a byproduct a scoring of the input features in terms of their usefulness to predict the output information. These so-called *variable importances* may be used in practice to analyze the impact of the different features and hence to better understand the problem under consideration. We will use these importance measures in our case study to analyze the impact of different features on the recourse decisions. We refer the interested reader to [7] for further information about the computation and nature of these variable importances.

D. Post-processing of predicted recourses

Given the way we have formulated the supervised learning problems, we expect to obtain recourse strategies which are close to optimal, but which do not necessarily satisfy coupling constraints among generating units and which may also not satisfy some of the individual operating constraints of each unit (e.g. dynamic constraints limiting ramping or start-up/shut-down times, or non convex constraints defining the range of admissible power levels). Therefore, a post-processing stage must be applied to these strategies.

We first notice that data on which the learning is done satisfies coupling constraints and also the individual dynamic and operating constraints for each unit (see section III-C). For example, the generation interval of a unit is non-convex: it is either 0 MW, either between P_{min} and P_{max}, a unit must stay on for a certain amount of time once it has been started, ... However, the Extra-Trees algorithm produces decision strategies which are essentially convex combinations of the output values of the training sample, and which in general do not satisfy all constraints.

In the current version of our approach, we decided to post-process the decisions produced by the learning algorithm for a given generation unit, so that they satisfy the individual dynamic and operating constraints of this unit, so as to yield a feasible operating strategy. On the other hand, we decided to not enforce the coupling constraints, hence if the adjusted generating plan leads to a gap between the total generation and the total load, this gap is compensated by reserve energy purchase and is (strongly) penalized when evaluating cost induced by this strategy. As we will see however in our case study, the resulting increase of cost is in practice rather small and hence does not jeopardize the interest of the approach.

To ensure the satisfaction of real-time feasibility constraints of a given unit, we impose them a posteriori, at the moment where the recourse action is applied. First we compute a recourse based on the information gathered in ξ_{t_r} and the decision rules built by supervised learning. Then we modify these recourses and impose constraints unit by unit. To do this, we compute the closest recourse that satisfies the constraints of the unit, by formulating a simple optimization problem which may be solved quickly and independently for each unit.

E. Validation of the recourse strategies

In order to validate the recourse strategies computed by supervised learning, we use an independent set of scenarios in the following way. First, each scenario is solved optimally, in the same fashion as we computed the recourse decisions for the learning sample. This yields for each validation scenario a generation schedule that minimizes the costs of operation under the hypothesis of perfect information. Then, for each scenario the recourses are computed by using the Extra-Trees based decision rules, by post-processing them, and by computing the overal induced operating costs, including the penalization of the possible violation of the coupling constraint. Finally, by comparing the resulting costs, we may measure the distance between the inferred strategies under different conditions (e.g. using different settings of the Extra-Trees method, different sets of input features, or different sizes of learning samples) and assess them also with respect to the (admittedly unreachable) ideal strategy based on perfect information, or any other candidate strategy.

III. Case study

A. Test system

The medium-sized test system we use is composed of eight thermal units of different capacities, generation costs and technical characteristics, and three hydroelectric valleys. It comes from real data provided by Electricité de France (EDF). The simulations follow EDF's industrial practice : to compute the reference planning for the next day the optimization horizon is of two days and is divided into 48 periods of 30 minutes.

The corresponding optimization model contains $30,000$ variables, one half of them being binary, and $40,000$ constraints [8]. The optimization problems are solved using the branch-and-cut algorithm implemented in CPLEX [9].

We then consider a single recourse stage t_r located at 6 AM and apply our framework to compute the recourse strategy on a half-hourly basis until the end of the day, based on the deviations observed between 0 and 6 AM.

B. Scenario generation

The load scenario generator uses a statistical model of the load derived from three years of historical data, gathered in a vector $d = (d_1, ..., d_N)$, where N is the complete horizon. Each day contains T samples of the historical load. Each component of this vector is decomposed in a seasonal part plus a daily mean-corrected part:

$$d_n = \bar{d}_i + d_{i,\tau}, \quad n = 1, ..., N.$$

In this expression, i is the day corresponding to the time step n, $i = \lceil n/T \rceil$, and τ is the time step inside this day:

$$\tau = n - (i-1) \times T.$$

The seasonal part is obtained by taking the average load for each day: $\bar{d}_i = 1/T \sum_{n:\lceil n/T \rceil = i} d_n$. The daily part is gathered in a matrix $D \in \mathbb{R}^{\lceil N/T \rceil \times T}$. We then assign a category function of the load profile to each row of D, i.e. to each day record. For clustering we used an algorithm similar to the k-means called pam, which is based on the medoids instead of the means [10], and chose the value of $k = 4$ by experiment. Roughly, this value of k divides days in the following categories: working and week end days for the two daily saving time periods. We then use classical time series modelling tools to obtain an auto regressive model of the error between the forecast and the realization in each category. These models allow us to generate perturbations around a given scenario. We finally add a constant offset to each generated signal using the distribution of the daily mean load for the corresponding cluster.

As concerns unexpected outages of generation units, we consider that only a single unit may be outaged in each scenario between 0 and 6 AM and may not be restarted until 12 PM of the same day.

C. Mathematical optimization model

For the sake of completeness we provide a synthetic description of the generation scheduling model that we used, more details may be found in [8], and a discussion of similar models in [11] or [12]. Here we merely provide an enumeration of the decision variables for each type of generation unit, of the type of constraints that apply to their dynamics and of the coupling constraints. The overall problem is then stated as a Mixed Integer Linear Program. As already mentioned, although we focus on one day ahead planning, the optimization is run over two consecutive days so as to avoid side effects at the end of the first day.

1) Thermal units: At each time step the power generated is either 0 MW or a value in the range $[P_{\min}, P_{\max}]$, where $P_{\min} > 0$ MW. Hence both a binary and a real variable are needed to model the power of a unit. Also two variables are needed to model the contribution of a generation unit to primary and secondary reserves and some constraints restrict the power that is actually generated to fulfill these reserves. Two additional variables represent start-ups and shut-downs, and a last one the start-up cost which is a function of the shut-down time. Additional constraints model the minimum up and down times respectively, as well as maximum ramp rates. A fixed cost as well as a cost proportional to the power generated are incurred when the unit is on.

2) Hydro-electric generation: A valley is considered as a single generation unit, with water levels of reservoirs constrained in an acceptable range. We model the possibility to spill some water when a reservoir is full. Valleys may contain pumped storage. Generation plants have a piecewise linear power-water flow curve to model the sequential activation of turbines. The contribution to primary reserve is a percentage of the power generated and some secondary reserve levels are associated to break-points in the piecewise linear curve. The actual secondary contribution is interpolated from these values and the value of the water flow. Thus a valley is represented as a graph where reservoirs and plants are nodes, and we must decide the amount of water that flows (real variables) along the edges and hence the number of turbines or pumps to activate in each plant (binary variables) at each time step. From these variables we express the other variables: the generation level, the primary and secondary contributions and the level of the reservoirs. An opportunity cost is assigned to the water of some reservoirs to penalize (or reward) the use of water.

3) Coupling constraints: Load as well as ancillary services fulfillment constitute constraints linking all the generation units. In each such constraint a slack variable compensates lack or surplus of generation or reserve.

4) Objective function: The objective function gathers the thermal and hydro-electric generation costs as well as the penalization of the slack variables of the coupling constraints. Except for the surplus of reserve which are not penalized, a piecewise linear and convex penalization function is used with slope values depending on the type of slack variable.

D. Evaluation of the approximate recourse strategies

We have treated the sub-problems (cf. Table I) related to the approximation of a recourse strategy for each generation unit. The Extra-Trees clearly outperformed the ε-SVR on these problems, and we thus only report on their results.

We have analyzed the optimality in terms of generation costs of the plannings yielded by the recourse strategies approximated with the Extra-Trees and after the post-processing stage. For each sub-problem we built an ensemble of 100 trees using default settings for $n_{\min}$ and K (see [4]).

Figure 4 shows the obtained results on a set of about 400 independent test scenarios: each point refers to a scenario of deviations combining the loss of a generation unit before 6 AM and a deviation of the load curve from its forecast. Over

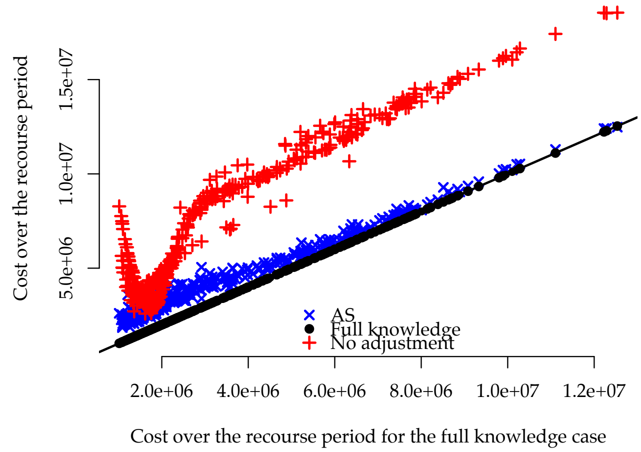


Fig. 4. Scatter plots of the planning cost during the recourse period vs. the cost of the planning adjusted having a full knowledge.

the horizontal axis, these scenarios are sorted according to the total cost associated to them if perfect (full) knowledge of the scenario is exploited to re-optimize the generation plan; over the vertical axis they are sorted according to the actual incurred cost depending on the adjustment strategy. For each scenario, three different adjustments have been evaluated, corresponding to three different points at the same horizontal coordinate:

- the first strategy consists in applying no recourse action at all (the corresponding points are depicted using red $+$ symbols). In this case the loss of the generation unit and the deviation of the load are compensated by purchasing rather expensive reserves (their price is modeled by a piecewise linear and convex function). This strategy constitutes the worst case behavior;
- the second strategy (represented by black $\bullet$ symbols) corresponds to the perfect information case. In this case the points are located on the line $y = x$; their cost on the vertical axis represent a lower bound for all possible recourse strategies;
- the last strategy is the one built using the proposed procedure as described in Table II to build an approximate strategy (AS) (it is represented by blue $\times$ symbols).

We note that our approximated plannings yield costs which are often much lower than the not-adjusted reference planning and quite close to those assuming perfect knowledge.

Figure 5 is another representation of the information contained in Figure 4. It is a cumulative histogram of the additional cost induced by the strategies compared to the cost of the plannings deterministically optimized knowing a perfect forecast of the system conditions. The red curve corresponds to the red $+$ symbols of Figure 4, while the black curve corresponds to the $\times$ symbols. The last (green) curve illustrates the reduction of the performance of approximate strategy when a ten times smaller number of training simulations are used for learning it (in this case 43 training scenarios are used instead of 432).

To analyze the influence of input features on the recourse decisions, we show their variable importances in Figure 6,

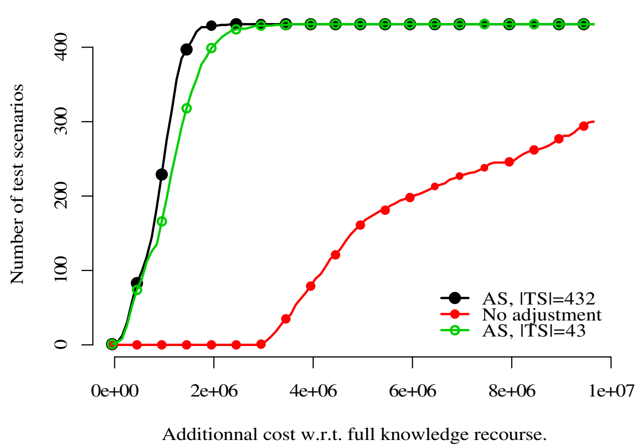


Fig. 5. Cumulative histogram of the additional cost compared to the optimally adjusted plannings. $|TS|$ denotes the size of the training set.

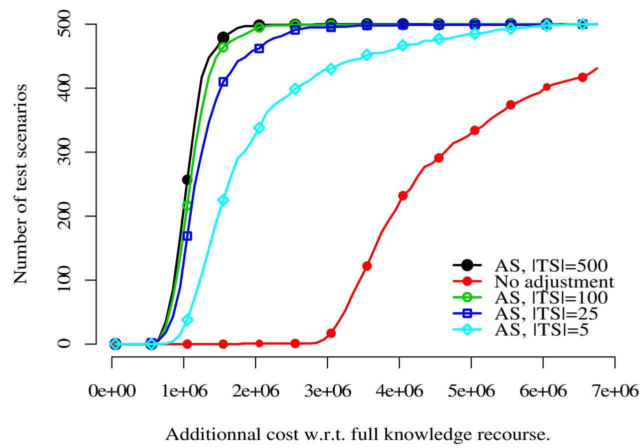


Fig. 7. Cumulative histogram of the additional cost compared to the optimally adjusted plannings when considering only load deviations.

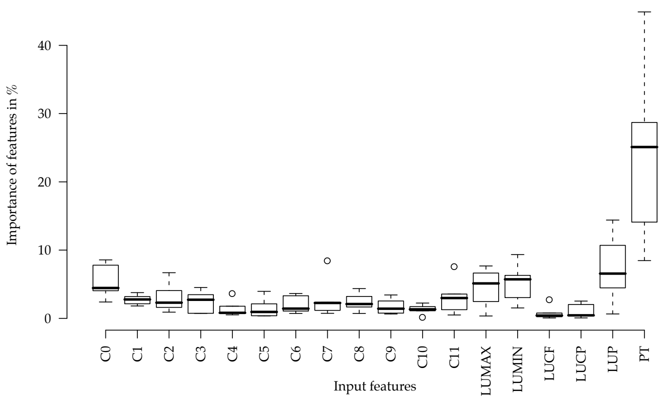


Fig. 6. Input features importances distributions.

where the horizontal axis corresponds to the different input features and the vertical axis to their relative importances. For each feature, a box plot represents the distribution of its importances over the decision rules corresponding to the different generating units. We observe that the most important variable is the prediction time (denoted by PT) while the next most important ones correspond to the size of the generating unit that is lost (denoted by $LUMAX$ and $LUMIN$ respectively). The variables denoted by C_i correspond to the deviation of the load scenario with respect to the forecast.

Finally we illustrate on Figure 7 results obtained when building recourse strategies to cover only load deviations. We observe that in this case there is no scenario for which the approximate strategies yield costs equal to the cost of the full knowledge strategy. This results from the fact that from the sole observation of the load deviation during the interval 0 to 6 AM it is actually not possible to perfectly predict its subsequent values and hence the generation schedule computed by assuming full information is not non-anticipative. As we observe from Figure 7 there seems to be a cost gap of about 5×10^5 between the full knowledge strategy and our non-anticipative approximated strategy.

Conversely, if one considers scenarios corresponding to the loss of a very large generation unit, it becomes possible to predict almost exactly the prefect knowledge reschedule from the sole information gathered at time t_r. This may be observed on Figure 4, where the right-most scenarios (which correspond to the loss of a very large unit together with a strong increase of load) lead to almost identical costs for the approximate strategy and for the strategy assuming perfect knowledge. The optimal recourse for these scenarios consists roughly in starting up as soon as possible a standby unit and by ramping up to their maximum level the already running ones.

IV. Relation with Stochastic Programming

A closely related framework, which has already been studied in the literature ([13], [14]), is the multistage stochastic programming (SP) framework. In this approach one also makes use of scenarios to represent realizations of the uncertain processes. The two-stage case has a single recourse stage like in our approach: the first stage decision corresponds to the scheduling from time t_0 to time t_r, while the recourse corresponds to the scheduling from time t_r+1 to time T and may be adjusted according to the realization of ξ_{t_r}. This paradigm thus consists in optimizing jointly the first stage decision and the recourses, which leads to the statement of a huge optimization problem comprising decision variables and constraints for all the scenarios and where the objective is to minimize the average cost over the set of scenarios. Moreover, non-anticipativity constraints need to be imposed, in order to enforce a single first stage decision and to enforce recourse decisions which depend only on the information in ξ_{t_r}. On the other hand, our method assumes that the first stage decision is computed beforehand and is the starting point to compute the recourse strategy. Then the non-anticipativity of recourse decisions is imposed during the learning phase and dynamic constraints are imposed by post-processing the resulting recourse decisions.

Thus recourse decisions are optimized according to the first stage, but the converse is not true. However, our approach may also be valuable to compare different first stage decisions because it makes it practically possible to construct explicit decision strategies[4] and to evaluate them by Monte-Carlo simulation with reasonable computing resources. Remember that the different scenarios are solved independently, leading to a set of tractable optimization problems which may be solved in parallel with existing unit-commitment software packages.

V. Discussion

In this paper we have proposed a novel approach based on Monte-Carlo simulations and supervised learning for the systematic and essentially automatic design of near-optimal recourse strategies for intra-daily generation management. The approach is intrinsically scalable to large scale generation management problems, and may in principle handle all kinds of uncertainties and practical constraints. It also provides interpretable information in the form of explicit decision strategies and measures of influence of different parameters on the decision strategies. This approach has been evaluated through a detailed case study on a medium size generation scheduling problem representative for a single intra-daily recourse stage involving a mixed hydro-thermal unit commitment. Our results show the feasibility of the approach and are also very promising in terms of economic efficiency of the resulting strategies.

Immediate further work will aim at extending the approach to multiple recourse stages, to improve the treatment of dynamic and coupling constraints in the post-processing module, to cover other kinds of uncertainties, and to further validate the approach on other instances of the problem.

Over the longer term, we plan to apply similar approaches to other contexts in sequential decision making under uncertainty and also to couple the construction of the reference decision strategy with the construction of the recourse strategies. Indeed, ideally the reference schedule should be computed in such a way that the optimal recourses strategies are of maximal economic efficiency and safety, while satisfying the practical (context dependent) constraints.

[4]Contrary to the output of SP which is a set of decisions attached to the nodes of the scenario tree.

Acknowledgements

BC thanks EDF and FRIA (Belgian Fund for Research in Industry and Agriculture) for allowing him to carry out this research as well as PEPITe, a spin-off of the University of Liège, who gave him the opportunity to use the data mining software PEPITo. This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors.

References

[1] L. Dubost, R. Gonzalez, and C. Lemaréchal, “A primal-proximal heuristic applied to the french unit-commitment problem,” *Mathematical Programming*, vol. 104, pp. 129–151, 2005.

[2] A. Smola and B. Schölkopf, “A tutorial on support vector regression,” NeuroCOLT2 Technical Report Series, Tech. Rep., 1998.

[3] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Wadsworth and Brooks, 1984.

[4] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” *Machine Learning*, vol. 36, no. 1, pp. 3–42, 2006. [Online]. Available: http://www.montefiore.ulg.ac.be/services/stochastic/pubs/2006/GEW06a

[5] B. Cornélusse, C. Wera, and L. Wehenkel, “Automatic learning for the classification of primary frequency control behaviour,” in *Proc. IEEE Power Tech Conference, Lausanne, July 2007*, 2007, pp. 273–278.

[6] P. Geurts, “Contributions to decision tree induction: bias/variance tradeoff and time series classification,” Ph.D. dissertation, University of Liège, Belgium, May 2002.

[7] V. A. Huynh-Thu, L. Wehenkel, and P. Geurts, “Exploiting tree-based variable importances to selectively identify relevant variables,” *JMLR: Workshop and Conference Proceedings*, vol. 4, pp. 60–73, 2008. [Online]. Available: http://www.montefiore.ulg.ac.be/services/stochastic/pubs/2008/HWG08b

[8] B. Cornélusse, “Application of supervised learning to very short-term decision making for electric power generation,” Master’s thesis, Université de Liège, 2008.

[9] ILOG, *ILOG CPLEX 11.0 User’s manual*, September 2007.

[10] L. Kaufman and P. J. Rousseeuw, *Finding Groups on Data: an Introduction to Cluster Analysis.* John Wiley & Sons, 2005.

[11] M. Carrion and J. Arroyo, “A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem,” *Power Systems, IEEE Transactions on*, vol. 21, no. 3, pp. 1371–1378, Aug. 2006.

[12] M. P. Nowak and W. Römisch, “Stochastic lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty,” *Annals of Operations Research*, vol. 100, no. 1-4, pp. 251–272, December 2000. [Online]. Available: http://www.springerlink.com/content/p31567037m133467/

[13] P. Carpentier, G. Cohen, J. Culioli, and A. Renaud, “Stochastic optimization of unit commitment: a new decomposition framework,” *Power Systems, IEEE Transactions on*, vol. 11, no. 2, pp. 1067–1073, 1996.

[14] N. Gröwe-Kuska and W. Römisch, *Applications of Stochastic Programming*, ser. MPS-SIAM Series in Optimization. SIAM, 2005, ch. Stochastic unit commitment in hydro-thermal power production planning. [Online]. Available: http://www-iam.mathematik.hu-berlin.de/ romisch/papers/Chapter30.pdf