An Efficient Robust Solution to the Two-Stage Stochastic Unit Commitment Problem

Blanco, Ignacio; Morales González, Juan Miguel

Published in: IEEE Transactions on Power Systems

Link to article, DOI: 10.1109/TPWRS.2017.2683263

Publication date: 2017

Document Version: Peer reviewed version


General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
Abstract—This paper provides a reformulation of the scenario-based two-stage unit commitment problem under uncertainty that allows finding unit-commitment plans that perform reasonably well both in expectation and for the worst case. The proposed reformulation is based on partitioning the sample space of the uncertain factors by clustering the scenarios that approximate their probability distributions. The degree of conservatism of the resulting unit-commitment plan (that is, how close it is to the one provided by a purely robust or stochastic unit-commitment formulation) is controlled by the number of partitions into which the said sample space is split. To efficiently solve the proposed reformulation of the unit-commitment problem under uncertainty, we develop two alternative parallelization and decomposition schemes that rely on a column-and-constraint generation procedure. Finally, we analyze the quality of the solutions provided by this reformulation for a case study based on the IEEE 14-node power system and test the effectiveness of the proposed parallelization and decomposition solution approaches on the larger IEEE 3-Area RTS-96 power system.

Keywords—Stochastic and robust unit commitment, column-and-constraint generation, parallel computing, clustering, scenario reduction.

NOMENCLATURE

The notation used throughout the paper is stated below for quick reference. Other symbols are defined as required.

A. Indexes and Sets

<table>
<thead>
<tr>
<th>Set</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>Set of time periods ( t ).</td>
</tr>
<tr>
<td>( N )</td>
<td>Set of nodes ( n ).</td>
</tr>
<tr>
<td>( G )</td>
<td>Set of conventional generation units ( g ).</td>
</tr>
<tr>
<td>( F )</td>
<td>Set of stochastic power production units ( f ).</td>
</tr>
<tr>
<td>( L )</td>
<td>Set of loads ( l ).</td>
</tr>
<tr>
<td>( \Omega )</td>
<td>Set of scenarios ( \omega ), ranging from 1 to ( \lambda ).</td>
</tr>
<tr>
<td>( P )</td>
<td>Set of partitions ( p ), ranging from 1 to ( k ).</td>
</tr>
<tr>
<td>( \Omega_p )</td>
<td>Set of scenarios ( \omega ) in partition ( p ).</td>
</tr>
<tr>
<td>( F_n )</td>
<td>Set of stochastic power production units located at node ( n ).</td>
</tr>
<tr>
<td>( G_n )</td>
<td>Set of conventional generation units located at node ( n ).</td>
</tr>
<tr>
<td>( L_n )</td>
<td>Set of loads connected at node ( n ).</td>
</tr>
<tr>
<td>( M_n )</td>
<td>Set of nodes ( m \in N ) that are connected to node ( n ) by a transmission line.</td>
</tr>
<tr>
<td>( \Omega_p' )</td>
<td>Reduced set of scenarios ( \omega ) in partition ( p ).</td>
</tr>
</tbody>
</table>

B. Parameters

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{g}^{F} )</td>
<td>Fixed/variable production cost of conventional generation unit ( g ).</td>
</tr>
<tr>
<td>( C_{g}^{V} )</td>
<td>Start-up/Shut-down cost of conventional generation unit ( g ).</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>Number of time periods.</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>Number of scenarios.</td>
</tr>
<tr>
<td>( M )</td>
<td>Number of loads.</td>
</tr>
<tr>
<td>( P )</td>
<td>Number of partitions.</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>Number of conventional generation units.</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>Number of nodes.</td>
</tr>
<tr>
<td>( \Omega )</td>
<td>Number of scenarios in each partition.</td>
</tr>
<tr>
<td>( \rho )</td>
<td>Weight associated with partition ( p ).</td>
</tr>
</tbody>
</table>

A. Parameter Definitions

- \( L_{l,t} \): Demand for load \( l \) at time \( t \).
- \( R U_{g}, R D_{g} \): Ramp-up/Ramp-down rate for conventional generation unit \( g \).
- \( U T_{g}, D T_{g} \): Minimum-up/Minimum-down time for unit \( g \).
- \( L_{UP_{f}, L_{DW_{f}}} \): Number of time periods conventional generation unit \( g \) must be online/offline counting from \( t = 1 \).
- \( IS_{g} \): Initial status of unit \( g \), equal to 1 if online at \( t = 0 \) and 0, otherwise.
- \( ON_{g}, OFF_{g} \): Number of time periods unit \( g \) has been online/offline prior to \( t = 1 \).
- \( X_{n,m} \): Reactance of line \( n - m \).
- \( F_{n,m} \): Maximum flow capacity of line \( n - m \).
- \( p_{f}^{max}, p_{min} \): Maximum/minimum power production of conventional generation unit \( g \).
- \( P_{SU_{g}}, P_{SD_{g}} \): Maximum starting-up/shutting-down power production of conventional generation unit \( g \).
- \( P_{IS} \): Power output of conventional unit \( g \) at \( t = 0 \).
- \( C_{L} \): Cost of involuntary load curtailment.
- \( W_{f,t,\omega} \): Power production from stochastic generation unit \( f \) at time \( t \) in scenario \( \omega \).
- \( \pi_{\omega} \): Probability of scenario \( \omega \).
- \( \rho_{p} \): Weight associated with partition \( p \). |

C. First-stage variables

- \( u_{g,t} \): Binary variable equal to 1 if unit \( g \) is online at time \( t \) and 0, otherwise.
- \( y_{g,t}/z_{g,t} \): Binary variable equal to 1 if unit \( g \) is starting up/shutting down at time \( t \) and 0, otherwise.

D. Second-stage variables

- \( P_{g,t,\omega} \): Power produced by conventional generation unit \( g \) in scenario \( \omega \) at time \( t \).
- \( L_{SH_{l,t,\omega}} \): Power curtailment from load \( l \) in scenario \( \omega \) at time \( t \).
- \( W_{f,t,\omega}^{SP} \): Power curtailment from stochastic power production unit \( f \) in scenario \( \omega \) at time \( t \).
- \( \delta_{n,t,\omega} \): Voltage angle at node \( n \), time \( t \) and scenario \( \omega \).
- \( \alpha \): Auxiliary variable used in the scenario-based robust unit commitment formulation.
- \( \theta_{p} \): Auxiliary variable used in the hybrid unit commitment formulation.

I. INTRODUCTION

The increasing reliance on partly unpredictable renewable power supply has prompted the revision of the procedures used for power system operations. This is the case, for example, of the tool used by system operators to decide the commitment of power plants, that is, to solve the so-called unit commitment problem (UC). Two-stage stochastic
programming [1] and robust optimization [2] have become the most popular and explored techniques of optimization under uncertainty to improve unit-commitment decisions in terms of both cost-efficiency and system reliability.

The formulation and solution of the unit commitment problem using either stochastic programming or robust optimization—the result of which is typically referred to as \textit{stochastic} and \textit{robust unit commitment}, respectively—has been subject of numerous studies by the scientific community; see, for instance, [3]–[11], among many others and variants.

Essentially, the stochastic unit commitment problem (SUC) makes use of a probabilistic model for the uncertain input factors such as demand, equipment failures and partly-predictable renewable power production to minimize a certain quantile of the induced system cost distribution or its expectation. Most often than not, this probabilistic model is approximated by a set of scenarios that describe plausible realizations of such random factors. In order for the stochastic solution to be reliable, the amount of scenarios that need to be considered must be large, which may render an intractable optimization problem, or carefully generated, which motivates the topic of \textit{scenario reduction techniques} [12]–[14]. Furthermore, the probabilistic model from which these scenarios may be drawn may carry, in itself, some level of uncertainty as well.

In contrast, the robust unit commitment problem (RUC) seeks a commitment plan that allows the system to withstand the worst-case realization of the uncertain factors at a minimum cost. While this approach saves the decision-maker from having to probabilistically characterize these factors, it may yield too conservative solutions as the worst-case scenario rarely occurs.

In recent years, several methods have been proposed to make decisions under uncertainty that perform \textit{relatively well} under the premises of both the stochastic and the robust approaches, that is, in expectation and for the worst case. Illustrative examples of these methods can be found in [15]–[18], where hybrid stochastic-robust solution strategies are developed for optimal air-quality and municipal solid-waste management, electricity trading for power microgrids and energy contracting for a portfolio of renewable power generation technologies, respectively. What makes all these solution strategies \textit{hybrid} is that some of the uncertain parameters are assumed to follow certain probability distributions, while others are solely known to belong to some uncertainty sets.

Within the context of the unit commitment problem, we highlight the work in [19] and [20]. More specifically, the authors in [19] propose a mathematical formulation that delivers the unit-commitment plan that minimizes a user-controlled weighted sum of the expected and the worst-case costs. The solution approach introduced in [20], even if presented as a method to tackle the stochastic unit commitment problem, seeks to determine a unit-commitment plan that is robust against an \textit{ambiguous} probability distribution of renewable energy generation, an ambiguity that is the result of the always limited availability of data and that is modeled in practice as a vector of imperfectly known probabilities. Thus, as the amount of historical data increases, the ambiguity of such a probability distribution diminishes and so does the need for robustness and the degree of conservatism of the stochastic unit-commitment solution. This approach can also be regarded as a form of \textit{distributionally robust optimization} [21].

Our work shares with [19] and [20] the aim of finding a solution to the stochastic unit commitment problem that is robust in some sense, but our motivation and the methodology we propose to this end are essentially different. We assume that the probability distributions of the uncertain parameters—in our case, the wind power production—are known, but that, as it normally occurs in practice, computational tractability only allows us to solve the stochastic unit commitment problem for a scenario-based approximation of such distributions. In principle, we shall consider a large number of scenarios for this approximation to be accurate enough. In any case, we group these scenarios using a clustering technique—for instance, the \textit{k-means clustering algorithm} [24], which has been reported to feature good performance in similar contexts [25], [26]—. Each of the so-obtained clusters is referred to as a \textit{partition}. We then formulate and solve a two-stage unit-commitment problem that minimizes the expected value of the system operating costs, where the expectation is taken over the collection of worst-case scenarios within each partition. The probability assigned to each of these worst-case scenarios is equal to the probability of the partition it belongs to, which is, in turn, computed by summing up the probabilities of the scenarios that form part of the partition in question.

For convenience, we employ the term \textit{hybrid unit commitment problem} and the acronyn HUC to refer to the proposed reformulation of the UC problem. This reformulation brings two major advantages, namely:

1) It allows finding solutions to the two-stage unit commitment problem with a decreasing degree of conservatism by increasing the number of partitions. In fact, if only one partition is considered, the HUC delivers the (scenario-based) robust unit-commitment solution (where by “scenario-based”, we mean the solution given by the robust formulation of the two-stage unit-commitment problem in the particular case that the uncertainty is modeled as a finite set of atoms, outcomes or scenarios). In contrast, if the number of partitions is made to coincide with the number of scenarios, the HUC solution boils down to the stochastic unit-commitment plan. Furthermore, as we show later, the computational time required to find the HUC solution increases with the number of partitions considered, with the solution algorithm being very fast for a small number of them. This provides a practical way to adjust the level of conservatism of the unit-commitment solution depending on time availability.

2) It is amenable to decomposition and parallelization in various ways and levels and, hence, it can be efficiently solved. Indeed, based on the column-and-constraint generation procedure described in [9], we provide and compare two alternative decomposition schemes to solve the HUC problem. These two schemes basically differ in whether the worst-case scenarios within each partition are identified independently for each partition or not.

The remainder of this paper is organized as follows. Section II begins by providing mathematical formulations for the scenario-based two-stage stochastic and robust unit commitment problems, in that order, and finishes with the formulation of the proposed hybrid unit commitment problem. Furthermore, in this section we explain how we use a clustering method to construct the partitions in the HUC model and how these can be employed to control the degree of conservatism of the resulting unit-commitment plan. Section III introduces the proposed parallelization and decomposition strategies to solve the HUC problem. Section
IV analyzes and discusses results from two case studies based on standard IEEE power systems. Finally, in Section V the main conclusions of our work are summarized, including possible avenues for future research.

II. MATHEMATICAL FORMULATION

In the two-stage unit commitment problem under uncertainty, decision variables are divided into two groups. The first group constitutes the commitment plan itself and consists of the 0/1 variables $u_{g,t}$, $y_{g,t}$, $z_{g,t}$, which determine the on/off status, the start-up, and the shutdown of generating unit $g$ in time period $t$, respectively. These decisions are to be made, in general, one day in advance of the actual delivery of electricity and, in any case, before the realization of the uncertain factors. In this paper, we consider for simplicity that the system uncertainty stems only from the wind power production, which is modeled as a finite set $\Omega$ of scenarios $W_{f,t,\omega}$ with $\omega \in \Omega$.

The second-stage decision variables, namely, $P_{g,t,\omega}$, $L_{t,\omega}^{SH}$, $W_{f,t,\omega}^{SP}$ and $\delta_{n,t,\omega}$ determine the economic dispatch of the conventional generating units, the amount of load that is involuntarily shed, the amount of wind power that is curtailed, and the voltage angles at the network nodes, respectively. These variables adapt to the specific realization of the uncertainty and as such, are augmented with the scenario index $\omega$.

We start by providing the mathematical formulation of the two-stage stochastic unit commitment problem. In all cases, we consider that the marginal production cost of the wind generation is zero.

A. Two-Stage Stochastic Unit Commitment (SUC)

The two-stage stochastic unit commitment problem can be formulated as follows:

$$\begin{align*}
\text{minimize}_{\mathcal{H},\mathcal{W}} & \quad \sum_{t \in T} \sum_{g \in G} \left( C_{g}^{F} u_{g,t} + C_{g}^{SU} y_{g,t} + C_{g}^{ED} z_{g,t} \right) \\
& \quad + \sum_{\omega \in \Omega} \pi_{\omega} \left[ \sum_{t \in T} \sum_{g \in G} C_{g}^{V} P_{g,t,\omega} + \sum_{t \in T} \sum_{l \in L} C_{l}^{L} L_{l,t,\omega}^{SH} \right] \\
\text{s.t.} & \quad y_{g,t} - z_{g,t} = u_{g,t} - u_{g,t-1} \quad (\forall g, \forall t \in \{2, \ldots, T\}) \\
& \quad y_{g,t} - z_{g,t} = u_{g,t} - IS_{g} \quad (\forall g, \forall t \in \{1\}) \\
& \quad y_{g,t} + z_{g,t} \leq 1 \quad (\forall g, \forall t \in \{1, \ldots, T\}) \\
& \quad (L_{g}^{UP} + L_{g}^{DW}) > 0, \forall g, \forall t \leq L_{g}^{UP} + L_{g}^{DW} \\
& \quad \sum_{t = t - DT_{u} + 1}^{t} y_{g,\tau} \leq u_{g,t} \quad (\forall g, \forall t > L_{g}^{UP} + L_{g}^{DW}) \\
& \quad \sum_{t = t - DT_{u} + 1}^{t} z_{g,\tau} \leq 1 - u_{g,t} \quad (\forall g, \forall t > L_{g}^{UP} + L_{g}^{DW})
\end{align*}$$

where $\mathcal{H} = \{u_{g,t}, y_{g,t}, z_{g,t}\}$ and $\mathcal{W} = \{P_{g,t,\omega}, L_{t,\omega}^{SH}, W_{f,t,\omega}^{SP}, \delta_{n,t,\omega}, \omega \in \Omega\}$ are the sets of here-and-now and wait-and-see decisions, respectively. Furthermore, following [28], the initial state conditions are given by

$$IS_{g} = \begin{cases} 1 & \text{if } ON_{g} > 0 \\ 0 & \text{if } ON_{g} = 0 \end{cases}$$

$$L_{g}^{UP} = \min\{T, (UP_{g} - ON_{g}) IS_{g}\}$$

$$L_{g}^{DW} = \min\{T, (DT_{g} - OFF_{g}) (1 - IS_{g})\}$$

Problem (1)–(19) takes the form of a standard two-stage unit commitment formulation, which is similar, to a large extent, to those provided in the numerous works on the topic, see, for instance, [6], [27] and references therein. The objective is to minimize the expected system operating cost.
cost (1), which is made up of the no-load, start-up, shutdown, and variable production costs of the conventional generating units, and the cost of involuntarily load curtailment, in that order. Equations (2)–(4) model the changes in the on/off-commitment status of the power plants as these are started up or shutdown throughout the scheduling horizon, while (5)–(7) impose their minimum up- and down-time requirements. Equalities (8) constitute the set of nodal power balance equations according to a DC power flow model. The maximum and minimum power outputs of the generating units are enforced by (9) and (10), respectively, and their ramping limits through (11)–(14), as in [8] and [10]. The sets of inequalities (15) and (16) limit the involuntary load curtailment and the wind power spillage to the eventual power that is consumed and the eventual wind power that is produced, respectively. The set of equations (17) guarantee compliance with the transmission capacity limits. Finally, constraints (18) and (19) constitute variable declarations.

B. Two-Stage Robust Unit Commitment (RUC)

The two-stage robust unit commitment problem can be written as follows:

\[
\min_{\mathcal{H}} \sum_{t \in T, g \in G} \left( C_g^P y_{g,t} + C_g^{SU} y_{g,t} + C_g^{SD} z_{g,t} \right) + Q(\mathcal{H})
\]

s.t. (2) – (7), (19)

(20)

where

\[
Q(\mathcal{H}) = \max_{\omega \in \Omega} \min_{\mathcal{W}} \sum_{t \in T, g \in G} C_g^V P_{g,t,\omega} + \sum_{t \in T} \sum_{l \in L} C_l^{LH} L_{l,t,\omega}
\]

s.t. (8) – (18)

(21)

In the particular case that the uncertainty set \( \Omega \) is comprised of a finite number of atoms, outcomes or scenarios \( \omega \), the two-stage robust unit-commitment problem (20)–(23) can be equivalently recast as follows:

\[
\min_{\mathcal{H}, \mathcal{W}, \alpha} \sum_{t \in T, g \in G} \left( C_g^P y_{g,t} + C_g^{SU} y_{g,t} + C_g^{SD} z_{g,t} \right) + \alpha
\]

s.t. (2) – (19)

(24)

\[
\alpha \geq \sum_{t \in T, g \in G} C_g^V P_{g,t,\omega} + \sum_{t \in T} \sum_{l \in L} C_l^{LH} L_{l,t,\omega}, \ \forall \omega \in \Omega
\]

(25)

(2) – (19)

(26)

For convenience and ease of reference, the set of equations (24)–(26) is hereafter referred to as the scenario-based formulation of the two-stage robust unit-commitment problem. Note that the auxiliary variable \( \alpha \) equals the worst-case dispatch cost at the optimum. This variable is bounded from below by a finite set of linear constraints (25), one per scenario, that involve the second-stage decision variables \( P_{g,t,\omega} \) and \( L_{l,t,\omega}^{LH} \). Thus, the objective of the problem (24)–(26) is to minimize the total system operating cost for the worst-case scenario of the uncertainty.

In the following section, we introduce the proposed hybrid formulation of the two-stage unit commitment problem under uncertainty.

C. Hybrid Unit Commitment Problem (HUC)

Let us now split the set \( \Omega \) into \( k \) partitions with \( \Omega = \{1, \ldots, k\} \) being the partition set. We then define the series of subsets \( \Omega_1, \ldots, \Omega_p, \ldots, \Omega_k \), with \( \Omega_i \cap \Omega_j = \emptyset \) for all \( i \neq j \) and \( \Omega_1 \cup \ldots \Omega_p \cup \ldots \Omega_k = \Omega \), such that \( \Omega_p \) is linked to partition \( p \in P \). Furthermore, each partition \( p \in P \) is assigned a probability \( \rho_p \geq 1 \) such that \( \sum_{p \in P} \rho_p = 1 \).

The proposed hybrid two-stage unit commitment problem writes as follows:

\[
\min_{\mathcal{H}, \mathcal{W}, \rho_p} \sum_{t \in T, g \in G} \left( C_g^P y_{g,t} + C_g^{SU} y_{g,t} + C_g^{SD} z_{g,t} \right) + \mathcal{R}(\mathcal{H})
\]

s.t. (2) – (7), (19)

(27)

where

\[
\mathcal{R}(\mathcal{H}) = \sum_{p \in P} \rho_p \left( \max_{\omega \in \Omega_p} \min_{\mathcal{W}} \sum_{t \in T, g \in G} C_g^V P_{g,t,\omega} + \sum_{t \in T} \sum_{l \in L} C_l^{LH} L_{l,t,\omega} \right)
\]

s.t. (8) – (18)

(28)

In the particular case that the uncertainty set \( \Omega \) is finite, that is, \( \Omega = \{1, \ldots, \lambda\} \), where \( \lambda \) is the total number of possible outcomes or scenarios, problem (27)–(30) can be equivalently reformulated as follows:

\[
\min_{\mathcal{H}, \mathcal{W}, \rho_p} \sum_{t \in T, g \in G} \left( C_g^P y_{g,t} + C_g^{SU} y_{g,t} + C_g^{SD} z_{g,t} \right) + \sum_{p \in P} \rho_p \theta_p
\]

s.t. (2) – (19)

(31)

where \( \Omega_p \) is comprised of all the scenarios \( \omega \in \Omega \) that belong to partition \( p \in P \) and where the probability \( \rho_p \) assigned to partition \( p \) is computed as the sum of the probabilities of the scenarios that form part of it, that is,

\[
\rho_p = \sum_{\omega \in \Omega_p} \pi_\omega, \ \forall p \in P
\]

(32)

The objective (31) is then to minimize the expected system operating cost over the scenarios that deliver the worst-case dispatch cost within each partition.

Indeed, the auxiliary variable \( \theta_p \), one per partition \( p \in P \), equals the worst-case dispatch cost within partition \( p \), in a similar way as the auxiliary variable \( \alpha \) does in the robust unit commitment formulation (24)–(26) for the whole set of scenarios \( \Omega \). This way, problem (31)–(33) is expected to yield a unit-commitment plan that is “in between” the robust and the stochastic unit-commitment solutions in terms of the expected and the worst-case system operating cost. Furthermore, the closeness of the HUC solution to the stochastic and robust unit-commitment plans, and consequently its degree of conservatism, are controlled by the number \( k \) of partitions or clusters into which the scenarios are grouped. Indeed, if the number of partitions equals the number of scenarios, that is, \( k = \lambda \), the HUC model (31)–(33) reduces to (1)–(19) and the stochastic solution is obtained. In contrast, if only one single partition is considered \( (k = 1) \), we have that \( \Omega_1 = \Omega \) and therefore, problem (31)–(33) boils down to the scenario-based robust unit-commitment formulation (24)–(26). As a
result, the HUC solution coincides with the robust solution in such a case.

Hence, we can increase the degree of conservatism of the HUC solution by diminishing the number of partitions, and vice versa. For $1 < k < \lambda$, however, how efficiently and quickly the HUC solution transits from the robust to the stochastic unit-commitment plan, as $k$ increases, depends on the performance of the clustering technique. Later on in this paper, we evaluate and compare the performance of the proposed HUC formulation when solved using a range of different clustering methods. More specifically, we consider three non-hierarchical clustering techniques, namely, the k-means, k-medoids, and k-shape algorithms, which are used to group a data set into a given number $k$ of clusters. In brief, these algorithms assign each scenario $\omega \in \Omega$ to the partition $\Omega_p$, $p \in P$, with the nearest mean (k-means), the closest representative scenario (k-medoids) [22], or the most similar scenario-shape using a cross-correlation measure (k-shape) [23]. In particular, the k-means algorithm [24] has been reported to showcase a good performance in other related applications, e.g., for solving a probabilistic production cost model in [25] and a transmission and generation expansion planning problem in [26]. Furthermore, we also test an agglomerative hierarchical clustering method [22], where the dissimilarity of two clusters is measured as the maximum of the pairwise distances of the scenarios in the clusters. This method produces a hierarchy of partitions whereby the two nearest clusters merge into a new one as one moves up the hierarchy. Therefore, the top level in the hierarchy consists of one single cluster that comprises the complete set of scenarios.

### III. Solution Strategy: Parallelization and Decomposition

It is well known that the unit commitment problem is mixed-integer, NP-hard, and generally requires long solution times. This is especially true for realistic instances of the two-stage unit commitment problem under uncertainty. In the following we describe two alternative parallelization-and-decomposition schemes that we have designed to efficiently solve the proposed HUC formulation (31)–(33). For ease of exposition, we divide this description in two parts. In the first one, we explain how problem (31)–(33) can be decomposed per partition and scenario, while in the second part we elaborate on how the solution to the decomposed problem can be parallelized.

#### A. Problem Decomposition via Column-and-constraint Generation

We provide and discuss below two alternative ways to decompose the HUC problem (31)–(33), which we present as two variants of the same Scenario Partition and Decomposition Algorithm (SPDA).

1) Scenario Partition and Decomposition Algorithm—Variant 1 (SPDA1): Let us consider a certain partition $p \in P$ that comprises the subset of scenarios $\Omega_p$. Note that, for determining the optimal solution to the HUC problem (31)–(33), we only need those (hopefully few) scenarios in $\Omega_p$ that deliver the worst-case dispatch cost over partition $p$ for any feasible unit-commitment plan. Identifying those scenarios, however, may be computationally very costly. Instead, we describe below a procedure to find a subset $\Omega'_p \subseteq \Omega_p$ under which the HUC formulation provides a unit commitment solution close to the one given under the full scenario set $\Omega_p$. Furthermore, the proposed algorithm builds the reduced sets $\Omega'_p$, $p \in P$, in parallel for each partition $p$.

To build $\Omega'_p$ from $\Omega_p$, the latter being the outcome of a certain clustering algorithm, we develop a master-subproblem decomposition scheme per partition based on the column-and-constraint generation procedure described in [9]. In the sequel we will refer to this decomposition scheme as Primal Cut Algorithm after the solution strategy introduced in [10] whereby the master problem is gradually enlarged with the addition of cuts expressed in terms of the primal variables.

Each master problem (one per partition) is a mixed-integer programming problem that involves both first-stage and second-stage decision variables and that has the following form at iteration $i$ of the column-and-constraint generation algorithm:

$$\begin{align*}
\text{minimize} & \quad \mathbf{w}^i, \mathbf{w}^{\delta, \theta}_p, \theta_p \sum_{t \in T} \sum_{g \in G} \left( C^F_{g,t} y^i_{g,t} + C^S_{g} y^i_{S,g,t} + C^D_{g} z^i_{g,t} \right) + \theta_p \\
\text{s.t.} & \quad (2)-(7), (19) \\
\theta_p & \geq \sum_{t \in T} \sum_{g \in G} C^V_{g} p^i_{g,t}, \forall \omega \in \Omega^i_p \\
(8)-(18), & \quad \forall \omega \in \Omega^i_p
\end{align*}$$

where $\mathcal{H}^i = \{ u^i_{g,t}, y^i_{g,t}, z^i_{g,t} \}$ and $\mathcal{W}^i = \{ p^i_{g,t}, W^i_{g,t}, \delta_{n,t,\omega}^i \}, \forall \omega \in \Omega^i_p$. Note that $\Omega^i_p = \emptyset$. As the algorithm proceeds, $\Omega^i_p$ is augmented with those possibly few scenarios $\omega \in \Omega_p$ that are needed to reconstruct the partition-worst-case recourse cost as a function of the first-stage decision variables $u^i_{g,t}$, $y^i_{g,t}$, and $z^i_{g,t}$ in the form of (37)–(38).

Constraint (37) can be interpreted as a primal cut, as compared to those cuts that are constructed from dual information, as it is the case, for example, of a standard Benders cut.

The subproblems are linear programming problems (LP) that determine the second-stage decision variables $p^i_{g,t,\omega}$, $W^i_{g,t,\omega}$, and $\delta_{n,t,\omega}^i$ with $u^i_{g,t}$, $y^i_{g,t}$, and $z^i_{g,t}$ fixed at the values given by the master problem. A subproblem like (39)–(40) is solved for each scenario $\omega \in \Omega_p$.

$$\begin{align*}
\text{minimize} & \quad \mathcal{W}_{\omega} \sum_{t \in T} \sum_{g \in G} C^V_{g} p^i_{g,t} \sum_{t \in T} \sum_{g \in G} C^L_{g} L^i_{g,t,\omega} + \sum_{t \in T} \sum_{g \in G} C^L_{g} L^i_{g,t,\omega} \\
\text{s.t.} & \quad (8)-(18)
\end{align*}$$

where $\mathcal{W}_{\omega} = \{ p^i_{g,t,\omega}, L^i_{g,t,\omega}, W^i_{g,t,\omega}, \delta_{n,t,\omega}^i \}$.

The scenario $\omega'$ for which the associated subproblem (39)–(40) yields the highest dispatch cost or is infeasible is used to construct a set of primal constraints in the form of (37)–(38) that is added to the master problem by setting $\Omega^i_{p+1} = \Omega^i_p \cup \{ \omega' \}$. It is worth noticing, however, that this subproblem infeasibility is not a concern in our case due to the possibility of shedding load and spilling wind.

One instance of the primal cut algorithm is run for each partition $p \in P$ in parallel. Each of these instances works, therefore, with one master problem and a number of subproblems equal to the number of scenarios in each partition, that is, equal to $\text{card}(\Omega_p)$. Furthermore, each instance of the algorithm concludes by delivering the set
of selected scenarios $\Omega'_p \subset \Omega_p$ for partition $p$. The last step of our solution strategy consists then in solving the HUC problem (31)–(33) where $\Omega_p$ is replaced with the reduced scenario set $\Omega'_p$.

We describe below how this solution strategy proceeds step by step.

1. Choose the number $k$ of partitions and apply a clustering method to the complete set of scenarios $\Omega$ in order to assign each scenario to a certain partition $p$.
2. Create one instance of the primal cut algorithm for each partition $p \in P$.
3. Initialization: Set $i = 0$ and $\Omega_p^0 = \emptyset$.
4. Solve the master problem (MP). Return the optimal solution found by the branch-and-cut algorithm and denote this solution by $(u_{i1}, y_{i1}, z_{i1})$. Calculate a lower bound $LB$ as $\sum_{i \in T} \sum_{g \in G} (C^F_{g} u_{i1, g} + C^SU_{g} y_{i1, g} + C^{SD}_{g} z_{i1, g}) + \theta_p$.
5. Solve the subproblems (SP) with the first-stage decision variables fixed at $(u_{i1}, y_{i1}, z_{i1})$. Once the SP are solved, the scenario $\omega_i$ associated with the subproblem that yields the highest dispatch cost is identified and included into the reduced set $\Omega_p^{i+1}$, i.e., $\Omega_p^{i+1} = \Omega_p^i \cup \{\omega_i\}$. Compute an upper bound $UB$ as $\sum_{i \in T} \sum_{g \in G} (C^F_{g} u_{i, g} + C^SU_{g} y_{i, g} + C^{SD}_{g} z_{i, g}) + \sum_{i \in T} \sum_{g \in G} C^{V}_{g} P_{g, \omega_i} + \sum_{i \in T} \sum_{l \in L} C^{L}_{g, l, \omega_i}$.
6. Convergence check: If $|UB - LB| \leq \epsilon$, then set $i := i + 1$ and go to step 4.
7. Once all the instances of the primal cut algorithm have converged, the HUC problem (31)–(33) is solved for all $p \in P$ and for all $\omega \in \Omega_p$. The reduced set $\Omega_p^i$ is made up of all those scenarios $\omega \in \Omega_p$ that determine the worst-case dispatch cost within partition $p$.

A pseudocode for the proposed decomposition scheme is provided in Algorithm 1. For ease of notation, let $x(x^i)$ denote the vector of first-stage variables (at iteration $i$).

**Algorithm 1 Scenario Partition and Decomposition Algorithm: Variant 1 (SPDA1)**

1. Choose $k$ and apply $k$-means to $\Omega$.
2. for all $p \in P$ do
3.   Set $i := 0$ and $\Omega_p^0 = \emptyset$.
4.   repeat
5.     Solve Master Problem
6.     Return optimal solution $x^i$
7.     Compute Lower Bound $LB$
8.     Set $x := x^i$ and solve SP $\forall \omega \in \Omega_p$
9.     Compute Upper Bound $UB$
10. Identify worst-case scenario $\omega^i$
11. Set $\Omega_p^{i+1} := \Omega_p^i \cup \{\omega^i\}$
12. Set $i := i + 1$
13. until $|UB - LB| \leq \epsilon$
14. Set $\Omega_p := \Omega_p^{i-1}$
15. end for
16. Solve HUC replacing $\Omega_p$ with $\Omega'_p, \forall p \in P$.

Notice that SPDA1 works in a similar way to a scenario reduction technique that retains the most detrimental scenarios in terms of system operating cost. This confers robustness to the solution of the proposed HUC problem. Moreover, the last command line in SPDA, which involves solving the HUC model for the reduced scenario sets $\Omega'_p, \forall p \in P$, could be carried out as well via further decomposition (see, for instance, [29]), although this possibility has not been explored in this paper.

2) Scenario Partition and Decomposition Algorithm—Variant 2 (SPDA2): Algorithm SPDA1 is inspired from the idea that only a few scenarios in the full set $\Omega_p$ may be needed to determine the worst-case dispatch cost for partition $p$ and for any feasible unit-commitment plan. Based on this, Algorithm SPDA1 generates one instance of the Primal Cut Algorithm per partition.

The reader should notice, however, that, in order to compute the minimum expected system operating cost over the scenarios that deliver the worst-case dispatch cost within each partition, one might not need to calculate the partition-worst-case dispatch cost for any feasible unit-commitment plan and therefore, one might not need all those scenarios that algorithm SPDA1 aims to identify, but possibly a smaller number of them. With this in mind, we construct a second variant of our Scenario Partition and Decomposition Algorithm, which we denote SPDA2.

Unlike its first variant, SPDA2 only generates one single instance of the Primal Cut Algorithm, that is, a single master-subproblem system. The master problem is a mixed-integer programming problem that takes the following form at iteration $i$ of the column-and-constraint generation procedure:

$$\begin{align*}
\text{minimize} & \quad \sum_{t \in T} \sum_{g \in G} (C^F_{g} u_{i, g} + C^SU_{g} y_{i, g} + C^{SD}_{g} z_{i, g}) + \\
& \quad \sum_{p \in P} \rho_p \theta_p \\
\text{subject to} & \quad (2) - (7), \quad (19) \\
& \quad \theta_p \geq \sum_{t \in T} \sum_{g \in G} C^V_{g} P_{g, t, \omega} + \sum_{t \in T} \sum_{l \in L} C^L_{g, l, \omega} \\
& \quad (\forall p, \forall \omega \in \Omega^i_p) \\
& \quad (8) - (18), \quad \forall p \in P, \forall \omega \in \Omega^i_p,
\end{align*}$$

where $H^i_p = \{w_1, w_2, ..., w_{j^i}, \omega^i\}$ and $W^{SP, i} = \{p_0, p_1, ..., p_{i-1}\} \cup \{W^{SP, i} \}$.

We provide next a step-by-step description of SPDA2.

1. Choose the number $k$ of partitions and apply a clustering method to the full set of scenarios $\Omega$ in order to assign each scenario to a certain partition $p$.
2. Create one instance of the primal cut algorithm.
3. Initialization: Set $i = 0$ and $\Omega_p^0 = \emptyset, \forall p \in P$.
4. Solve the master problem (MP). Return the optimal solution found by the branch-and-cut algorithm and denote this solution by $(u_{i1}, y_{i1}, z_{i1})$. Calculate a lower bound $LB$ as $\sum_{i \in T} \sum_{g \in G} (C^F_{g} u_{i1, g} + C^SU_{g} y_{i1, g} + C^{SD}_{g} z_{i1, g}) + \sum_{p \in P} \rho_p \theta_p$.
5) Solve the subproblems (SP) with the first-stage decision variables fixed at \((u_{g,t}^j, y_{g,t}^j, z_{g,t}^j)\). Once the SP are solved, the scenarios \(\{\omega_1^p, \omega_2^p, \ldots, \omega_{k}^p, \ldots, \omega_{k}^p\}\) associated with the subproblems that yield the highest dispatch cost in each partition \(p\) are identified and included into the reduced sets \(\Omega_p^{k+1}, p \in P\), i.e., \(\Omega_p^{k+1} = \Omega_p^k \cup \{\omega_p^k\}\). Compute an upper bound \(UB\)

\[
UB = \sum_{t \in T} \sum_{g \in G} \left( C^{P}_{g} u_{g,t}^j + C^{SU}_{g} y_{g,t}^j + C^{SD}_{g} z_{g,t}^j \right) + \sum_{p \in P} \rho_p \left( \sum_{t \in T} \sum_{g \in G} C^{V}_{g} p_{g,t}^{j} \omega_p^j + \sum_{t \in T} \sum_{t' \in L} C_{t,t'}^{L,S}_{t,t'} \right).
\]

6) Convergence check: If \(|UB - LB| \leq \epsilon\), being \(\epsilon\) a user-specified tolerance value, the iterative process stops and \((u_{g,t}^j, y_{g,t}^j, z_{g,t}^j)\) is returned as the optimal solution to the HUC problem. If \(|UB - LB| > \epsilon\), then set \(i := i + 1\) and go to step 4.

A pseudocode of this solution strategy is provided in Algorithm 2 below.

**Algorithm 2 Scenario Partition and Decomposition Algorithm: Variant 2 (SPDA2)**

1. Choose \(k\) and apply k-means to \(\Omega\).
2. Set \(i := 0\) and \(\Omega_i^p = \emptyset\) for all \(p \in P\).
3. repeat
   4. Solve Master Problem
   5. Return optimal solution \(x_i\)
   6. Compute Lower Bound \(LB\)
   7. Set \(x := x^*\) and solve SP \(\forall \omega \in \Omega\)
   8. Compute Upper Bound \(UB\)
   9. Identify worst-case scenario \(\omega_p^k\) in each \(p \in P\)
   10. Set \(\Omega_{i+1}^p := \Omega_i^p \cup \{\omega_p^k\}\) for all \(p \in P\)
11. Set \(i := i + 1\)
12. until \(|UB - LB| \leq \epsilon\)
13. Return \(x := x^{i-1}\) as the solution to HUC

The two variants of the proposed Scenario Partition and Decomposition Algorithm have several key differences, namely:

1. SPDA2 guarantees convergence to the HUC solution that is optimal under the partition \(P\) of the full scenario set \(\Omega\), while SPDA1 does not. Indeed, SPDA1 is heuristic, because it does not ensure that the scenarios the algorithm selects from each partition \(p\) are enough to deliver the worst-case dispatch cost for that partition under any first-stage solution. Our numerical experiments show, however, that SPDA1 performs very well in the sense that it provides unit commitment solutions that are optimal for the full HUC problem (31)–(33). Furthermore, it should be noticed that SPDA1 can be used to warm-start SPDA2.

2. Let \(\Omega_i^p\) and \(\Omega_i^p\) be the reduced set of scenarios retained by the master problems of SPDA1 and the sole master problem of SPDA2 at the end of the algorithms, respectively. In all simulations (see the case study in Section IV) it holds that \(\text{card}(\Omega_i^p) \leq \text{card}(\Omega_i^p) \leq \text{card}(\Omega) = \lambda\).

3. At every iteration, SPDA2 solves one single master problem, while SPDA1 solves \(k\), but smaller master problems, with \(k = \text{card}(P)\).

4. SPDA1 must solve the HUC problem for the reduced scenario set \(\Omega_i^p\) at the end of the algorithm.

Differences 2-4 above determine how these two variants compare in computational terms. In particular, it is expected that, since SPDA1 works with smaller master problems at each iteration, but must solve a possibly larger HUC problem in the end, SPDA1 will perform better than SPDA2 when the number of partitions is kept sufficiently low. This is corroborated by our numerical experiments.

**B. Parallelization of the Solution Algorithm**

In the case of SPDA1 both the outer “for-loop” and the solution to the \(\omega\)-indexed subproblems in the pseudocode of “Algorithm 1” are amenable to parallelization. For this purpose, we make use of the DTU High Performance Computing (HPC) Facility [30]. We create \(k\) jobs, each representing an instance of the primal cut algorithm for each of the \(k\) partitions into which we divide the scenario set \(\Omega\). These jobs are simultaneously submitted to the HPC Cluster, where they are concurrently executed, as there is no need for communication in between the workers (nodes or cores).

We submit each of the \(k\) jobs to a different node, using the same amount of resources per node. Within every node, the subproblems are solved in a multi-threaded environment using the Gather-Update-Solve-Scatter Facility in GAMS [31]. This tool allows treating each subproblem, one per scenario in the partition under consideration, as a different parametrization of the same linear programming model, which is then generated only once by GAMS. Likewise, the solutions to all the subproblems (or portions thereof) are retrieved back to GAMS in a single transaction.

In the case of SPDA2, the solution to the \(\omega\)-indexed subproblems are also solved in the same way as in SPDA1, that is, by means of the the Gather-Update-Solve-Scatter Facility in GAMS [31].

**IV. Case Studies**

In the following the unit-commitment solution provided by the proposed HUC formulation is tested on the IEEE 14-node power system [32] and the IEEE 3-Area RTS-96 system. The latter power system is also used to evaluate and compare the performance of the SPDA algorithms in Sections IV-C and IV-D.

The IEEE 14-node system comprises 14 nodes, 5 generators, 20 lines and 11 loads. We also add one wind farm to node 5, whose power production is modeled by the ten scenarios provided in Table IV of the Appendix.

The IEEE 3-Area RTS-96 system consists of 72 nodes, 96 generators, 107 lines, and 51 loads. Besides, we add 15 wind farms of 200-MW capacity each and location given by Table V in the Appendix. Thus, the wind power capacity represents 29% of the total generating capacity installed in the power system. The technical characteristics of the generating units, the demand and the transmission lines are available online [33]. The wind power production scenarios used for this case study come from [34], where the spatio-temporal dependencies of wind power generation are considered. More specifically, the study in [34] provides 100 scenarios of wind power production that were generated for 15 control areas and 43 lead times in western Denmark. However, for this work, only 50 equiprobable scenarios and 24 lead times are considered.

We set the MIP tolerance gap to 0 in all the simulations pertaining to the IEEE 14-node system, while we allow for
TABLE I: Results of the comparison between Zhao model and the proposed HUC formulation.

<table>
<thead>
<tr>
<th>UCP</th>
<th>CCD [$\times$10^4]</th>
<th># Partitions</th>
<th>ZWF</th>
<th>ETC [$\times$10^4]</th>
<th>WC0TC [$\times$10^4]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>62569</td>
<td>13</td>
<td>0.8</td>
<td>1</td>
<td>28602.2</td>
</tr>
<tr>
<td>α</td>
<td>3.7510</td>
<td>1 - 5</td>
<td>2.0</td>
<td>1.0 - 1.8</td>
<td>28713.14</td>
</tr>
<tr>
<td>2</td>
<td>35381</td>
<td>1 - 5</td>
<td>2.0</td>
<td>1.0 - 1.8</td>
<td>30942.10</td>
</tr>
</tbody>
</table>

a MIP tolerance gap of up to $3 \cdot 10^{-4}$ (0.03%) in those numerical experiments carried out on the IEEE 3-Area RTS-96 system.

A. Evaluation of the Effect of the Number of Partitions

We consider the IEEE 14-node system and compare the unit-commitment plans resulting from the proposed HUC formulation with those obtained using the method proposed in [19], which we refer to as Zhao model hereafter. This method minimizes a convex combination of the expected and the worst-case costs, where the convex combination is defined by the decision-maker through a weighting factor $\alpha$. This way, Zhao model allows for a Pareto-efficient control of the degree of conservatism of the resulting unit-commitment solution in terms of the expected and worst-case system operating costs. For this reason, we use Zhao model here as an ideal benchmark against which we compare the ability of our proposed HUC formulation to find unit-commitment plans with a varying degree of conservatism by changing the number of partitions. The comparison is conducted for a number of partitions in the HUC problem ranging from 1 to 10 and a number of values for the weighting factor in Zhao model varying from 0 to 1.

The outcome of this comparison is that both Zhao model and our HUC formulation deliver the same only two unit-commitment plans, which we denote by UCP1 and UCP2. We measure the quality of these two different unit-commitment solutions in terms of both the expected and the worst-case system operating cost. The results are collated in Table I. The columns of this table provide the unit-commitment plan (UCP), the commitment cost (CCD), the number of partitions considered in our HUC formulation, the value of the weighting factor used in Zhao model (ZWF), the expected total cost (ETC) and the worst-case total cost (WC0TC), in that order. Costs are given in US dollars. Furthermore, recall that the pure stochastic and robust unit-commitment solutions are obtained for a number of partitions equal to 10 and 1 in the proposed HUC formulation, and for values of the weighting factor in Zhao model equal to 1 and 0, respectively.

For this specific test we have used the k-means clustering method to define the partitions.

We can see from Table I that UCP2 is more conservative than UCP1 and that there exists a one-to-one correspondence between the unit-commitment solutions provided by Zhao model and those given by the proposed HUC formulation. Thus, the proposed HUC formulation offers an alternative way to control the degree of conservatism of the resulting unit-commitment solution by way of the number of partitions. Besides, the efficient frontier (made up of Pareto-efficient points) is, in general, non-linear and non-continuous. Consequently, tuning the solution conservatism through the number of partitions, which is a natural number that ranges from 1 to the number of scenarios considered, is more practical than doing it through a weighting factor $\alpha$ that is a real number belonging to the closed real interval [0, 1].

B. Evaluation of the Impact of the Clustering Technique

Next we analyze how the proposed HUC formulation performs when different clustering techniques are used to construct the partitions. For this purpose, we conduct some numerical tests on the IEEE 3-Area RTS-96 system.

![Fig. 1: HUC total cost as a function of the number of partitions and for four different clustering techniques.](image_url)

Fig. 1 represents the objective function value (31) of the HUC formulation as the number of partitions increases. Each plot corresponds to a different clustering technique. As mentioned above, we consider the non-hierarchical clustering algorithms k-means, k-shape, and k-medoids, and an agglomerative hierarchical method.

In general terms, all the four clustering techniques prompt HUC solutions for which the HUC objective function value decreases as the number of partitions increases, as expected, but they do so at different speeds. In particular, the k-means and the hierarchical procedures achieve large reductions in the HUC cost with very few partitions. In practice, this means that these two clustering algorithms are able to prompt solutions from the HUC formulation that are close to the pure stochastic one with few partitions. This is clearly an advantage for two reasons at least: first, because our objective is, all in all, to approach the stochastic unit-commitment solution in a robust manner, that is, while keeping the worst-case cost under control; second, because, as we shall see later, the time required to solve the proposed HUC formulation decreases as so does the number of partitions.

In contrast, the plot corresponding to the k-shape method transits slowly from the pure robust solution to the pure stochastic one, which means that this methods requires a higher number of partitions to reduce the degree of conservatism of the UC solution.

It is interesting to note that, unlike the hierarchical clustering technique, the k-means, k-shape and k-medoids algorithms do not guarantee a monotone decrease in the HUC cost with the number of partitions. This is due to the non-hierarchical nature of these clustering methods, whereby the partitions they produce for $k - 1$, $k$ and $k + 1$ clusters bear no relation, that is, they do not necessarily result from splitting or merging existing clusters.
The results that follow have been obtained using the hierarchical clustering technique, although very similar results are also obtained if the k-means algorithm is used.

C. Evaluation of the Decomposition Schemes

In this section we use the case study based on the IEEE 3-Area RTS-96 system to investigate and compare three different approaches for solving the proposed HUC formulation, namely, the two variants of the proposed Scenario Partition and Decomposition Algorithm (SPDA1 and SPDA2) and a solution strategy that merely consists in directly solving the “raw” HUC problem (without decomposition). All these alternative solution approaches are coded in GAMS using CPLEX 12.6.1 and implemented in the DTU HPC Cluster. The DTU HPC Cluster is a composite of a variety of hardware components of different technical characteristics. Therefore, we refer the reader to [30] for further and detailed information on the cluster and its components. We solve the raw HUC (without the decomposition) and the SPDA2 using one node in a multi-threading configuration that counts on 10 Intel cores, while SPDA1 is implemented in a multi-node and multi-threaded environment. In particular, partitions are solved in parallel, each in a different node of the cluster with up to 10 Intel cores per node. Lastly, the reduced HUC problem is solved using again one node employing up to 10 Intel cores. In both SPDA1 and SPDA2 the subproblems are solved using the Gather-Update-Solve-Scatter Facility in GAMS.

Table II provides solutions times, achieved MIP gap (in percentage), number of scenarios retained in the final master problem, and expected and worst-case costs (in US dollars) for the three different solution strategies and for a different number of partitions. It is apparent that both SPDA1 and SPDA2 achieves remarkable time reductions without jeopardizing the solution quality at all. As expected, SPDA2 solves a last master problem with a few number of scenarios. However, in the intermediate steps of the algorithm, SPDA1 works with smaller master problems which may make it more computationally efficient for a low number of partitions. Note also that the time savings attained by the parallelization and decomposition algorithms gradually decrease with the number of partitions.

All the results obtained so far suggest that, in a practical setup, we could tackle the stochastic unit commitment problem as follows: We solve the proposed HUC formulation using the SPDA1 algorithm, starting with one single partition (corresponding to the most risk-averse solution) and gradually increasing the number of partitions until either the decision-maker is satisfied with the performance of the last solution obtained (recall that the level of conservatism of the solution decreases with the number of partitions) or the amount of available time has been reached. Bear in mind that the k-means and the hierarchical clustering methods with few partitions prompt solutions from the HUC formulation that are close to the stochastic one and that the SPDA1 is precisely most efficient when the number of partitions is small. Note that such a procedure is not that easy to implement with the method presented in [19], because the weighting factor $\alpha$ is a real number ranging from 0 to 1, both inclusive, and the relation between $\alpha$ and the efficient frontier can be non-linear and non-continuous. Therefore, it would be very difficult to define a step size for gradually increasing $\alpha$ as we can naturally do through the number of partitions.

D. Comparison with a Scenario Reduction Technique

We conclude our numerical study by comparing the unit-commitment solution given by SPDA1 or SPDA2 with that yielded by the widely-used fast forward scenario reduction technique [35]. The results of this comparison are summarized in Table III in terms of solution time (in seconds), MIP gap finally achieved (in percentage), and expected and worst-case costs (in US dollars). We use the fast forward selection algorithm to produce reduced scenario sets of varying cardinality. For the sake of a fair comparison, we make the cardinality of the reduced scenario set coincide with the number of scenarios retained (in the form of primal cuts) in the last master problem solved by SPDA1 or SPDA2 when one, two, five, eight and ten partitions are considered.

It is evident that, even though the use of a scenario reduction technique notably reduces the computational burden of the stochastic unit commitment problem, the unit-commitment solutions provided by SPDA1 and SPDA2 are significantly better in terms of the worst-case cost, while achieving expected costs close to that of the pure stochastic solution. Indeed, our solution approach yields unit-commitment plans that result in a worst-case cost that is, at least, 1.6% lower than the best worst-case cost provided by the scenario reduction technique (with 18 scenarios). Likewise, our solution approach provides unit-commitment plans that result in an expected cost that is, at most, 0.07% worse than that corresponding to the pure stochastic solution. This supports the conclusion that solving the proposed HUC formulation using either SPDA1 or SPDA2 constitutes a computationally efficient strategy to determine well-performing unit-commitment solutions in terms of the expected and worst-case costs.

<table>
<thead>
<tr>
<th># Scenarios</th>
<th>Time [min]</th>
<th>GAP [%]</th>
<th>EC [$10^{14}$]</th>
<th>WC [$10^{14}$]</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2.8</td>
<td>0.0284</td>
<td>182.3564</td>
<td>212.0134</td>
</tr>
<tr>
<td>8</td>
<td>5.3</td>
<td>0.0293</td>
<td>181.0866</td>
<td>201.2821</td>
</tr>
<tr>
<td>9</td>
<td>7.4</td>
<td>0.0263</td>
<td>180.9487</td>
<td>200.6126</td>
</tr>
<tr>
<td>10</td>
<td>9.7</td>
<td>0.0294</td>
<td>180.9537</td>
<td>200.6188</td>
</tr>
<tr>
<td>11</td>
<td>11.1</td>
<td>0.0300</td>
<td>180.7298</td>
<td>197.9442</td>
</tr>
<tr>
<td>12</td>
<td>11.3</td>
<td>0.0291</td>
<td>180.6371</td>
<td>197.0820</td>
</tr>
<tr>
<td>14</td>
<td>11.7</td>
<td>0.0269</td>
<td>180.3185</td>
<td>194.0619</td>
</tr>
<tr>
<td>17</td>
<td>19.4</td>
<td>0.0287</td>
<td>180.2918</td>
<td>193.7437</td>
</tr>
<tr>
<td>18</td>
<td>25.6</td>
<td>0.0297</td>
<td>180.2918</td>
<td>193.7437</td>
</tr>
</tbody>
</table>
V. CONCLUSION AND FUTURE RESEARCH

In this paper we propose a new formulation of the unit commitment problem under uncertainty that allows us to find unit-commitment plans that perform relatively well in terms of both the expected and the worst-case system operating cost. The new formulation relies on clustering the scenario data set into a number of partitions. The expectation of the system operating cost is then taken over those scenarios that result in the worst-case dispatch cost within each partition. The conservatism of the so-obtained unit-commitment solution (that is, how close it is to the pure scenario-based stochastic or robust unit-commitment plan) is controlled via the user-specified number of partitions. We also develop parallelization-and-decomposition schemes to efficiently solve the proposed unit-commitment formulation. Our numerical results show that our scheme is able to dramatically reduce the required running time while improving the optimality of the solution found.

We envision three avenues of future research at least. First, we would like to investigate how our methodology adapts to the case where the partitions are continuous uncertainty sets. Second, it would also be interesting to explore different possibilities to speed up the SPDA algorithms, for example, by warm-starting the solution to the proposed HUC formulation for \( k \) partitions with the solution obtained for \( k - 1 \) partitions. Finally, we would like to extend our formulation and the associated solution approach to a multi-stage setup.

REFERENCES


APPENDIX

This Appendix provides the wind power production scenarios used in the case study based on the IEEE 14-node system (Table IV) and the location of wind farms in the 3-Area RTS-96 system (Table V).

TABLE IV: Wind power production scenarios [MW] for the IEEE 14-node system.

<table>
<thead>
<tr>
<th>t</th>
<th>f1</th>
<th>f2</th>
<th>f3</th>
<th>f4</th>
<th>f5</th>
<th>f6</th>
<th>f7</th>
<th>f8</th>
<th>f9</th>
<th>f10</th>
<th>f11</th>
<th>f12</th>
<th>f13</th>
<th>f14</th>
<th>f15</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
</tr>
<tr>
<td>2</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
</tr>
<tr>
<td>3</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
</tr>
<tr>
<td>4</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
</tr>
<tr>
<td>5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
</tr>
<tr>
<td>6</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
</tr>
<tr>
<td>7</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
</tr>
<tr>
<td>8</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
</tr>
<tr>
<td>9</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
</tr>
<tr>
<td>10</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
</tr>
<tr>
<td>11</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
</tr>
<tr>
<td>12</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
</tr>
<tr>
<td>13</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
</tr>
<tr>
<td>14</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
</tr>
<tr>
<td>15</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
<td>55.5</td>
</tr>
</tbody>
</table>

TABLE V: Location of wind farms in the IEEE 3-Area RTS-96 system.

<table>
<thead>
<tr>
<th>Unit</th>
<th>Node</th>
<th>Unit</th>
<th>Node</th>
<th>Unit</th>
<th>Node</th>
<th>Unit</th>
<th>Node</th>
<th>Unit</th>
<th>Node</th>
<th>Unit</th>
<th>Node</th>
<th>Unit</th>
<th>Node</th>
<th>Unit</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>1s</td>
<td>1</td>
<td>2s</td>
<td>3</td>
<td>4s</td>
<td>5</td>
<td>6s</td>
<td>7</td>
<td>8s</td>
<td>9</td>
<td>10s</td>
<td>11</td>
<td>12s</td>
<td>13</td>
<td>14s</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>14</td>
<td>20</td>
<td>24</td>
<td>30</td>
<td>34</td>
<td>40</td>
<td>44</td>
<td>50</td>
<td>54</td>
<td>60</td>
<td>64</td>
<td>70</td>
<td>74</td>
<td>80</td>
<td>84</td>
</tr>
</tbody>
</table>

Ignacio Blanco received the B.S. degree in Energy Engineering in 2013 from the Rey Juan Carlos University, Spain. In 2015 he obtained the M.Sc. degree in Energy Management from the Technical University of Madrid. He is currently completing his Ph.D. at the Technical University of Denmark.

His research interests are large-scale optimization, data analysis, energy planning and decision-making under uncertainty for integrated energy systems.

Juan M. Morales (S’07-M’11-SM’16) received the Ingeniero Industrial degree from the Universidad de Málaga, Málaga, Spain, in 2006, and a Ph.D. degree in Electrical Engineering from the Universidad de Castilla-La Mancha, Ciudad Real, Spain, in 2010. He is currently an associate professor in the Department of Applied Mathematics at the University of Málaga in Spain.

His research interests are in the fields of power systems economics, operations and planning; energy analytics and smart grids; decision-making under uncertainty, and electricity markets.