A Dynamic Multistage Stochastic Unit Commitment Formulation for Intraday Markets

Bita Analui and Anna Scaglione

***Abstract*—As net-load becomes less predictable there is a lot of pressure in changing decision models for power markets such that they account explicitly for future scenarios in making commitment decisions. This paper proposes to make commitment decisions with multiple gate closures. Our proposed model also leverages a state-space formulation for the commitment variables, through which the operational constraints of generation units participating in the market are respected. We also study the problem of constructing scenario tree approximations for stochastic processes and evaluate our algorithms on scenario tree libraries derived from real net-load data.**

***Index Terms*—Stochastic unit commitment, multistage stochastic optimization, intraday markets, load scenario tree library.**

NOMENCLATURE

A. Sets

$\mathcal{G}$	Set of all generation units g.
$\mathcal{B}$	Set of all Buses b.
$\mathcal{G}^b$	Set of all generation units at bus b.
$\mathcal{B} \times \mathcal{B}$	Set of all transmission lines $(\hat{l}, \check{l})$.
$\mathcal{X}(\boldsymbol{\xi})$	Feasible set with components $\boldsymbol{x} := (\boldsymbol{\sigma}; \boldsymbol{x}')$.

B. Indices

k, t	Time index.
k_0	Certain starting time.
κ	Decision epoch.

C. Random Parameters

$\Xi[k]$	Discrete-time stochastic process.
$\hat{\Xi}[k]$	Quantized stochastic process.
ξ	Realization of Ξ.

D. Scenario Tree/Library Components

$\mathcal{T}$	True Scenario tree.
$\mathcal{T}_{\text{opt}}$	Scenario tree of the optimal solution.
$\widetilde{\mathcal{T}}$	Finite approximation of $\mathcal{T}$.
$\widetilde{\mathcal{T}}_a$	Scenario tree with adapted root node.
$\mathfrak{T}^0$	Baseline library of scenario trees.
$\mathcal{V}$	Set of all nodes v on the tree.
$\mathcal{L} \subset \mathcal{V}$	Set of leaf nodes.
$(u, v) \in \mathcal{E}$	Set of edges.
$\mathfrak{F}$	Filtration structure.
$\mathcal{F}_{[k]}$	Sigma algebra structure.
$\mathbb{P}$	Filtered probability space.
$\prod_{t=0}^{T-1} \mathcal{P}_t$	Set of all paths on the tree with ρ as its elements.
$p_{u,v}$	Conditional probability.
v_-	Parent (direct predecessor) of node v.
v_+	Arbitrary child (direct successor) of node v.

E. Optimization Components

$\mathbb{E}[\cdot]$	Expectation function.
$C(\boldsymbol{x}, \boldsymbol{\xi})$	Convex cost function.
$\boldsymbol{\sigma}$	State component of decision array $\boldsymbol{x}$.
$\boldsymbol{x}'$	Non-state component of decision array $\boldsymbol{x}$.

1) Decision Variables

$x_v^g \in \mathbb{R}_+$	Schedule of unit g at node v.
$y_v^g \in \{0, 1\}$	Commitment of unit g at node v.
$\overline{s}_v^g \in \{0, 1\}$	Start-up of unit g at node v.
$\underline{s}_v^g \in \{0, 1\}$	Shut-down of unit g at node v.
$o_v^g \in \mathbb{R}_+$	Residual minimum on time of unit g at node v .
$d_v^g \in \mathbb{R}_+$	Residual minimum off time of unit g at node v.

2) Parameters

$\overline{S}^g$	Start-up cost of unit g.
$\underline{S}^g$	Shut-down cost of unit g.
c_0^g	Commitment cost of unit g.
c_1^g	Operation cost of unit g.
O_n^g	Minimum up time of unit g.
O_f^g	Minimum down time of unit g.

I. INTRODUCTION AND MOTIVATION

THE integration of more renewable generation in recent years has prompted significant interest in incorporating stochastic optimization tools into decision models for the electric power market. Perhaps, the most consequential of such decision models is the *Unit Commitment* (UC) problem [1], that schedules the optimal set of generating units to be activated to meet the electric net-load (load minus renewable generation) on the next hours or the next day. The net-load is unknown, but the UC schedule solves a deterministic problem using net-load forecasts. Therefore, uncertainty is managed by scheduling for reserves to be online and then dispatching them in real time. Uncertainty is either due to outages of generators and transmission

Manuscript received December 16, 2016; revised April 18, 2017, August 11, 2017, and October 3, 2017; accepted October 7, 2017. Date of publication November 2, 2017; date of current version June 18, 2018. This work was supported in part by NSF Award no. 1549923. Paper no. TPWRS-01881-2016.
(Corresponding author: Bita Analui.)

The authors are with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287-0112 USA (e-mail: bita.analui@asu.edu; anna.scaglione@asu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRS.2017.2768384

0885-8950 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

components or the systems' net-load realization. Even though our focus in this paper is on managing the latter, next we briefly review methodologies in the power system literature to model and manage both, because the tools and context are relevant. The *Security Constrained Unit Commitment* (SCUC) is the decision problem ensuring that the system can sail through any single generation or transmission component failure. Recently, several two-stage stochastic SCUC approaches have been proposed, that account for various sources of uncertainty in addition to contingencies. For instance, the authors in [2] address the problem by modeling load uncertainties. Instead, [3] and more recently in [4] and [5], the SCUC model takes into account intermittent wind power. However, since variations of renewable power are not rare events, it is more appropriate to directly commit the generating units for the base-load, not just the reserves, at the UC stage. Considering this, several alternative approaches were introduced to mitigate the aforementioned inefficiencies in the UC decision model.

One alternative is the so called *Chance Constrained Unit Commitment* (CCUC). Based on a confidence level, the CCUC goal is to avoid conservative solutions that would lead to large power spillage [6] by allowing the violation constraints that correspond to random events. An example of this approach is [7] where the authors propose a UC formulation in which wind uncertainty is not modeled explicitly through a probabilistic model, but empirically through a set of actual past scenarios. The bundle of scenarios are then used to produce a probability score for wind spillage and restrict it to the desired value through the decision variables. In addition, the authors in [8] developed an algorithm to address the computational challenge associated with chance-constrained two-stage UC model. The authors first adopt the bilinear reformulation to represent the probabilistic constraints and then convert that bilinear model into a linear one through the McCormick linearization method.

The approach that we discuss in this paper falls in a different class of decision models which capture the uncertainty directly, using a stochastic optimization framework (cf. [9], [10]). Stochastic optimization in general replaces deterministic objectives with an expectation on the cost (or disutility) of the solution and it explicitly verifies that there is a feasible solution in all foreseeable scenarios. In the context of the energy market this class of decision models is referred to as *Stochastic Unit Commitment* (SUC). The most commonly studied versions of SUC formulations are two-stage SUC problems.

The extensive existing literature in this category is differentiated primarily by the solution algorithm, the model and dimension of the underlying uncertainty and the inclusion of risk-averse objective functions. Since the second stage decisions are not directly linked within scenarios, if the problem is linear in the second stage, L-shaped algorithms or Benders Decomposition [11] are commonly used to solve two-stage SUC. In cases where the objective function is non-convex in the second stage decision variables, then the *convexification* of the second stage, or integer L-shaped methods, or a combination of those is required [12], [13]. More recently, [14] proposed a hybrid UC (HUC) formulation that allows finding solutions to the two-stage stochastic UC problem with a decreasing degree of conservatism by increasing the number of partitions. Also [15] proposes a tractable two-stage stochastic UC formulation that is a modification of the single stage formulation and simultaneously focuses on demand response and price elasticity.

The *Multistage* stochastic unit commitment (MSUC) is an extension of the two-stage SUC where the decisions are taken sequentially at certain period. While MSUC allows for smoother boundary conditions regarding the commitment variables (i.e., the *on* and *off* decisions), it has received less attention than two-stage problems because their complexity is often prohibitive. For solving MSUC problems, the authors in [16] pioneered the decomposition methods based on Lagrangian relaxation and sparked a series of follow up work focused on curbing the MSUC complexity (see e.g., [17]–[20] and the references therein). Specifically, the cutting plane algorithms proposed in [21] defines valid inequalities on a scenario tree and leads to more efficient and accurate solutions. As an alternative, the authors in [22] propose a heuristic approach that combines the progressive hedging algorithm with tabu search. Later on, [23] have shown that for non-convex cases, such as mixed-integer linear programs (MILP), progressive hedging (PH) is a very promising heuristic to parallelize the algorithm. In a similar vein, the authors in [24] proposed a formulation which improves the two-stage stochastic unit commitment problem by introducing a dynamic decision making approach to emulate the multistage formulation. The method is based on introducing dynamic constraints to force agreement for commitment decisions in all scenarios that correspond to similar wind levels, facilitating the enforcement of non-anticipativity.

To the best of our knowledge, all the work cited above focuses on a single instance of stochastic optimization, which could be applied to replace the deterministic day ahead UC in the wholesales market. The MSUC as a decision tool has the advantage of explicitly representing the underlying uncertainty and account for long term constraints, but that is typically limited to the horizon that the MSUC includes. The benefit of our approach in this context is incorporating multiple trading time frames in the power markets, so that fast switching units can be committed differently based on how the uncertainty unfolds.

According to a report by European Wind Energy Association [25], although the most important gate closure for trading in power market is the day-ahead of delivery, commitment plans may need to be updated for various reasons after this gate closure, e.g., suppliers may have an incorrect forecast of their production, an updated weather forecast might become available, or there may be an unforeseen downtime of a transmission line or an unscheduled outage of a generation unit. These adjustments can be made much more economically and efficiently in *intraday* markets. In some EU countries, intraday markets have been set up recently to fine-tune trading positions closer to real time delivery. It has been shown that intraday markets have positive impacts not only on generators but also on the operation of power systems. When allowing a generation unit leverage the more accurate and close to delivery information, real time balancing and therefore prices are reduced. It also facilitates the integration of renewable wind energy to the market too. Although intraday electricity markets are not yet established in the United States, the urge for it is sensed among the ISOs.

A. Contributions

Motivated by the potential benefits of allowing certain generation units to change their position in the gate closures of intraday markets, we propose a novel dynamic formulation for MSUC problem. Also, as new information is acquired, our algorithm includes a dynamic adjustment of the optimization parameters, in the form of library of scenario trees. Our contributions in this work are then two-fold:

First, we propose a Dynamic Multistage Stochastic Unit Commitment (DMSUC) model by introducing two additional state variables in MSUC to record and track the minimum-up/down time of the generation units. By introducing these variables, there is no requirement for carrying the conventional constraints along the decision horizon and yet, our solution maintains the continuity for those units whose operation is beyond the gate closures of intraday markets. In addition to the market structure, the number and duration of each intraday epoch, must also be balanced with the increased computational complexity and scalability of the underlying network. Independently, the authors in [26] very recently introduced an analogous idea to solve a deterministic formulation of the UC problem, noting that it is more amenable to LP-based solution methods and that it facilitates two consecutive epochs. We further note that in the case of the DMSUC, the use of these state variables has the additional benefit of facilitating the nodal formulation of the multi-stage stochastic program, since all constraints link nodal variables only to their corresponding parent, instead of an entire path along the scenario tree.

Second, in order to accurately represent the probability space structure for the DMSUC problem, we propose the construction of a *baseline* library of scenario trees that is also dynamically updated. More specifically, with the arrival of a full path, the library is updated to optimally include the observed history in the stochastic model.

In combining these two modules, our work enhances the MSUC framework by providing a sequential decision architecture for intraday markets that leverages the dynamic update

1) of the feasible action space due to the past decisions that cannot be reversed, and
2) of the optimization parameters, in the form of a dynamic library of scenario trees.

We explore the performance of our proposed method by first constructing the baseline library of scenario trees from actual net-load trajectories [27] and testing the DMSUC on the IEEE Reliability Test System (RTS) sample grid [28]. In our simulations we compare the proposed approach with solutions reproducing the state-of-the-art.

The remainder of this paper is organized in the following manner. In Section II, we first introduce the mathematical concepts to set the layout for formulating DMSUC problem and discuss its performance in Section II-B. In Section III, we discuss the issues arising when modeling the stochastic processes and present the algorithm to construct a library of scenario trees from net-load data. Our computational and ex-post performance results are presented in Section IV. Finally, Section V concludes the paper and states our future research.

II. Problem Formulation

A. Dynamic Multistage Stochastic Optimization

In the conventional Static Multistage Stochastic (SMS) optimization a decision maker minimizes the expected (constrained) cost of her actions under uncertainty. The expectation is taken averaging over all possible realizations ξ of a discrete time stochastic process $\Xi[k]$ over a future horizon including T time intervals that start at a certain time k_0. Without loss of generality, we can refer to the process $\Xi[k]$ for $k_0 \le k \le k_0 + T - 1$ as: $\{\Xi[k_0 + t] : t = 0, \dots, T-1\}$. The scenario tree $\mathcal{T} = \{\mathcal{V}, \mathcal{E}, \mathbb{P}; \boldsymbol{\xi}\}$ represents the basic structure that is used in SMS optimization problems to represent the gradual unfolding of information through *filtration* $\mathfrak{F} = \{\mathcal{F}_{[k_0]} \subset \mathcal{F}_{[k_0+1]} \subset \cdots \subset \mathcal{F}_{[k_0+T-1]}\}$ for the stochastic process $\{\Xi[k_0 + t] : t = 0, \dots, T-1\}$. For now we shall assume that this mapping is bijective. However, later we will see in details that for computational purposes this mapping is reduced to be surjective, such that the scenario tree is seen as finite-discrete approximation of the true filtration. The tree is a directed graph, i.e., the paths go from the root to the leaves through edges $(u, v) \in \mathcal{E}$ where u is connected to v, but not vice versa. Each node $v \in \mathcal{V}$ on the tree has a corresponding value $\xi_v \in \boldsymbol{\xi}$. The present, ξ_0, is unique (an event with probability *one*) and represents the root of the tree. The collection of values of an individual path along the tree represent a possible realization of $\{\Xi[k_0 + t] : t = 0, \dots, T-1\}$. Given any node v, the probability law $\mathbb{P}$ is specified by associating to each edge (u, v) a probability $p_{u,v}$ of the outcome $\Xi[k_0 + t] = \xi_v$ given the unique realization $\boldsymbol{\Xi}_{t-1} = \boldsymbol{\xi}_{0:u}$ that goes from the root to node u. Using the chain rule the probability of a certain sample path or *scenario* can be computed recursively as follows:

$$\pi_0 = \text{Prob}(\Xi[k_0] = \xi_0) = 1 \tag{1}$$

$$\pi_v = \text{Prob}(\boldsymbol{\Xi}_t = \boldsymbol{\xi}_{0:v}) = p_{u,v}\pi_u, \ (u, v) \in \mathcal{E}. \tag{2}$$

The general structure of the decision model becomes:

$$\min_{\boldsymbol{x}} \mathbb{E}[C(\boldsymbol{x}, \boldsymbol{\xi})] \text{ s.t. } \boldsymbol{x} \in \mathcal{X}(\boldsymbol{\xi}), \ \boldsymbol{\xi} \sim \mathcal{T} \tag{3}$$

where both the convex cost function $C(\boldsymbol{x}, \boldsymbol{\xi})$ and constraints set $\mathcal{X}(\boldsymbol{\xi})$ depend on the stochastic process outcomes. The notation $\boldsymbol{\xi} \sim \mathcal{T}$ refers to the fact that the optimization provides solutions for all paths on the scenario tree. The components of decision array $\boldsymbol{x}$ themselves can be associated with filtration described by tree with exactly the same structure $\mathcal{T}_{\text{opt}} = \{\mathcal{V}, \mathcal{E}, \mathbb{P}; \boldsymbol{x}\}$ but with the decision variables x_v in lieu of the random outcomes ξ_v associated to the nodes $v \in \mathcal{V}$. This property is known as *non-anticipativity* in stochastic optimization literature [29]. The conventional formulation looks at a single decision epoch, but normally decisions of this kind have to be taken repeatedly over consecutive epochs. This is certainly true in power market operations and motivates our quest for a dynamic formulation. First, we separate the components of the decision array into the *state* denoted by $\boldsymbol{\sigma}$, whose choice impacts the future feasible decisions, and those that do not, denoted by $\boldsymbol{x}'$, to partition vector $\boldsymbol{x} := (\boldsymbol{\sigma}; \boldsymbol{x}')$. Correspondingly, the variable associated to node

$v \in \mathcal{V}$ is denoted by $x_v = (\sigma_v; x'_v)$. The state components $\boldsymbol{\sigma}$ are the part of decisions which constrain and couple consecutive decision horizons. Also note that, a dynamic formulation typically reduces this coupling to only two consecutive epochs, hence the notion of *state*. Given all of the above, the dynamic formulation of an SMS problem is conceptually a straightforward extension, since it simply amounts to changing the time index k_0 (which is entirely superfluous in the SMS) into the current time k. Before we formally define the Dynamic Multistage Stochastic (DMS) optimization problem, let us establish the following conventions: **(i)** We will consider consecutive decision epochs of equal length T, each starting at present times k for the corresponding horizon κ with $k = \kappa T,\ \kappa \in \mathbb{Z}$. **(ii)** $\mathcal{T}[\kappa]$ indicates the scenario tree that represents the filtration of the incoming segment $\Xi[\kappa T : (\kappa+1)T - 1]$, conditioned on the past and present observations up to time k. Mathematically, the dynamic nature of the DMS is captured by explicitly indicating that all structures and variables defined above depend on κ. Also, how the set of constraints in decision epoch $\kappa + 1$ are affected by the actions and in particular the states of its prior epoch κ. Let $\boldsymbol{x}^{\text{opt}}[\kappa]$ be the optimum set of decisions that corresponds to the $\boldsymbol{\xi} \sim \mathcal{T}[\kappa]$. At time $(\kappa+1)T$ the decision maker will have observed the path that occurred in the previous epoch and, therefore, implemented a specific path of decisions out of $\boldsymbol{x}^{\text{opt}}[\kappa]$. Let this path correspond to the leaf that we denote as $v^* \in \mathcal{L}[\kappa]$, the decisions that were taken be $\boldsymbol{x}^{\text{opt}}_{0:v^*}[\kappa]$ and, in particular, the state at the leaf be $\sigma^{\text{opt}}_{v^*}[\kappa]$. Only this state $\sigma^{\text{opt}}_{v^*}[\kappa]$ carries the necessary information about the past decisions that need to be transferred in to the next horizon. Therefore, the optimization problem for epoch $\kappa + 1$ is the following extension of (3):

$$\min_{\boldsymbol{x}[\kappa+1]} \ \mathbb{E}[C_{\kappa+1}(\boldsymbol{x}[\kappa+1], \boldsymbol{\xi}[\kappa+1])] \tag{4}$$

$$\text{s.t. } \boldsymbol{x}[\kappa+1] \in \mathcal{X}\left(\boldsymbol{\xi}[\kappa+1], \sigma^{\text{opt}}_{v^*}[\kappa]\right), \tag{5}$$

$$\boldsymbol{\xi}[\kappa+1] \sim \mathcal{T}[\kappa+1] \tag{6}$$

In Sections III-D and IV we will explain in greater details how $\mathcal{T}[\kappa+1]$ is chosen and the state of the system from $\mathcal{T}_{\text{opt}}[\kappa]$ is transferred to it.

B. Dynamic Multistage Stochastic Unit Commitment

1) One epoch MSUC: To introduce the DMSUC formulation let us first consider a single epoch. Power systems are composed of several generating units $g \in \mathcal{G}$ at each bus $b \in \mathcal{B}$. All the buses are connected through a set of transmission lines $(\hat{l}, \check{l})$. The generation schedule corresponding to a certain node v on the scenario tree is the pair of continuous and binary variables $(x_v^g, y_v^g) \in \mathbb{R}_+ \times \mathbb{B}$. In addition to these typical schedule and commitment variables, two binary variables $\overline{s}_v^g$ and $\underline{s}_v^g$ indicate the switching action from *off* to on and from *on* to off respectively. The switching variables will take respectively value 0 unless the unit is turned on or off, in which case they will take respectively values 1. However, it is shown in subsequent sections that variables $\overline{s}_v^g$ and $\underline{s}_v^g$ are easily expressed in terms of binary variable y_v^g. Therefore, they can be relaxed to belong to $[0, 1]$.

In addition to these standard variables, we introduce two *state* variables to handle O_n^g and O_f^g, namely: o_v^g and d_v^g. The variable $o_v^g,\ v \in \mathcal{V}$, for v corresponding to t^{th} time instant in the horizon, is the residual time unit g needs to stay on after t, which depends on the parent state $o_{v_-}^g$; so, only when $o_v^g = 0$ the unit can be turned off and the state persists for the next generations as long as the machine continues to stay on or, if is switched off, for as long as it is off and not switched on again. The variable $d_v^g, v \in \mathcal{V}$, is the complementary variable handling the off time constraint, and represents the residual time the unit needs to remain off after time t depending on the state of the parent of node v_-. Similarly, only when $d_v^g = 0$ the optimization can turn the unit on and $d_v^g = 0$ persists after the unit is turn on until the next switch-off event. Note that this set of constraints are applied only to those units $g \in \mathcal{G}^s \subset \bigcup_{b \in \mathcal{B}} \mathcal{G}^b$ with $O_n^g > 1$ and $O_f^g > 1$. Below the full nodal formulation of MSUC problem is presented and in the following the corresponding constraints are described.

$$\min \sum_{v \in \mathcal{V}} \pi_v \sum_{g \in \mathcal{G}} \left[S_v^g + C_v^g + \frac{o_v^g}{O_n^g} + \frac{d_v^g}{O_f^g} \right] \tag{7a}$$

$$\text{w.r.t } (\mathbf{y}, \mathbf{x}, \mathbf{o}, \mathbf{d}, \overline{\mathbf{s}}, \underline{\mathbf{s}})$$

$$\text{s.t. } \boldsymbol{\xi} \sim \mathcal{T} \tag{7b}$$

$$\sum_{b \in \mathcal{B}} \left(\sum_{g \in \mathcal{G}^b} x_v^g - \xi_v^b \right) = 0 \qquad (\forall v \in \mathcal{V}) \tag{7c}$$

$$-L_{\hat{l}\check{l}} \le \sum_{b \in \mathcal{B}} D^b_{\hat{l}\check{l}} \left(\sum_{g \in \mathcal{G}^b} x_v^g - \xi_v^b \right) \le L_{\hat{l}\check{l}}, \forall \left(\hat{l}, \check{l}\right) \tag{7d}$$

$$o_v^g \ge \overline{s}_v^g (O_n^g - 1) \tag{7e}$$

$$\max\{0, o_{v_-}^g - y_v^g\} \le o_v^g \le o_{v_-}^g + \overline{s}_v^g (O_n^g - 1) \tag{7f}$$

$$o_{v_-}^g - o_v^g \le y_v^g \le 1 \quad (\forall v \in \mathcal{V} \setminus \{0\},\ g \in \mathcal{G}^s) \tag{7g}$$

$$d_v^g \ge \underline{s}_v^g \left(O_f^g - 1\right) \tag{7h}$$

$$\max\{0, d_{v_-}^g - 1 + y_v^g\} \le d_v^g \le d_{v_-}^g + \underline{s}_v^g \left(O_f^g - 1\right) \tag{7i}$$

$$0 \le y_v^g \le 1 - d_v^g + d_{v_-}^g \ (\forall v \in \mathcal{V} \setminus \{0\},\ g \in \mathcal{G}^s) \tag{7j}$$

$$y_v^g - y_{v_-}^g \le \overline{s}_v^g \tag{7k}$$

$$\underline{s}_v^g = y_{v_-}^g - y_v^g + \overline{s}_v^g \qquad (\forall v \in \mathcal{V} \setminus \{0\}) \tag{7l}$$

$$\underline{G}^g y_v^g \le x_v^g \le \overline{G}^g y_v^g \qquad (\forall v \in \mathcal{V}) \tag{7m}$$

$$-\underline{G}^{g\prime} y_v^g \le x_v^g - x_{v_-}^g \le \overline{G}^{g\prime} y_{v_-}^g \qquad (\forall v \in \mathcal{V} \setminus \{0\}) \tag{7n}$$

$$(y_{0_-}^g, x_{0_-}^g, o_{0_-}^g, d_{0_-}^g) = \text{Initial condition} \qquad (\forall\, g \in \mathcal{G}^s) \tag{7o}$$

$$\mathbf{y} \in \mathbb{B}^{|\mathcal{G}| \times |\mathcal{V}|},\ \mathbf{x}, \mathbf{o}, \mathbf{d} \in \mathbb{R}_+^{|\mathcal{G}| \times |\mathcal{V}|},\ \overline{\mathbf{s}}, \underline{\mathbf{s}} \in [0,1]^{|\mathcal{G}| \times |\mathcal{V}|}$$

In the formulation above, (7a) expresses objective i.e., to minimize the expected total cost. The actions of switching on and off the units come with the so-called start up $\overline{S}^g$ and shut down cost $\underline{S}^g$. In addition to these costs, there is a price bid that is a linear function of the production level x_v^g. Therefore the nodal cost term is expressed as:

$$S_v^g = \overline{S}^g \overline{s}_v^g + \underline{S}^g \underline{s}_v^g \tag{8}$$

$$C_v^g = c_1^g x_v^g + c_0^g y_v^g. \tag{9}$$

(7c) and (7d) constraints are added to account for the grid balance requirement under load uncertainty and line flow constraints for flow limits $[-L_{\hat{l}\check{l}}, L_{\hat{l}\check{l}}]$ for each transmission line $(\hat{l}, \check{l}) \in \mathcal{B} \times \mathcal{B}$ connecting bus $\hat{l}$ to bus $\check{l}$. Constraints (7e)–(7g) will enforce the behavior which was described in defining the o_v^g for all $v \in \mathcal{V}$ which have as parent node v_-. Constraint (7e) sets the state variable to the residual on-time when the unit is switched on. Constraint (7f) and the $\frac{o_v^g}{O_n^g}$ term in the objective, guarantee that the unit has a monotonically decreasing state when $1 \leq o_{v_-}^g \leq O_n^g - 1$. But once the unit reaches $o_{v_-}^g = 0$ it keeps the unit in that state until $\overline{s}_{v_+}^g = 1$ (v_+ denotes an arbitrary successor of node v or $v \prec v_+$). Note that, in this case the state goes from 0 to $O_n^g - 1$ so is not monotonically decreasing and that is why the upper-bound for o_v^g contains the term $\overline{s}_v^g(O_n^g - 1)$ which allows the transition to a higher state to happen. Constraint (7g) forces the unit to remain on (i.e., $y_v^g = 1$) as long as the state is monotonically decreasing by one, except when the parent $o_{v_-}^g = 0$, which means that the unit cannot be turned off until the minimum-up time is over. Note also that, when the unit is off $y_v^g = 0$, irrespective of what is going on with d_v^g, the state must be $o_v^g = 0$ and remains unchanged until the unit is switched on. The explanation of the constraints in (7h)–(7j) follow exactly the same line of reasoning. Constraints (7k)–(7l) express the switching variables in terms of the commitment variable. Moreover, (7m)–(7n) guarantee the production and ramping capacity limits respectively. It is also essential to address that the non-anticipativity constraint is already inherent in the nodal formulation. Next we discuss the role of the initial condition in (7o).

2) Dynamic Formulation of the MSUC: Recalling the discussion for the general dynamic multistage stochastic optimization problem in Section II-A, after one update in the tree $\mathcal{T}[\kappa]$, going from the static to the dynamic formulation amounts to finding how the past (optimal) decision variables constrain the decisions in the decision horizon corresponding to scenario tree $\mathcal{T}[\kappa+1]$. The partition for the decision variables of MSUC problem into state variables and the rest is as follows:

$$(\underbrace{y_v^g, x_v^g, o_v^g, d_v^g}_{\sigma_v}; \underbrace{\overline{s}_v^g, \underline{s}_v^g}_{x'_v}). \tag{10}$$

Recall (4)–(6) and as we stated in the previous section, let $v^* \in \mathcal{L}[\kappa]$ be the leaf node indicating the path that was observed during the κ^{th} horizon and just elapsed. In addition, let $(y_{v^*}^g[\kappa], x_{v^*}^g[\kappa], o_{v^*}^g[\kappa], d_{v^*}^g[\kappa])^{\text{opt}}$ be the quadruplet of optimum decisions that correspond to the leaf node v^*. To compute the optimal decisions corresponding to $\kappa+1$ epoch, with the dynamic formulation, the state value in (7o) is initialized with:

$$(y_{0_-}^g, x_{0_-}^g, o_{0_-}^g, d_{0_-}^g) = (y_{v^*}^g[\kappa], x_{v^*}^g[\kappa], o_{v^*}^g[\kappa], d_{v^*}^g[\kappa])^{\text{opt}}. \tag{11}$$

After the state of the system is transferred from decision epoch κ to $\kappa+1$, the MSUC problem (7a)–(7o) is solved for underlying tree $\mathcal{T}[\kappa+1]$. By finding solutions for all nodal variables $(\mathbf{y}[\kappa+1], \mathbf{x}[\kappa+1], \mathbf{o}[\kappa+1], \mathbf{d}[\kappa+1], \overline{\mathbf{s}}[\kappa+1], \underline{\mathbf{s}}[\kappa+1])$ the decision maker performs the action and awaits the information regarding the next epoch and the process continues for every incoming interval.

In the next section we complement our dynamic formulation with a dynamic construction of an approximate scenario tree.

III. Dynamic Construction of Load Scenario Tree

Typically, multistage stochastic optimization employs a finite and discrete approximation of the true stochastic process model, in the form of scenario tree. Before introducing our method, we next discuss the type of errors expected, which is a vital step to learn about the quality of approximate solution.

A. Modeling and Approximation Errors

In formulating the MSUC and its extension to DMSUC, we used notation *true* scenario tree $\mathcal{T}[\cdot]$ which includes the true probability model and filtration. In practice, however, capturing the true underlying model is unlikely for two reasons. First, in most cases the true probability model of the stochastic process is not available. Therefore, one can only rely on past observations to fit a parametric model or infer the empirical model in a non-parametric fashion. In both cases, *modeling* errors are consequential. Second, even when the true probability model is available, for computational purposes, the continuous process filtration must be replaced by a finite approximation of $\mathcal{T}[\cdot]$, denoted by $\widetilde{\mathcal{T}}[\cdot]$. The approximation, to the extent that is computationally tractable, will cause the *approximation* error in the scenario tree of decisions.

These aforementioned errors lead to sub-optimal solutions. It is shown in [30] that under specific continuity assumptions, the approximation error can be bounded scaling appropriately the nested distance between the true process and the approximate one. Unfortunately, with the inclusion of binary variables in the MSUC (and DMSUC) such results are not applicable. To the best of our knowledge, only the authors in [31] explored this issue. They proposed a general bounding technique based on solving independent sub-problems on scenario trees, which could be applied in mixed-integer MSO problems too. It can certainly be said that the bounds proposed in [30] would still be applicable if and only if the *optimal* integer variables remain unchanged in the surjective mapping between each path in $\mathcal{T}[\cdot]$ and the corresponding path in $\widetilde{\mathcal{T}}[\cdot]$. However, finding such a $\widetilde{\mathcal{T}}[\cdot]$ is an open problem. Therefore, in the following we use heuristics[1] to construct our approximation and test the efficacy of our approach numerically. We leave the theoretical study of bounds for the errors in mixed-integer formulations as future work.

[1]The same limitation appears in applying PH for stochastic mixed-integer optimization problems such that the solution quality relative to the optimal objective function value can not be provided [32].

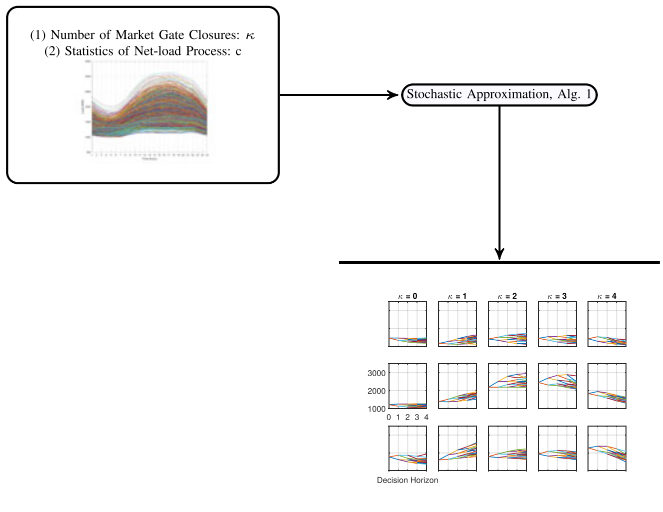

Fig. 1. Constructing baseline scenario tree library from historical net-load trajectories.

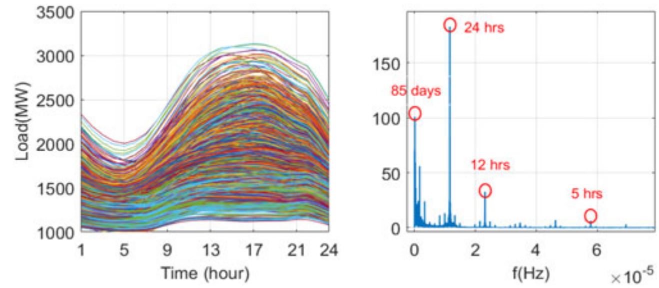

Fig. 2. Left: Hourly summer load. Right: Cyclic frequencies.

B. Scenario Tree Construction

The optimization of power generation schedules is inherently a stochastic problem and the underlying uncertainty is exogenous, meaning that the decision maker can not influence, by their decisions, the realizations of the stochastic process [33]. This property enables the decision maker to construct the scenario tree in advance and then solve the optimization problem, which is a scenario tree of actions. An important aspect of the DMSUC is that, by moving a look-ahead horizon to the next, the decision maker must use different statistics for the net-load, not only due to its past evolution and present value (i.e., the current state), but also because of the specific time of day. Furthermore, as new sample paths are observed, the model can be refined. This aspect motivated us to integrate in the DUMC a module that dynamically updates the filtration, based on the current state of the process, time of day and the sample path observed.

The approach we propose is to model the uncertainty through a library that contains net-load scenario trees, which are updated once the corresponding period elapses and a new scenario is observed. Fig. 1 depicts this procedure. These trees can be used as the underlying structures to DMSUC formulation. As explained in Section II-B2 each tree is selected to represent the uncertainty in the incoming epoch.

C. Exploiting Cyclostationarity Property of Load Data

A benefit of the proposed dynamic selection of the scenario tree is that the libraries used for different values of κ can be different, which means that the non-stationarity of the process can be directly captured. To illustrate the proposed procedure we use hourly measurements of PJM [27] net-load data, during summer months for years 2000–2015 *aggregated* from the different load zones within the PJM regional transmission organization. Therefore, the load scenario trees is constructed from a univariate stochastic process. However, due to spatial correlation of the load across the system, upon availability of separate measurement, a multivariate scenario tree represents the underlying probability space. The algorithm we describe for the scalar case has an identical counterpart in the multivariate case, if each scalar quantity is replaced with a vector. As an illustration, the original summer net-load trajectories are depicted in Fig. 2, for years 2000–2015. Obviously, the trajectories show cyclic features. It is also evident from Fig. 2, that load is not a stationary process and exhibits interdaily, daily and annual seasonalities. More formally, a cyclostationary process is defined as:

Definition 1: A discrete time stochastic process $\Xi[k]$ is cyclostationary in the strict sense with period P, if and only if for any set of indexes $(k_1, \ldots, k_n)$ the vector $(\Xi[k_1], \ldots, \Xi[k_n])$ and $(\Xi[k_1 + \ell P], \ldots, \Xi[k_n + \ell P])$ have the same joint distribution for any $\ell \in \mathbb{Z}$.

In the subsequent part of this section, we present the procedure for constructing a library of net-load scenario trees from empirical samples.

Let us denote the discrete time stochastic load process by Ξ with ξ be its realizations and the decision horizon for which the scenario tree is constructed by T. Even though the process exhibits multiple periods, we treat the hourly summer net-load as a cyclostationary stochastic process with period equal to one day ($P = 24$-hours) and divide the period into P/T consecutive different epochs. For the construction of the baseline tree library, the sample paths are selected from records of the same cycle for the corresponding decision horizon T. In addition, $\kappa \in \mathbb{Z}$ indexes the arbitrary consecutive decision epochs of length T. The choice of the duration T depends on the frequency of the arrival of new information, the structure based on which the market operates and can be chosen also based on the desired complexity of the MSUC to be solved. In addition to the choice of decision epochs κ, as was discussed in [34], it is also essential to account for all possible scenario trees that carry distinct information about the present. Since, it is indeed problematic if one relies only on the past data. We propose to deal with this issue through an approximation. Meaning that, prior to the scenario tree generation, we perform a *root* node Lloyd-Max clustering algorithm (cf. [35]). This step allows us to divide past scenario trajectories that emanate from each cluster and use them as representative of random scenarios that start from the same root. This way, we form a scenario tree library inwhich future scenarios are also distinguished based on the observed present value of the stochastic process. The optimization procedure is done in order to find the positions (or centroids) $\tilde{\xi}_0^l[\kappa T]$ for clusters $l = 1, \ldots, c$ and for each decision epoch κ. Each of

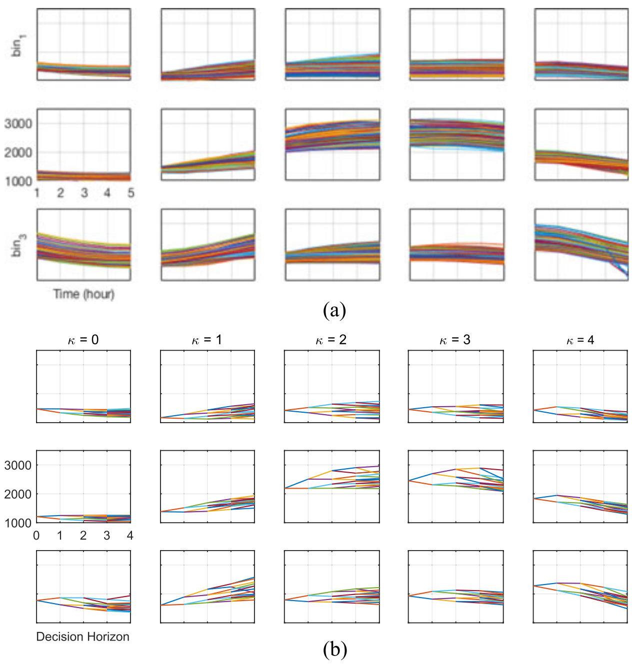

Fig. 3. (a) Net-load (MW) trajectories ξ^l_κ corresponding to quantization bins. (b) Net-load scenario trees corresponding to quantization bins. From hour 00:00 to 01:00AM+1, with $c = 3$ quantization bins and $T = 5$. (For all figures ylim = [1000, 3500] Megawatt). As evidenced by this figure, for a given bushiness of the tree the dynamic change of the library allows for a more accurate representation of the uncertainty ahead of the decision maker for a given present value and a given epoch κ.

these optimal solutions $\tilde{\xi}^l_0[\kappa T], l = 1, \ldots, c$ is a centroid associated with a *Voronoi* cell V^l_κ which contains particular (and not necessarily equal) number of trajectories. Having determined the root nodes and corresponding sample trajectories as shown in Fig. 3(a), next we will describe the scenario tree generation through stochastic approximation approach as was proposed in [36], modified to fit our dynamic formulation.

D. Library of Scenario Trees

Let the baseline library of scenario trees be $\mathfrak{T}^0 = \{\widetilde{T}^l[\kappa] : \kappa \in \mathbb{Z}$ with root $\tilde{\xi}^l_0[\kappa T], l = 1, \ldots, c\}$. Fig. 3(b), depicts the library $\mathfrak{T}^0$ for $c = 3$. Superscript 0 corresponds to the fact that library $\mathfrak{T}$ is constructed based on the historical data, however, it can be dynamically updated and evolve after the realization of a full path at the end of each epoch κ. We fix the graph of the tree $\{\mathcal{V}, \mathcal{E}\}$ to a *binary* structure (i.e., $\forall\ v \in \mathcal{V}, |v_+| = 2$) in order for the filtration to be represented uniformly along all consecutive decision epochs. The general part of the algorithm, which is a stochastic approximation iteration analogous to that in [36], is explained below. The construction of the library of scenario trees is briefed in Algorithm 1.

Let $\tilde{\boldsymbol{\xi}}^{(0)}$ be the initialization of the nodes based on the empirical marginal distributions (cf. [36, Algorithm 3-(i)]). Within the sample set of each root cluster a trajectory:

$$tr^{(n)}\left(\xi^l\right) = \left[\tilde{\xi}^l_0[\kappa T], \xi^l[\kappa T - \ell P + 1 : (\kappa+1)T - \ell P - 1]\right],$$

Algorithm 1: Constructing Library of Scenario Trees from Empirical Load Samples.

1: $\{\mathcal{V}, \mathcal{E}\}$, $\tilde{\boldsymbol{\xi}}^{(0)}$, c, κ, N, and $\mathfrak{T}^0 = \{\}$ are given.
2: $\forall\kappa$ and for $l = 1, \ldots, c$, centeroids $\tilde{\xi}^l_0[\kappa T]$ are computed.
3: repeat for κ
4: repeat for l
5: for $n = 1 : N$
6: randomly pick trajectory $tr^{(n)}(\xi^l)$ from corresponding bin
7: find ρ^* such that: $\tilde{\boldsymbol{\xi}}^{(n-1)}_{\rho^*} \in \operatorname{argmin} \mathbf{d}(tr^{(n)}(\xi^l), \tilde{\boldsymbol{\xi}}^{(n-1)}_\rho)$
8: update the path:
$$\tilde{\boldsymbol{\xi}}^{(n)}_{\rho^*} = \tilde{\boldsymbol{\xi}}^{(n-1)}_{\rho^*} - 2a_n \mathbf{d}(tr^{(n)}(\xi^l), \tilde{\boldsymbol{\xi}}^{(n-1)}_{\rho^*}) \nabla \mathbf{d}(tr^{(n)}(\xi^l), \tilde{\boldsymbol{\xi}}^{(n-1)}_{\rho^*})$$
9: update $\iota^{(n)}_{\rho^*}$
10: $n = n + 1$ and return to (5)
11: $\widetilde{T}^l[\kappa] = \tilde{\boldsymbol{\xi}}^{(N)}$, add new tree to the library: $\mathfrak{T}^0 = \mathfrak{T}^0 \cup \widetilde{T}^l[\kappa]$

is picked at random, with $n = 1, \ldots, N$ denoting the iteration number. By searching stage-wise through the paths on the tree, the candidate path for stochastic approximation step is found. We introduce set $\mathcal{P}_0 \times \mathcal{P}_1 \times \mathcal{P}_2 \times \ldots \times \mathcal{P}_{T-1}$ and it elements by ρ. Each vector ρ contains the node indices on the tree corresponding to each scenario path from root node to the leaves. In addition, a counting variable $\iota^{(0)}_\rho = \vec{0}$ is initialized to $zero$ and assigned to each scenario path ρ. We find $\rho^* \in \mathcal{P}_0 \times \mathcal{P}_1 \times \mathcal{P}_2 \times \ldots \times \mathcal{P}_{T-1}$ such that:

$$\tilde{\boldsymbol{\xi}}^{(n-1)}_{\rho^*} \in \operatorname{argmin} \mathbf{d}\left(tr^{(n)}\left(\xi^l\right), \tilde{\boldsymbol{\xi}}^{(n-1)}_\rho\right), \tag{12}$$

$\mathbf{d}(\cdot, \cdot)$ is an l_2-norm here. Consequently, path $\tilde{\boldsymbol{\xi}}^{(n-1)}_{\rho^*}$ is updated through the following gradient equation:

$$\tilde{\boldsymbol{\xi}}^{(n)}_{\rho^*} = \tilde{\boldsymbol{\xi}}^{(n-1)}_{\rho^*} - 2a_n \mathbf{d}\left(tr^{(n)}\left(\xi^l\right), \tilde{\boldsymbol{\xi}}^{(n-1)}_{\rho^*}\right) \nabla \mathbf{d}\left(tr^{(n)}\left(\xi^l\right), \tilde{\boldsymbol{\xi}}^{(n-1)}_{\rho^*}\right) \tag{13}$$

$$\iota^{(n)}_{\rho^*} = \iota^{(n-1)}_{\rho^*} + \mathbb{1}, \tag{14}$$

with a_n being the step size (cf. [36, Remark 10]). The iterations (12) and (13) are repeated for $n = 1, \ldots, N$ and until all the successor nodes on the tree are fixed. Finally, the conditional probabilities p_ρ are computed:

$$p_\rho = \frac{\iota^{(N)}_\rho}{N}, \ \forall \rho \in \ \mathcal{P}_0 \times \mathcal{P}_1 \times \mathcal{P}_2 \times \ldots \times \mathcal{P}_{T-1} \tag{15}$$

The following algorithm summarizes the procedure for generating the scenario tree library $\mathfrak{T}^0$.

Since tree library $\mathfrak{T}^0$ is constructed based on the historical net-load data, it can be viewed as a repository of trees. Obviously, considering that the scenario tree construction is purely data driven, updating the trees in the library with observation of a new complete path can bring cumulative benefits. This update will relocate the centroids in order to account for new realized path.

1) Here-and-Now Update: The root node on the scenario tree contains the information about the present event with probability *one*. Therefore, when the present value $\tilde{\xi}_{\mathrm{p}}[\kappa T]$ of the load for decision epoch κ is known, it is used to map library $\mathfrak{T}^0$ into the correct $\widetilde{\mathcal{T}}^l[\kappa]$ by finding index l^* such that:

$$\tilde{\xi}_0^{l^*}[\kappa T] \in \operatorname{argmin}_l \mathbf{d}\left(\tilde{\xi}_{\mathrm{p}}[\kappa T], \tilde{\xi}_0^l[\kappa T]\right) \tag{16}$$

When index l^* is found, tree $\widetilde{\mathcal{T}}^{l^*}[\kappa]$ is picked from the library and its root node is replaced with present value $\tilde{\xi}_{\mathrm{p}}[\kappa T]$. The corresponding tree is also denoted by the augmented tree $\widetilde{\mathcal{T}}_a^{l^*}[\kappa] := (\widetilde{\mathcal{T}}^{l^*}[\kappa]; \tilde{\xi}_{\mathrm{p}}[\kappa T])$.

2) Wait-and-See Update: After the decision horizon is completed, the actual net-load trajectory for decision epoch κ is fully realized. This full trajectory $tr^{(N+1)}$ is an additional information and can be used in iterative procedures (12) and (13) to update the augmented tree $\widetilde{\mathcal{T}}_a^l[\kappa]$.

The computational results presented in Section IV below, will demonstrate the stability of optimal solutions achieved based on our proposed approach and compare the performance of DMSUC with Rolling horizon and Static-Commitment MSUC approaches.

IV. Numerical Comparison

We perform our analysis on the IEEE RTS case, which is composed of 24 buses (specifically, the IEEE-24 test case), 33 generation units and 38 transmission lines [28]. Moreover, in order to test the performance, we use PJM net-load data from summer 2016 in all our numerical tests. The net-load data are rescaled to the same level of peak load in the IEEE 24 bus test case (cf. [28, Table 5]). In order to perform the benchmark analysis, in this section we compare the solution of our proposed approach with respect to the commitment and schedule process under different κ parameters and with the state-of-the-art. Namely, we will solve three variations of DMSUC:

a) DMSUC with $T = 5$ i.e., $\kappa = 0, \ldots, 4$,
b) DMSUC with $T = 8$ i.e., $\kappa = 0, \ldots, 2$,
c) Rolling Horizon (RH) for $\kappa = 0, \ldots, 23$.

For comparison, we also solved a single instance for a full $24h$ horizon of the deterministic UC and of the MSUC. Since the latter seeks a single commitment solution for all scenarios we refer to it as "Static-Commitment MSUC". The benchmarks in this analysis consists of:

d) Deterministic or day-ahead UC,
e) Static-Commitment MSUC.

Initially, we demonstrate the solution steps for DMSUC and then, we show how the actionable commitment scenarios and the generation production are implemented in real-time, by solving the Optimal Power Flow problem for the units that are committed for all the cases (a)-(e). The algorithms are all implemented in MATLAB using the YALMIP toolbox and the GUROBI 6.5.1 solver. The numerical experiments were all parallelized over 32 cores and conducted on an Intel Xeon 2.30GHz CPU with 256 GB available memory.

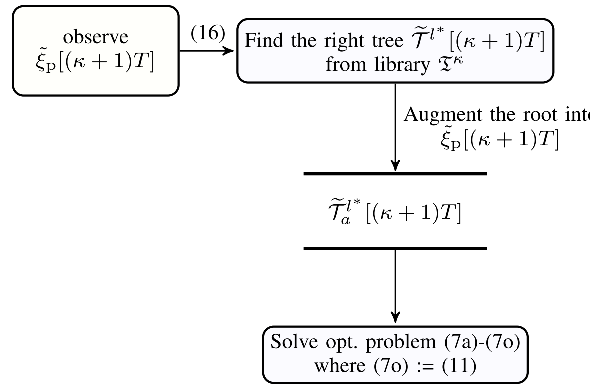

Fig. 4. Solving DMSUC for epoch $\kappa+1$. Note that baseline library $\mathfrak{T}^0$ is dynamically updated at the end of each epoch.

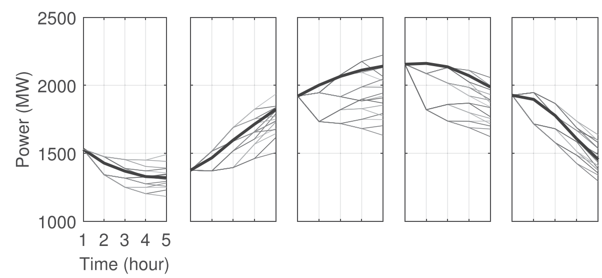

Fig. 5. Adopting the correct tree from $\mathfrak{T}^{0:4}$ after $\tilde{\xi}_{\mathrm{p}}[\kappa T]$ observed. Actual load (solid blue line) vs. optimal schedule scenarios in the augmented trees. One can clearly observe how the tree selected envelope the realization observed, confirming the importance of adapting the tree library with κ.

A. DMSUC's Optimal Solution

The diagram in Fig. 4, illustrates the required steps for solving DMSUC for epoch $\kappa+1$ showing how trees $\widetilde{\mathcal{T}}^l[\kappa]$ are chosen from library $\mathfrak{T}^0$ (Fig. 3(b)) as well as the *hear-and-now* update of the trees (solid blue line in Fig. 5) Based on (16), with $\kappa = 0, \ldots, 4$, $c = 3$, $T = 5$ and the values of actual presents $\tilde{\xi}_{\mathrm{p}}$, trees $\{\widetilde{\mathcal{T}}_a^1[0], \widetilde{\mathcal{T}}_a^2[1], \widetilde{\mathcal{T}}_a^3[2], \widetilde{\mathcal{T}}_a^3[3], \widetilde{\mathcal{T}}_a^2[4]\}$ are picked from the library and the corresponding root nodes are updated. Consequently, after solving optimization problem (7a)–(7o), Fig. 5. compares the net-load trajectory with the optimal schedule scenarios for the same decision horizons. From Fig. 5, it is clear that although the net-load trajectory is not exactly coinciding with a specific scenario, the optimal schedule scenario tree enfolds the trajectory almost everywhere and the discrepancy is compensable through ramping actions or procurement of reserves. Moreover, it is probable to observe where the actual load partially lies outside of the range of the optimal schedule scenarios as in stage 2 of decision epochs $\kappa = 2, 3$ in Fig. 5. In this case the discrepancy is managed through the deployment of reserves.

B. Comparison of Commitment and Schedule Decisions

In this section we compare the commitment and schedule process obtained from the approaches discussed in this paper for intraday markets, with the commitment and schedules of the day ahead deterministic UC and the Static Commitment MSUC. We generalize our proposed algorithm to the Rolling Horizon (RH)

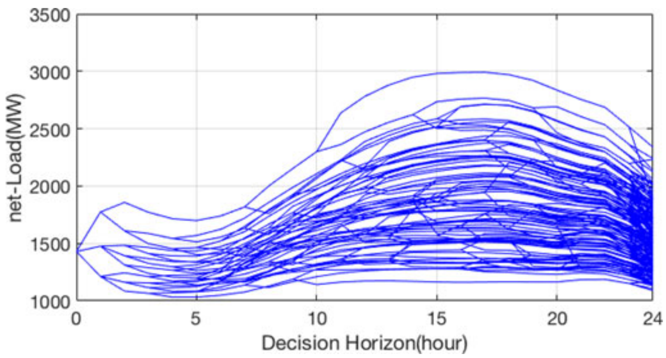

Fig. 6. Scenario Tree corresponding to the full day-ahead horizon.

implementation. The RH DMSUC allows for adjusting short-term commitment variables at the fasted possible pace (1h) while considering several future stages, offering the full adjustability for intraday gate closures in the market. By adjustability we mean that we incorporate information more frequently and re-shape our derisions accordingly. Although, we have exactly that same constraints as the conventional UC problem.

DMSUC and RH, provide the decision makers with a set of optimal actionable scenarios based on the underlying probability space a-head of time. However, based on the current practice and the infrastructure, market players may not yet agree to probabilistic contracts, therefore we compare our approach with the static commitment solutions and only optimize the schedule scenarios on the tree. This as shown here, it is certainly a more costly solution as it freezes the adjustability of the commitment on the fly. It is, however, practical in order to solve the decision process with regard to the existing contractual frameworks in the market. We employ this additional approach to instruct the generating units on their expected commitment positions. For this purpose, we construct a full day-ahead scenario tree using scenario reduction algorithm proposed by [37]. Fig. 6 illustrates this corresponding scenario tree with 1561 nodes and 225 scenarios and on this scenario tree we solve the Static-Commitment MSUC problem. Fig. 7, compares the actionable commitment processes and generation schedule for Deterministic UC, Static-Commitment, DMSUC and Rolling Horizon when solving the dynamic optimal power flow problem for all of our problem instances. On the left column, the figure shows the committed units (in black) in the scenario that is closest to the actual net-load trajectory. On the right column, the schedule process for the corresponding committed units are shown, and are obtained as the solution to the dynamic OPF. Evidently, with larger κ there are more commitment decisions that change to best adapt to new information. Larger values of κ also reduce the computational burden, due to the reduced number of stages necessary to cover the look-ahead horizon.

C. Ex-Post Performance

In this section we present the performance of DMSUC with respect to schedule scenarios ex-post. More specifically, we fix the commitment decisions and then solve a deterministic version of the optimization to meet the actual load demand

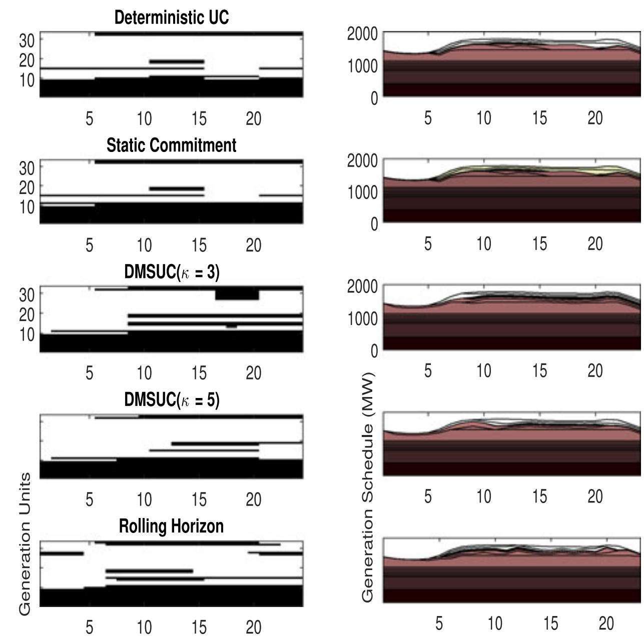

Fig. 7. Comparing the actionable scenarios for deterministic UC, static-commitment process, DMSUC with $\kappa = 0, 1, 2$, DMSUC with $\kappa = 0, 1, \ldots, 4$ and rolling horizon.

that happens in real-time[2]. We call the *ex-post* objective the following quantity:

Definition 2: The ex-post objective $C_e[\kappa]$ for each epoch κ is defined as:

$$C_e[\kappa] = \sum_{v \in \rho^+} \sum_{g \in \mathcal{G}} \left[S_v^g + C_v^g + \frac{o_v^g}{O_n^g} + \frac{d_v^g}{O_f^g} \right]$$

with ρ^+ indicating $\tilde{\xi}_{\rho^+} \in \widetilde{\mathcal{T}}_a^l[\kappa]$, which is the closest scenario to the actual net-load trajectory.

The results for the average $C_e[\kappa]$ are shown in Fig. 8, for all approaches (a)-(e) demonstrated that by solving for the commitment variables in all 84 problem instances for the Deterministic UC, Static-Commitment, DMSUC and RH, with increasing adjustability of the commitment decisions, the sample average of the ex-post objective $C_e[\kappa]$ also decreases.

Moreover, we have seen that the DMSUC in (7a)–(7o) is designed to provide optimal scenarios for schedule variables in all decision epochs κ. Fig. 9 demonstrates analysis of DMSUC optimal objective, ex-post for all problem instances corresponding to summer 2016 net-load. On the left, the empirical cumulative distribution function (cdf) of the optimal objective is plotted for all epochs $\kappa = 0, 1, \ldots, 4$ and clearly reveals the daily load pattern. On the right, the average ex-post objective $C_e[\cdot]$ across all problem instances is plotted and compared to the sample average of DMSUC optimal objective for $\kappa = 0, 1, \ldots, 4$. Alignment of

[2]Of course, in the OPF objective we solve the startup and shutdown cost play no role, as the integer variables are all fixed. Typically, in real-time markets one schedules adjustments relative to the day-ahead schedule, not the entire amount of power needed. Since it is unclear what to choose as the schedule cleared in the day-ahead market, rather than making assumptions here we are simply trying to highlight the way the stochastic formulation provides long terms guarantees on the cost averaged over-time.

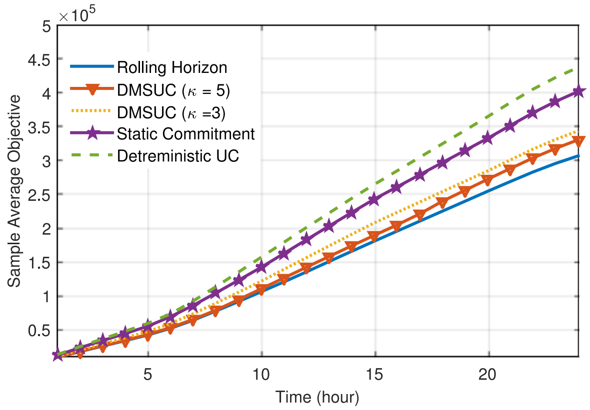

Fig. 8. Comparing the sample average generation cost for commitment decisions in RH, DMSUC, static-commitment and deterministic UC.

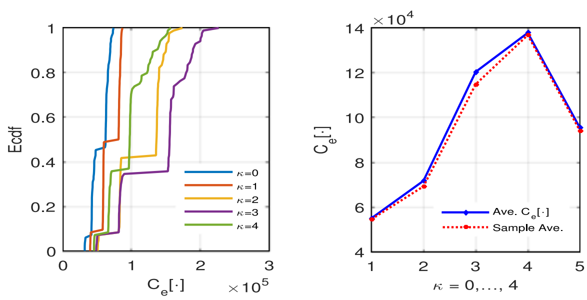

Fig. 9. Left: Empirical cdf of DMSUC. Right: Ave. $C_e[\cdot]$ and Sample Ave. of DMSUC for $\kappa = 0, \ldots, 4$.

these two measures well promotes the virtues of stochastic optimization in general, and of our proposed approach in particular, since it does lead to the minimum long term average cost.

Note that $C_e[\cdot]$ is not the cost of adjustment actions for procurement of reserves, but it is expected that the comparison of these real-time adjustment costs would be favorable to the DMSUC because of its ability to give a long term average objective that match the expected values and because its expected objective is, by design, lower then the ones for all other methods.

V. Conclusions

In this paper we presented a new dynamic formulation for the MSUC problem, the DMSUC. The DMSUC updates the feasible decision space and stochastic model to schedule units with minimum expected cost on the current look-ahead horizon. In our formulation, the decision horizon of a typical day-ahead UC problem is easily adapted to the structure of intraday markets. While accounting for the explicit representation of uncertainty, this approach uses the generation resources more flexibly. It also allows to solve a chain of smaller problem instances in a dynamic way, with better performance compared to a static day-ahead MSUC. Compared to [21] our judicious use of state variables allows the decision maker to solve smaller MSUC problems while keeping the continuity of commitment decisions along the full horizon (e.g., 24 hours) intact. Therefore, the minimum-up and minimum-down times of the generators do not have to be within the decision horizon. In addition, when the subset of commitment variables covering the baseline load are fixed at $\kappa = 0$, the size of the optimization problem in further epochs is significantly reduced and only involves updating their states. We also described the algorithm to construct a dictionary of scenario trees by exploiting the cyclostationary property of net-load process and to update the scenario tree library, as new sample paths are fully observed. Finally, we presented and discussed the performance of DMSUC compared to a deterministic UC (emulating the current day ahead practice) and a Static-Commitment day-ahead MSUC problem. Overall, the DMSUC approach outperforms the others, because of its ability to adapt the decision in light of new information. In addition, the proximity of ex-post objective and the sample average of the optimal objective showed yet another strong benefit for utilizing multistage stochastic optimization in market environments. The benefit of performing a DMSUC in intraday markets is that it allows, to an extent, to co-optimize the commitment schedule and real time balancing. It also enables the market players to change their positions dynamically based on the arrival of new information, increasing economic competition. Also, while it is true that one needs to solve many optimization problems, it is evident from our numerical results that a relatively small look-ahead is sufficient to provide schedules that are more efficient and stable compared to the state-of-the-art.

References

[1] S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis, "A genetic algorithm solution to the unit commitment problem," *IEEE Trans. Power Syst.*, vol. 11, no. 1, pp. 83–92, Feb. 1996.

[2] L. Wu and M. Shahidehpour, "Stochastic security-constrained unit commitment," *IEEE Trans. Power Syst.*, vol. 22, no. 2, pp. 800–811, May 2007.

[3] J. Wang, M. Shahidehpour, and Z. Li, "Security-constrained unit commitment with volatile wind power generation," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1319–1327, Aug. 2008.

[4] D. A. Tejada-Arango, P. Sanchez-Martin, and A. Ramos, "Security constrained unit commitment using line outage distribution factors," *IEEE Trans. Power Syst.*, 2017.

[5] M. Shahidehpour, C. Shao, X. Wang, B. Wang, and X. Wang, "Security-constrained unit commitment with flexible set for variable wind power," *IEEE Trans. Sustain. Energy*, vol. 8, no. 3, pp. 1237–1246, Jul. 2017.

[6] F. Bouffard and F. Galiana, "Stochastic security for operations planning with significant wind power generation," *IEEE Trans. Power Syst.*, vol. 23, no. 2, pp. 306–316, May 2008.

[7] C. Zhao, Q. Wang, J. Wang, and Y. Guan, "Expected value and chance constrained stochastic unit commitment ensuring wind power utilization," *IEEE Trans. Power Syst.*, vol. 29, no. 6, pp. 2696–2705, Nov. 2014.

[8] Y. Zhang, J. Wang, B. Zeng, and Z. Hu, "Chance-constrained two-stage unit commitment under uncertain load and wind power output using bilinear benders decomposition," *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 3637–3647, Sep. 2017.

[9] G. C. Pflug and A. Pichler, *Multistage Stochastic Optimization*. New York, NY, USA: Springer-Verlag, 2014.

[10] A. Ruszczynski and A. Shapiro, *Stochastic Programming*, 1st ed. (Handbooks in Operations Research and Management Science). Amsterdam, The Netherlands: Elsevier, 2003.

[11] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, P. Glynn and S. Robinson, Eds. New York, NY, USA: Springer-Verlag, 1997.

[12] Q. P. Zheng, J. Wang, P. M. Pardalos, and Y. Guan, "A decomposition approach to the two-stage stochastic unit commitment problem," *Ann. Oper. Res.*, vol. 210, no. 1, pp. 387–410, 2013.

[13] R. Jabr, "Tight polyhedral approximation for mixed-integer linear programming unit commitment formulations," *IET Gener., Transmiss. Distrib.*, vol. 6, no. 11, pp. 1104–1111, 2012.

[14] I. Blanco and J. M. Morales, "An efficient robust solution to the two-stage stochastic unit commitment problem," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4477–4488, Nov. 2017.
[15] V. K. Tumuluru and D. H. Tsang, "A two-stage approach for network constrained unit commitment problem with demand response," *IEEE Trans. Smart Grid*, 2016.
[16] S. Takriti, J. R. Birge, and E. Long, "A stochastic model for unit commitment problem," *IEEE Trans. Power Syst.*, vol. 11, no. 3, pp. 1497–1508, Aug. 1996.
[17] P. Carpentier, G. Cohen, J.-C. Culioli, and A. Renaud, "Stochastic optimization of unit commitment: A new decomposition framework," *IEEE Trans. Power Syst.*, vol. 11, no. 2, pp. 1067–1073, May 1996.
[18] M. Nowak and W. Römisch, "Stochastic Lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty," *Ann. Oper. Res.*, vol. 100, pp. 251–272, 2000.
[19] T. Shiina and J. R. Birge, "Stochastic unit commitment problem," *Int. Trans. Oper. Res.*, vol. 11, pp. 19–32, 2004.
[20] A. Papavasiliou, S. Oren, and R. O'Neill, "Reserve requirements for wind power integration: A scenario-based stochastic programming framework," *IEEE Trans. Power Syst.*, vol. 26, no. 4, pp. 2197–2206, Nov. 2011.
[21] R. Jiang, Y. Guan, and J.-P. Watson, "Cutting planes for multistage stochastic unit commitment problem," *Math. Program.*, vol. 157, no. 1, pp. 121–151, 2016.
[22] A. Løkketangen and D. L. Woodruff, "Progressive hedging and tabu search applied to mixed integer (0, 1) multistage stochastic programming," *J. Heuristics*, vol. 2, no. 2, pp. 111–128, 1996.
[23] S. M. Ryan, R. J.-B. Wets, D. L. Woodruff, C. Silva-Monroy, and J.-P. Watson, "Toward scalable, parallel progressive hedging for stochastic unit commitment," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2013, pp. 1–5.
[24] C. Uçkun, A. Botterud, and J. R. Birge, "An improved stochastic unit commitment formulation to accommodate wind uncertainty," *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 2507–2517, Jul. 2016.
[25] "Creating the internal energy market in Europe," Sep. 2012. [Online]. Available: http://www.ewea.org/uploads/tx_err/Internal_energy_market.pdf. Accessed: Feb. 2017.
[26] S. Atakan, G. Lulli, and S. Sen, "A state transition MIP formulation for the unit commitment problem," *IEEE Trans. Power Syst.*, 2017.
[27] "Metered load data." [Online]. Available: http://www.pjm.com/markets-and-operations/ops-analysis/historical-load- data.aspx, 2000–2015. Accessed: Aug. 2016.
[28] P. F. Albrecht, "IEEE Reliability Test System," *IEEE Trans. Power App. Syst.*, vol. PAS-98, no. 6, pp. 2047–2054, Nov. 1979.
[29] R. T. Rockafellar and R. Wets, "Nonanticipativity and L^1-martingales in stochastic optimization problems," in *Stochastic Systems: Modeling, Identification and Optimization, II*, 1976, pp. 170–187.
[30] G. C. Pflug and A. Pichler, "A distance for multistage stochastic optimization models," *SIAM J. Optim.*, vol. 22, no. 1, pp. 1–23, 2012.
[31] F. Maggioni, E. Allevi, and M. Bertocchi, "Monotonic bounds in multistage mixed-integer linear stochastic programming: Theoretical and numerical results," *Comput. Manage. Sci.*, vol. 13, pp. 423–457, 2016, doi: 10.1007/s10287-016-0254-5.
[32] D. Gade, G. Hackebeil, S. M. Ryan, J.-P. Watson, R. J.-B. Wets, and D. L. Woodruff, "Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs," *Math. Program.*, vol. 157, no. 1, pp. 47–67, 2016.
[33] B. Defourny, D. Ernst, and L. Wehenkel, "Multistage stochastic programming: A scenario tree based approach," *Decision Theory Models for Applications in Artificial Intelligence: Concepts and Solutions*. Hershey, PA, USA, IGI Global, 2011, pp. 97–147.
[34] B. Analui and A. Scaglione, "An approximation algorithm for future wind scenarios," in *Proc. IEEE Statist. Signal Process. Workshop*, 2016, pp. 1–5, doi: 10.1109/SSP.2016.7551821.
[35] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.
[36] G. C. Pflug and A. Pichler, "Dynamic generation of scenario trees," *Comput. Optim. Appl.*, vol. 62, no. 3, pp. 641–668, 2015.
[37] H. Heitsch and W. Römisch, "Scenario tree generation for multistage stochastic programs," *Comput. Manage. Sci.*, vol. 6, pp. 117–133, 2009.

Bita Analui received the Ph.D. in statistics and operations research from the University of Vienna, Vienna, Austria, in 2014. She was a postdoctoral scholar in Electrical, Computer and Energy Engineering School and Postdoctoral in Computing Best Practice Fellow with Arizona State University. Her expertises include mathematical and statistical modeling with focus on dynamic decision making under uncertainty and theory of stochastic optimization. She has conducted research on novel algorithms for power systems planning and operation with renewable integration.

Anna Scaglione (F'11) is a Professor with the Department of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA. Her expertise include broad area of signal processing for communication and power systems. She is the recipient of the 2013 IEEE Donald G. Fink Prize Paper Award, the 2013 SPS Young Author Best Paper Award (with her student), the Ellersick Best Paper Award (MILCOM 2005), the NSF Career Award in 2002, and the 2000 IEEE SIGNAL PROCESSING TRANSACTIONS Best Paper Award.