An Asynchronous Decomposition Algorithm for Security Constrained Unit Commitment under Contingency Events

Kibaek Kim,* Mihai Anitescu,∗† and Victor M. Zavala∗‡
∗Mathematics and Computer Science Division
Argonne National Laboratory, Lemont, IL, USA
†Department of Statistics
The University of Chicago, Chicago, IL, USA
‡Department of Chemical and Biological Engineering
University of Wisconsin-Madison, Madison, WI, USA

Abstract—We present a parallel and asynchronous decomposition algorithm for solving security constrained unit commitment (SCUC) problem under contingency events in system components in combination with uncertain wind power generation. The problem is formulated as a two-stage stochastic mixed-integer program, where the first stage schedules slow generators and the second stage adjusts the schedules for fast generators and dispatches power for a given contingency event. Our algorithm uses an asynchronous variant of the Lagrangian dual decomposition that can better handle high imbalance in computational times for scenario subproblems (which are themselves mixed-integer programs). We also present a special case of the SCUC problem that considers a deterministic wind power generation and develop a Benders decomposition for solving this special case. In our computational study, we solve the SCUC model for the WECC 225-bus system that incorporates the single contingency events on transmission lines. We show that the asynchronous algorithm can significantly reduce solution times (by up to a factor of eight) of an off-the-shelf synchronous dual decomposition algorithm.

Index Terms—Asynchronous parallel computing, security-constrained unit commitment, dual decomposition

I. INTRODUCTION

The development of security criteria is key to achieve reliability and stability of the power grid. The $n - K$ security criterion, in particular, ensures that the power system is capable of satisfying the load for any possible combinations of $K$ out of $n$ grid component failures. In particular, North American Electric Reliability Corporation has established transmission system planning performance requirements, which suggests the system being sustainable to every single component failure without load loss; that is, $n - 1$ compliance.

Security constrained unit commitment (SCUC) model has been used for modeling unit commitment with contingency events in load forecast, generation forecast, and component failures (e.g., [1], [2]). Papavasiliou and Oren [2] formulated the SCUC model as a two-stage stochastic mixed-integer program (SMIP), where the first stage schedules slow generators and the second stage schedules fast generators and dispatches power in order to satisfy load for a given uncertain contingency realization. The authors also presented the numerical examples using a test instance with 30 scenarios (10 wind scenarios and 2 contingency events).

In this paper, we present parallel decomposition methods for SMIP problems of the SCUC model, as used in [2]. In particular, we consider single contingency events on transmission lines (i.e., single line failures) as well as uncertain wind power generation. The SMIP problems that arise in our SCUC model are computationally challenging because the size of the problem (particularly, the number of binary variables) increases in the number of scenarios and because binary variables appear in the first and second stage. This prevents the use of Benders-type methods (e.g., [3], [4]).

We develop a dual decomposition (DD) method for solving the SCUC with a number of contingency scenarios. DD is the Lagrangian relaxation of the nonanticipativity constraints [5], which provides an effective parallel decomposition approach to solve this type of SMIP (e.g., [6], [7]). A key component of the DD method is the procedure to find a sequence of dual variables for the Lagrangian dual function. In our DD method, we apply a recently-developed bundle method with trust-region (TR) constraints, which we call bundle-trust-region (BTR) method. The BTR method is a variant of bundle methods that solves a master problem for finding a new dual variable with the regularization of the search space by the TR constraints. The master problem is obtained by approximating the Lagrangian dual function with a set of linear inequalities. The key advantage of the bundle methods is the finite termination of algorithm at optimum (i.e., an optimal Lagrangian dual value in the DD method).
The subgradient method is another common approach to updating dual variables, which uses a subgradient of the Lagrangian dual function with step-size rules (e.g., [2], [6]). However, finding an optimal step-size is computationally impractical in such methods. The subgradient method requires ad-hoc step-size selection criteria and thus suboptimal stopping criteria, as compared with the finite termination at optimum by the bundle methods.

Moreover, in the context of parallel DD method, another challenge is that the scenario (event) subproblems lead to severe imbalances in computing time, because different sets of component failures induce different network topologies [8]. To address this issue, we also present the asynchronous variant of the BTR method and the computational performances that show significant reduction in solution time.

We also develop a Benders decomposition method for solving the SCUC with all single contingency scenarios for a given wind power generation, where the contingency scenario subproblems are solved in parallel. A key observation is that a feasibility cut can be generated based on a binary logic when the first stage has binary variables only. Note, however, Benders-type optimality cuts cannot be generated due to the second-stage binary variables (see discussion in [7]). Our Benders method is used to identify a subset of active contingency scenarios that generate Benders-type feasibility cuts. In our computational study, we use the subset of active contingency scenarios in combination with wind power scenarios.

The rest of the paper is organized as follows. Section II describes the SCUC model and presents a special case of the model. In Section III we present asynchronous parallel dual decomposition methods for SCUC. Section IV presents numerical experiments by using WECC 225-bus test system. We discuss the conclusion and future work in Section V.

II. SECURITY CONSTRAINED UNIT COMMITMENT

The SCUC is formulated as a two-stage SMIP, where the first stage schedules generators and the second stage reschedules fast generators and redispaches power in order to minimize the operation cost while satisfying the feasibility for any given contingency scenarios. The scenarios are defined by a combination of wind scenarios and contingency scenarios. Each contingency represents failure on a transmission line.

A. Nomenclature

We define sets, parameters, and variables that are used for our optimization models. The units are described in brackets, if needed.

<table>
<thead>
<tr>
<th>Sets</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( D_n ) / ( D_n )</td>
<td>set of loads; loads at bus ( n )</td>
</tr>
<tr>
<td>( G_n ) / ( G_n )</td>
<td>set of generators; generators at bus ( n )</td>
</tr>
<tr>
<td>( G_S )</td>
<td>set of slow generators</td>
</tr>
<tr>
<td>( L_+ )</td>
<td>set of transmission lines</td>
</tr>
<tr>
<td>( L_n )</td>
<td>set of transmission lines to bus ( n )</td>
</tr>
<tr>
<td>( L_n )</td>
<td>set of transmission lines from bus ( n )</td>
</tr>
<tr>
<td>( N )</td>
<td>set of buses</td>
</tr>
<tr>
<td>( R_n )</td>
<td>set of non-dispatchable generators at bus ( n )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( B_{ls} )</td>
<td>susceptance at line ( l ) for scenario ( s )</td>
</tr>
<tr>
<td>( C_g )</td>
<td>power generation cost at generator ( g ) [$/MWh]</td>
</tr>
<tr>
<td>( D_{nt} )</td>
<td>load at bus ( n ) at time ( t ) [MW]</td>
</tr>
<tr>
<td>( DT_g )</td>
<td>minimum downtime for generator ( g ) [hour]</td>
</tr>
<tr>
<td>( K_g )</td>
<td>commitment cost of generator ( g ) [$/hour]</td>
</tr>
<tr>
<td>( p_{max} )</td>
<td>maximum generation capacity [MWh]</td>
</tr>
<tr>
<td>( p_{min} )</td>
<td>minimum generation requirement [MWh]</td>
</tr>
<tr>
<td>( R_g )</td>
<td>minimum ramp-down for generator ( g ) [MWh]</td>
</tr>
<tr>
<td>( R_{up} )</td>
<td>minimum ramp-up for generator ( g ) [MWh]</td>
</tr>
<tr>
<td>( M_{gts} )</td>
<td>non-dispatchable power generation at generator ( g ) at time ( t ) for scenario ( s ) [MWh]</td>
</tr>
<tr>
<td>( UT_g )</td>
<td>minimum uptime for generator ( g ) [hour]</td>
</tr>
<tr>
<td>( \rho )</td>
<td>value of load loss [$/MWh]</td>
</tr>
<tr>
<td>( \pi_s )</td>
<td>probability of scenario ( s )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{lts} )</td>
<td>power flow in line ( l ) at time ( t ) for scenario ( s ) [MWh]</td>
</tr>
<tr>
<td>( m_{gts} )</td>
<td>power spillage at non-dispatchable generator ( g ) at time ( t ) for scenario ( s ) [MWh]</td>
</tr>
<tr>
<td>( p_{gts} )</td>
<td>power generation at generator ( g ) at time ( t ) for scenario ( s ) [MWh]</td>
</tr>
<tr>
<td>( u_{gts} )</td>
<td>commitment at generator ( g ) at time ( t ) for scenario ( s )</td>
</tr>
<tr>
<td>( v_{gts} )</td>
<td>startup at generator ( g ) at time ( t ) for scenario ( s )</td>
</tr>
<tr>
<td>( z_l )</td>
<td>1 if failure in line ( l ); 0 otherwise</td>
</tr>
<tr>
<td>( \epsilon_s )</td>
<td>fraction of the system load that allows shed</td>
</tr>
<tr>
<td>( \theta_{nts} )</td>
<td>voltage angle at bus ( n ) at time ( t ) for scenario ( s )</td>
</tr>
</tbody>
</table>

B. Formulation

We present the two-stage SMIP formulation of SCUC model that uses the contingency scenarios as well as uncertain wind power generation. We consider contingency scenarios to represent single transmission line failures. While adopting the SCUC formulation used in [2], we introduce slack variable \( \epsilon_s \) in the first stage for precluding infeasible models due to forecast data. In particular, using our slack variable in combination with a large penalty cost \( \rho \) enforces that at least \((1 - \epsilon_s)\) fraction of the total system load must be served, as used for an \( n-K-\epsilon \) reliability criterion in [4].

Our SCUC model is formulated by

\[
\begin{align*}
\min & \sum_{s \in S} \pi_s \left[ \sum_{g \in G} \sum_{t \in T} (K_g u_{gts} + S_g v_{gts} + C_g p_{gts}) + \rho \epsilon_s \right] \\
\text{s.t.} & \quad u_{gts} = u_{gts, s-1}, \forall g \in G, t \in T, s \in S, \quad \epsilon_s = \epsilon_{s-1}, \forall s \in S, \\
& \quad v_{gts} \geq u_{gts} - u_{gts, s-1}, \forall g \in G, t \in T, s \in S, \\
& \quad \sum_{q=t-UT_s+1}^{t} v_{gqs} \leq u_{gts}, \forall g \in G, t \in \{UT_s, \ldots, T\}, s \in S,
\end{align*}
\]
We assume that includes generator startup and commitment cost, power π_m to the model. We also assume that non-dispatchable power constraints are not imposed on variable constraints (1b) and (1c). Note that the nonanticipativity constraint has the same values for all the scenarios by nonanticipativity shedding allowance. Hence, these variables are enforced to the commitment decisions of the slow generators and the load-shedding allowance. Equation (1g) represents the network topology and enforces the balance between the power supply and demand after any load shedding and power spillage in the system. Constraint (1h) allows the pre-defined amount of total system load lost. Constraint (1i) represents the (linearized) power flow equation for each contingency scenario, where line susceptance is defined as \( B_{ls} := 0 \) if line \( l \) is tripped for scenario \( s \). Constraint (1j) enforces the minimum and maximum generation capacity. Equation (1k) represents the ramping constraint. Bound constraints (1l)–(1q) are defined for each variable.

Problem (1) is challenging to solve because of the large number of binary variables \(|\mathcal{G}| \cdot |\mathcal{T}| \cdot |\mathcal{S}|\) in combination with continuous variables. Moreover, the binary variables appear in both the first and second stages. One can formulate the problem (1) with respect to the first-stage variables only, where the recourse function is introduced to represent the second-stage problem parameterized by the first-stage variable value. Note, however, that the recourse function is non-convex in the first-stage variable because of the binary variables in the second stage.

### C. Deterministic Wind Power Generation

We present a special case of the SCUC model (1), which considers a number of contingency scenarios for a given wind power generation (i.e., \( M_{\text{gts}} = M_{\text{gt},s-1} \) for \( s \in \mathcal{S} \)). We also assume that the model fixes \( \epsilon_s = 0 \) for \( s \in \mathcal{S} \). The resulting SMIP problem has pure-binary variables in the first stage and can be simplified as follows

\[
\min_{x,y} \quad c^T x + d^T y \quad \text{(2a)} \\
\text{s.t.} \quad (x, y_0) \in X_0, \quad (x, y_s) \in X_s, \quad \forall s \in S^C := S \setminus \{0\}, \quad \text{(2b)}
\]

where \( x \) is the first-stage binary variable and \( y_s \) are the second-stage mixed-binary variables for each scenario \( s \in \mathcal{S} \). In particular, scenario 0 represents the non-contingency state and the other scenarios represent the contingency states. The objective function (2a) is obtained by the assumption that \( \pi_0 = 1 \) and \( \pi_s = 0 \) for \( s \in S^C \). Throughout the section, we assume that the problem (2) is feasible and bounded.

We observe that the SCUC model with a deterministic wind power generation is significantly easier than the stochastic counterpart and can be solved by a Benders decomposition method. We develop a Benders decomposition method for solving the non-contingency scenario problem with a number of contingency scenario subproblems for checking feasibility. Note, however, that this is not true if we have more than one non-contingency scenarios (i.e., more than one wind scenario), in which case the method requires to generate optimality cuts.

The feasibility cut can be generated to exclude a binary variable value. Suppose that \((\hat{x}, \hat{y}_0) \in X_0\) and that there is no \( y_s \) such that \((\hat{x}, y_s) \in X_s\) for some \( s \in S^C \). We can exclude \((\hat{x}, \cdot)\) from \( X_0 \) by adding feasibility cut

\[
\sum_{j: \hat{x}_j = 1} x_j + \sum_{j: \hat{x}_j = 0} (1 - x_j) \leq n^r - 1, \quad \text{(3)}
\]
where \( x_j \) is the \( j \)th element of binary vector variable \( x \) and \( n^2 \) is the dimension of binary vector \( x \). Note that only a finite number of cuts (3) can be generated.

**Algorithm 1 Benders Decomposition Method**

1. Initialize \( X_0^k \leftarrow X_0 \) and \( k \leftarrow 0 \).
2. Find \((x^k, y^k) \in \arg \min_{(x,y) \in X^k_0} \{ c^T x + d^T y \} \).
3. while \( \exists s \in S^c \) such that \( \{ y : (x, y) \in X_s \} = \emptyset \) do
4. \( X^k_0 \leftarrow X^k_0 \cap \{ x : (3) \text{ generated at } x^k \} \).
5. \( k \leftarrow k + 1 \).
6. Find \((x^k, y^k) \in \arg \min_{(x,y) \in X^k_0} \{ c^T x + d^T y \} \).
7. end while

The Benders decomposition method is summarized in Algorithm 1. The algorithm initializes the set of feasible solutions for the non-contingency scenario (line 1). The Benders master problem is solved to find a candidate solution (lines 2 and 7). If the solution is violated by a contingency scenario, it is excluded from the solution set by adding the feasibility cut (3) (lines 3 and 4). Note that the feasibility check in line 3 can be parallelized for each scenario. Since there exists a finite number of binary variable values in the first stage, the algorithm terminates after generating a finite number of feasibility cuts.

**III. PARALLEL DUAL DECOMPOSITION METHODS**

In this section, we present parallel dual decomposition (DD) methods for solving the Lagrangian dual of SMIP by relaxing the nonanticipativity constraints.

A. **Dual Decomposition**

We consider the simplified formulation of model (1) as

\[
\min_{x, y} \sum_{s \in S} \pi_s (c^T x_s + d^T y_s) \tag{4a}
\]

subject to

\[
\sum_{s \in S} H_s x_s = 0, \tag{4b}
\]

\[
(x_s, y_s) \in X_s, s \in S, \tag{4c}
\]

where (4b) represents the nonanticipativity constraints (1b) and (1c) and set \( X_s \) is the set of feasible solutions \((x_s, y_s)\) that represents all the other constraints (1d) – (1r) for each scenario. Note also that both \( x_s \) and \( y_s \) are mixed-binary variables.

DD is obtained by the Lagrangian relaxation of the nonanticipativity constraints (4b) to the objective function. The goal of this approach is to find the best Lagrangian dual bound by solving

\[
z_{LD} := \max_\lambda \sum_{s \in S} D_s(\lambda), \tag{5}
\]

where for \( s \in S \) the Lagrangian dual function is defined as

\[
D_s(\lambda) := \min_{(x,y) \in X_s} \pi_s (c^T x + d^T y) - \lambda^T H_s x \tag{6}
\]

and \( \lambda \) is the Lagrangian multiplier for the nonanticipativity constraints (4b). We call problem (6) the Lagrangian subproblem. We also define \( D(\lambda) := \sum_{s \in S} D_s(\lambda) \).

B. **Asynchronous Bundle Method with Trust Region**

We present a bundle method for solving the Lagrangian dual problem (5). In particular, a TR constraint is added in order to regularize the search space for the dual variable \( \lambda \). The bundle method solves problem (5) by approximating the Lagrangian dual function \( D_s(\lambda) \) with a set of linear inequalities. Let \( k \) and \( l \) be the index of major and minor iterations, respectively. The major iteration updates the best dual bound, whereas the minor iteration approximates the Lagrangian dual problem.

We first define the model function

\[
\tilde{m}_{k,l}(\lambda) := \max_{\theta_s} \sum_{s \in S} \theta_s \tag{7a}
\]

subject to

\[
\theta_s \leq D_s(\lambda^l) + (H_s x_s^l)^T (\lambda - \lambda^l), \qquad i \in B^{k,l}, s \in S^l, \tag{7b}
\]

where linear inequalities (7b) outer-approximate the Lagrangian dual function \( D_s(\lambda) \). If the solution is violated by a contingency scenario, it is excluded from the solution set by adding the feasibility cut (3) (lines 3 and 4). Note that the feasibility check in line 3 can be parallelized for each scenario. Since there exists a finite number of binary variable values in the first stage, the algorithm terminates after generating a finite number of feasibility cuts.

The asynchronous algorithm is summarized in Algorithm 2. The algorithm initializes with an initial trial point \( \lambda^0 \) and parameters \( \epsilon \geq 0, D_{0,0} \in [0, \bar{D}], \xi \in (0, 0.5), \Lambda > 0, \Pi > 0, \bar{J} \subseteq J \) for \( \pi \in \Pi \). Note that Algorithm 2 becomes the synchronous counterpart when \( \Pi = |\Pi| \). For the initial trial
Algorithm 2 Asynchronous Bundle-Trust-Region Method

1: Initialize $\lambda^0$ and parameters, and set $k \leftarrow 0$ and $l \leftarrow 0$.
2: Evaluate $D_s(\lambda^0)$ for $s \in S$.
3: Initialize the model function $\tilde{m}_{0,0}$.
4: loop
5: Solve the master (8) to find $\lambda^{k,l}$.
6: if $\tilde{m}_{k,j}(\lambda^{k,l}) - D(\lambda^{k}) \leq \epsilon(1 + |D(\lambda^{k})|)$ then
7: Stop
8: end if
9: if $|\Lambda_{k,j}| < \bar{\lambda}$ then
10: $\Lambda_{k,j}^R(\pi) \leftarrow \Lambda_{k,j}^R(\pi) \cup \{\lambda^{k,l}\}$ for each $\pi \in \Pi$.
11: end if
12: for $\pi \in \Pi_{k,l}$ do
13: if $\Lambda_{k,j}^R(\pi) = \emptyset$ then
14: $\Lambda_{k,j} \leftarrow \Lambda_{k,j}\{\lambda^{k,l}\}$
15: end if
16: Choose the first element $\lambda^g \in \Lambda_{k,j}^R(\pi)$.
17: $\Lambda_{k,j} \leftarrow \Lambda_{k,j}\{\lambda^{k,l}\}$
18: end if
19: end for
20: repeat
21: RECEIVE $D_f(\lambda^0)$ and $x_j^g$ for $j \in J_\pi$ from $\pi$.
22: Evaluate $x_j^g$ for upper bounds.
23: $\Pi_{k,l} \leftarrow \Pi_{k,l} \cup \{\pi\}$.
24: until $|\Pi_{k,l}| \geq \Gamma$
25: serious $\leftarrow$ false.
26: if $\cap_{\pi \in \Pi} \Lambda_{k,j}^C(\pi) \neq \emptyset$ then
27: $\lambda^{k,l} \leftarrow \arg \max \{D(\lambda) : \lambda \in \cap_{\pi \in \Pi} \Lambda_{k,j}^C(\pi)\}$
28: $\Lambda_{k,j} \leftarrow \Lambda_{k,j}\{\lambda^{k,l}\}$
29: if descent test (9) holds then
30: Choose $\Delta_{k+1,0} \in [\Delta_{k,l}, \Delta^{\max}]$.
31: Choose $\Lambda_{k+1,0} \subseteq \Lambda_{k,l}$.
32: Set $\lambda^{k+1} \leftarrow \tilde{\lambda}^{k,l} , m_{k+1,0} \leftarrow \tilde{m}_{k,l} , \Pi_{k+1,0} \leftarrow \Pi_{k,l} , k \leftarrow k + 1$ and $l \leftarrow 0$.
33: serious $\leftarrow$ true.
34: else
35: Choose $\Delta_{k+l+1} \in (0, \Delta_{k,l}]$.
36: end if
37: end if
38: if serious $= false$ then
39: Update $\tilde{m}_{k,l+1}$ by adding cuts (7b).
40: $\Pi_{k,l+1} \leftarrow \Pi_{k,l} , \Lambda_{k,l+1} \leftarrow \Lambda_{k,l} , l \leftarrow l + 1$.
41: end if
42: end loop

point $\lambda^0$, the Lagrangian subproblems are solved in parallel (line 2) and synchronized to initialize the model function with the initial bundle information (line 3). The master problem is solved to find a new trial point in line 5. The algorithm terminates (line 7) if the model function value is close to the best dual bound $D(\lambda^k)$ within the gap of $\epsilon(1 + |D(\lambda^k)|)$ in line 6. If the queue is not full, the new trial point is stored to the queue and ready for evaluation (lines 9–11). In lines 12–24, the master process asynchronously sends and receives necessary data with some subproblem processes. The master process sends each available process the new trial point (line 14) or the first element of the dual variables ready for evaluation (line 16–17). Then, the master receives bundle information from at least $\Pi$ processes (lines 21–24). The primal solution $x_j^g$ may be evaluated to obtain upper bounds, which can also be done on separate processes in parallel. The descent test (9) is performed only for the dual variable evaluated by all the scenarios (lines 26–37). Otherwise, the model function is updated (lines 38–41). For $\epsilon > 0$, Algorithm 2 terminates after a finite number of steps, as shown in [8].

IV. Computational Study

In this section, we present the numerical results by using large-scale test problem instances of the SCUC model. We implemented the Benders decomposition (Algorithm 1) in Julia [9] by using CPLEX callback function and MPI library. For the DD methods (Algorithm 2 and the synchronous counterpart) we use a parallel decomposition solver DSP [7], which can read an optimization model from Julia and run on a high-performance computing clusters by using the MPI library. The master problem and Lagrangian scenario subproblems were solved by using CPLEX-12.7 with the relative gap tolerance of $10^{-5}$. In particular, the master problem was solved by the barrier optimizer in CPLEX. We have modeled the CCUC problem (1) with the JuMP modeling package [10] in Julia. All experiments were run on the 1024-node computing cluster Bepop at Argonne National Laboratory.

A. Problem Instances

We use the test data that represents the California Independent System Operator system interconnected to the Western Electricity Coordinating Council (WECC), as also used in [2]. The WECC test system consists of 225 buses, 375 transmission lines, 130 generation units, 40 loads, 5 import points, 5 wind farms, and 11 non-wind renewable generation units. We set 90 of the 130 generators to be capable of starting in response to a contingency event (i.e., fast generators). We consider a 24-hour time horizon with hourly time intervals. As a result, each problem instance has 1,000 binary variables, one continuous variable, and no constraint in the first stage and 8,750 binary variables, 15,734 continuous variables, and 35,149 constraints in the second stage for each scenario.

Eight problem instances were generated for each of spring, summer, fall, and winter and for each weekday (WD) and weekend (WE). The total net load for each time $t \in T$ is shown in Figure 1 for each problem instance. Moreover, as modeled in (1), we allow predetermined amount of load lost at a penalty cost $\rho = 1,000 \sum_{j \in D} \sum_{t \in T} D_{jt} \frac{s}{\text{per fraction}}$, which is equivalent to $1,000$ per MWh of load lost.

We consider each contingency event as a single transmission line failure. However, we assume that 6 of the 375 lines are protected and invulnerable, as removing each of the 6 lines creates an island of the system. This results in 370 contingency scenarios. In addition, we consider three wind power
We present the computational results from the Benders decomposition (Algorithm 1) for solving the SCUC model with the 370 contingency scenarios for a given wind power generation, as described in Section II-C. In particular, we aim to find a list of contingency scenarios (i.e., active generation, as described in Section II-C. In particular, we aim with the 370 contingency scenarios for a given wind power decomposition (Algorithm 1) for solving the SCUC model.

### B. Deterministic Wind Power Generation

We assume that each scenario has equal probability. Wind power scenarios of mild, medium, and high penetration levels with 43,162, 76,290, and 100,169 MW, respectively. We assume that each scenario has equal probability.

#### TABLE I
**COMPUTATIONAL RESULTS OF THE BENDERS DECOMPOSITION METHOD**

<table>
<thead>
<tr>
<th>Instance</th>
<th>Objective</th>
<th>Active Contingencies</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SpringWD</td>
<td>1654734</td>
<td>L321, L326, L352, L353</td>
<td>464</td>
</tr>
<tr>
<td>SpringWE</td>
<td>1143749</td>
<td>L326, L352, L353</td>
<td>1151</td>
</tr>
<tr>
<td>SummerWD</td>
<td>4890148</td>
<td>L216, L296, L321, L322</td>
<td>630</td>
</tr>
<tr>
<td>SummerWE</td>
<td>3300184</td>
<td>L321, L326, L352, L353</td>
<td>542</td>
</tr>
<tr>
<td>FallWD</td>
<td>2492479</td>
<td>L321, L326, L352, L353</td>
<td>631</td>
</tr>
<tr>
<td>FallWE</td>
<td>1589561</td>
<td>L321, L326, L352, L353</td>
<td>540</td>
</tr>
<tr>
<td>WinterWD</td>
<td>1947431</td>
<td>L321, L326, L352, L353</td>
<td>559</td>
</tr>
<tr>
<td>WinterWE</td>
<td>1346567</td>
<td>L321, L326, L352, L353</td>
<td>435</td>
</tr>
</tbody>
</table>

#### C. Synchronous Dual Decomposition

We present the computational results from the synchronous DD for the problem instances. For the numerical experiments of the DD methods, we use the combination of the wind power generation scenarios and the active contingency scenarios L321, L326, L352, and L353, as reported in Section IV-B, resulting in 15 scenarios for each problem instance. Each problem instance was solved by using 36 cores of the cluster with 10 hours of time limit. 15 cores were used for the scenario subproblems, another 15 cores were for upper bounding, and the other cores were used for solving the master.

We use the percent imbalance metric, as defined in [11], for quantifying the load imbalance of solving the Lagrangian subproblems. The percent imbalance metric $\nu_k$ for each iteration $k$ is given by $\nu_k := (t_k^{\max} / t_k) \times 100\%$, where $t_k^{\max}$ and $t_k$ are the maximum and mean subproblem solution times, respectively, for each iteration $k$.

Table II reports the computational results from the synchronous counterpart of Algorithm 2 by setting $\Pi = |\Pi|$, for solving each problem instance. None of the problem instances found optimal Lagrangian dual bounds in the 10-hour time limit. The best upper and lower bounds ($\times 10^3$) and duality gaps are reported in the table. The maximum, minimum, and mean percent imbalance metrics are also reported in columns $\nu^{\max}$, $\nu^{\min}$, and $\bar{\nu}$, respectively. For all the problem instances, every iteration suffers from the load imbalance of at least 20%. The maximum imbalance was 584% for the SummerWE instance.

![Figure 2](image-url) **Figure 2** A heatmap showing the Lagrangian subproblem solution time for each iteration and for each worker process, which visualizes the seriousness of load imbalances among the processes. For the highly imbalanced SummerWE instance in Figure 2b, processes 4, 9, and 14 were almost always significant bottlenecks of the DD. These processes were assigned to solve the scenario subproblems with nonzero probabilities, of which the objective functions are more complicated than those of the other scenario subproblems.

#### D. Asynchronous Dual Decomposition

We present the computational performance of the asynchronous DD (Algorithm 2) for the same problem instances, as used in Section IV-C. Table III reports the computational results and relative speedup for each instance. The speedup was calculated by 10 hours (of the time limit) divided by the
In the computational experiments using the WECC 225-bus test system, we found that the asynchronous method can significantly accelerate the solution time and also find tighter upper and lower bounds, as compared with the synchronous counterpart. In addition, we found that the SCUC model is much more difficult (i.e., it requires more iterations and wall clock time) than the stochastic unit commitment with uncertain wind generation (see [8] using the same problem instances). In particular, the latter is a SMIP with random right-hand sides only, whereas the former is a SMIP with both random matrices and right-hand sides.

Future work of interest includes the development of effective algorithms for solving SMIP problems with both random matrices and right-hand sides, particularly aiming to reduce the number of iterations of the DD method.

**V. Conclusions and Future Work**

We presented a parallel, asynchronous DD method in application to solving SCUC model with contingency events on system components in combination with uncertain (wind power) generation forecast. Adopting the SMIP formulation of the SCUC model used in [2], we also introduce a slack variable to avoid any infeasible problem instances due to the problem data, representing a predetermined amount of total system load lost, as used in [4]. We also presented a special case of the SCUC model that considers a deterministic power generation with contingency scenarios, which allows us to develop a parallel Benders decomposition. In particular, this model was used to find active contingency scenarios for feasibility of the SCUC model solution.

We applied a recently-developed asynchronous DD method for efficiently solving the problem in parallel. In the computational experiments using the WECC 225-bus test system, we found that the asynchronous method can significantly accelerate the solution time and also find tighter upper and lower bounds, as compared with the synchronous counterpart. In addition, we found that the SCUC model is much more difficult (i.e., it requires more iterations and wall clock time) than the stochastic unit commitment with uncertain wind generation (see [8] using the same problem instances). In particular, the latter is a SMIP with random right-hand sides only, whereas the former is a SMIP with both random matrices and right-hand sides.

Future work of interest includes the development of effective algorithms for solving SMIP problems with both random matrices and right-hand sides, particularly aiming to reduce the number of iterations of the DD method.

**References**
