Adaptive Robust Optimization for Daily Power System Operation

Andy Sun, Member, Alvaro Lorca, Student Member

Abstract—Robust optimization, as a powerful paradigm for optimization under uncertainty, has recently attracted increasing attention in power system operations. In particular, adaptive robust optimization models have been proposed for the day-ahead unit commitment as well as real-time economic dispatch problems in power systems with a high penetration level of renewable energy sources. In this paper, we review the recent advances in this area and propose some new ideas in uncertainty modeling and robust optimization formulations.

Index Terms—Unit commitment, economic dispatch, robust optimization, uncertainty sets, mixed-integer optimization.

I. INTRODUCTION

With the increasing penetration of renewable energy resources, power system operators face new challenges in managing significant generation uncertainty in day-ahead and real-time markets. One of the most critical operational problems facing such challenges is the unit commitment (UC) problem. Great attention has recently been drawn to design new optimization models and algorithms for the UC problem that produce both high economic efficiency and system reliability under increasing uncertainty.

In the current industry practice, the UC problem is modelled and solved as a large-scale deterministic mixed-integer program (MIP). The commitment status of generating units are scheduled to balance the net load forecasted for the next day. Renewable resources such as wind and solar power are intrinsically stochastic and intermittent. As a traditional approach to deal with uncertainties in load forecast and contingencies, system operators require additional generating resources, called reserves, to be online or quickly available [1]. The system operator can also adjust the level of reserves according to the system operating status.

In the recent work [2], we propose a two-stage adaptive robust optimization model and solution methods for the security-constrained unit commitment problem. Computational experiments on the power system operated by the ISO New England demonstrate that the robust UC model can significantly reduce operating costs and improve system reliability, comparing to the widely used deterministic UC model with adjustable reserves. Related models have been studied in several papers, e.g. [3]–[7]. For a recent review, see [8]. A significant extension of the two-stage robust model to a general multistage adaptive robust UC model is proposed and systematically studied in [9]. New data-driven approaches for constructing uncertainty sets to model dynamic relationships between uncertainties and decisions are proposed in [10] and [11].

The focus of this paper is to provide deeper exploration on several important issues in the modeling and solution methods of the two-stage robust UC and economic dispatch (ED) formulation. In particular, Section II presents the robust UC model. Section III proposes a general budget uncertainty model and compares exact and heuristic methods for solving the second-stage problem. Section IV explores the properties of the worst case scenarios identified by the robust model. Section V introduces dynamic uncertainty sets and studies worst case scenarios of wind power in a two-stage robust economic dispatch problem. Section VI concludes the paper.

II. TWO-STAGE ADAPTIVE ROBUST UNIT COMMITMENT

The two-stage adaptive robust UC model is formulated as follows:

\[
\min_{x \in X} \left\{ c^T x + \max_{d \in D} \min_{y \in Y(x,d)} b^T y \right\}.
\]

In this formulation, \( X \) is the feasible region for the unit commitment variables \((x^o, x^+, x^-)\):

\[
X = \left\{ (x^o, x^+, x^-) \in \{0, 1\}^{3N_gT} : \begin{align*}
    x^o_{i,t-1} - x^o_{i,t} &= x^+_i - x^-_i \quad \forall i \in N_g, t \in T, \\
    \sum_{\tau=t}^{t+UT_i-1} x^o_{i,\tau} &\geq UT_i x^+_i \quad \forall i \in N_g, \\
    \sum_{\tau=t}^{t+DT_i-1} (1 - x^o_{i,\tau}) &\geq DT_i x^-_i \quad \forall i \in N_g, \\
    \sum_{\tau=t}^{T} (x^o_{i,\tau} - x^+_i) &\geq 0 \quad \forall i \in N_g, \\
    \sum_{\tau=t}^{T} (1 - x^o_{i,\tau} - x^-_i) &\geq 0 \quad \forall i \in N_g, \\
    x^o_{i,t} &= 0, \quad \sum_{t=1}^{L_i} x^o_{i,t} = L_i \quad \forall i \in N_g,
\end{align*} \right\}
\]

where \( x^o_{i,t} \) is the on/off commitment status of generator \( i \) at time \( t \), \( x^+_i \) is the start-up decision and \( x^-_i \) is the shut-down decision.

Andy Sun and Alvaro Lorca are with the H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA.
decision. Constraint (2a) is the logic relationship between the \( x \) variables. Constraints (2b)-(2e) are the minimum up and down time constraints, together with boundary conditions (2f). \( N_g \) is the number of generators and \( T \) is the length of the horizon. See [12] for details of the above UC formulation.

The set \( Y(x,d) \) in (1) defines the feasible region for the economic dispatch problem with fixed commitment decision \( x \) and uncertain net load \( d \).

\[
Y(x,d) = \left\{ (p,q) : \begin{array}{l}
 p_{it}^{\min} x_{it}^o \leq p_{it} \leq p_{it}^{\max} x_{it}^o, \quad \forall i, t,
 
 p_{it} - p_{it-1} \geq -RD_i x_{it}^o - SD_i x_{it}^o, \quad \forall i, t,
 
 p_{it} - p_{it-1} \leq RU_i x_{it}^{o,t-1} + SU_i x_{it}^{\alpha}, \quad \forall i, t,
 
 \sum_{i \in N_g} \alpha_i^p p_{it} - \sum_{j \in N_d} \alpha_j^d q_{jt} - q_{it}^l \geq -f_{l_{\max}}, \quad \forall l, t,
 
 \sum_{i \in N_g} \alpha_i^p p_{it} - \sum_{j \in N_d} \alpha_j^d q_{jt} - q_{it}^l \leq f_{l_{\max}}, \quad \forall l, t,
 
 \sum_{i \in N_g} p_{it} + q_{it}^o - q_{it}^\alpha = \sum_{j \in N_d} d_{jt}, \quad \forall t,
 
 q_{it}^l, q_{it}^\alpha, q_{it}^v \geq 0, \quad \forall l, t,\end{array}\right\}, \quad (3g)
\]

where (3a) is the bound on production levels, (3b)-(3c) are ramping up and down constraints (notice that we distinguish between the normal ramping rates \( RU_i, RD_i \) from the starting-up or shutting-down ramping rates \( SU_i, SD_i \)), (3d)-(3e) are transmission line thermal constraints using DC power flows, and (3f) is the energy balance constraint for each hour. We have introduced penalty variables \( q_{it}^l \) and \( q_{it}^\alpha \) in the transmission and energy balance constraints, respectively. In this way, the second-stage max-min problem is guaranteed to be feasible and penalty cost is a measure of infeasibility. This complete recourse property is also important when we formulate the second-stage problem as a MIP, as will be shown later.

The objective function is assumed to be linear or piecewise linear in the commitment and dispatch variables. Linear cost function is used here for simple exposition. Penalty costs for violations of transmission and energy balance constraints are also considered.

\[
e^T x = \sum_{i \in N_g} \sum_{t \in T} \left( c_{it}^o x_{it}^o + c_{it}^\alpha x_{it}^\alpha + c_{it}^v x_{it}^v \right), \quad (4a)
\]

\[
b^T y = \sum_{i \in N_g} \sum_{t \in T} C_{it} p_{it} + \sum_{i \in T} \left( C^+ q_{it}^\alpha + C^- q_{it}^\alpha \right)
+ \sum_{i \in N_g, t \in T} C_{it}^f q_{it}^f. \quad (4b)
\]

The formulation (1) is a two-stage adaptive robust optimization model. The first-stage decision is the commitment decision \( x \), which has to be made before the uncertain net load is realized. The second-stage decision is the dispatch decision \( y \), which fully adapts to the observed realization of net load \( d \). That is, the second-stage problem produces a solution \( y(d) \) that is optimal to each specific realization of the uncertainty \( d \).

This two-stage adaptive robust UC model has become a fundamental model in the robust UC literature. It explicitly formulates the operational procedure of the unit commitment phase and the real-time dispatch phase into two stages, and models uncertainty of net load using an uncertainty set \( D \). It turns out that solving this two-stage robust UC model heavily depends on the structure of the uncertainty set. In the following, we first discuss solution algorithms to solve the two-stage robust model, then we provide a new exact formulation for a general budget uncertainty set, and compare the exact method and a heuristic for solving the second-stage problem.

III. SOLUTION ALGORITHMS: EXACT VERSUS HEURISTICS

The robust UC model (1) can be rewritten as:

\[
\min_{x \in X} c^T x + Q(x), \quad (5)
\]

where the second-stage cost function \( Q(x) \) is given by

\[
Q(x) = \max_{d \in D} \min_{y \in Y(x,d)} b^T y. \quad (6)
\]

From the structure of the dispatch problem (3), we can see that the feasible set \( Y(x,d) \) of the inner minimization problem is a polyhedron in \( y \) of the following compact form:

\[
By \geq g + Ax + Cd.
\]

In other words, the parameters \( x \) and \( d \) only appear in the right-hand side of the linear constraints in (3). This is a key feature of (6) and allows it to be reformulated as a maximization problem. In particular, if we take the dual of the inner minimization problem in (6), we have

\[
Q(x) = \max_{d \in \pi} \pi^T Cd + \pi^T (g + Ax) \quad (7)
\]

s.t. \( d \in D \)

\[
\pi \in P := \{ \pi^T B = b^T, \pi \geq 0 \}.
\]

Since the second-stage problem has complete recourse, the inner minimization is always feasible and achieves its optimum, therefore (6) and (7) are equivalent by linear programming strong duality. Now, problem (7) has a bilinear objective function in variables \( d \) and \( \pi \); further, the constraints on these two groups of variables are disjoint. We have the following two properties of (7):

1) There exists an optimal solution \((d^*, \pi^*)\) of (7), where \(d^*\) and \(\pi^*\) are an extreme point of \(D\) and \(P\), respectively. Note that this extreme point property holds even if \(D\) is a non-polyhedral convex set, e.g. an ellipsoid.

2) \(Q(x)\) is a convex function of \(x\). If the uncertainty set \(D\) is a polyhedron, then \(Q(x)\) is a piecewise linear convex function.

With these properties at hand, we can design an algorithm to solve the overall problem (5), which also involves evaluating \(Q(x)\) for any given \(x\). The fact that \(Q(x)\) is a convex function of \(x\) suggests that \(Q(x)\) can be approximated from below by linear functions, i.e., a two-level algorithmic framework can be developed, where the first level solves the master problem of the form

\[
\min_{x \in X} c^T x + Q(x),
\]
where \(Q(x)\) is a piecewise linear lower approximation of \(Q(x)\). The second level evaluates \(Q(x)\) and updates the approximation \(Q(x)\). It turns out that, although \(Q(x)\) is a convex function, evaluating its value for any fixed \(x\) is a hard problem. This is indicated by the bilinear structure of (7).

A. Two-Level Algorithmic Framework

The two-level algorithm for solving the two-stage robust UC problem consists of writing the problem in the following extended form:

\[
\min_{x \in X} c^T x + Q(x) = \min_{x, \eta} c^T x + \eta \\
\text{s.t. } x \in X \\
\eta \geq \max_{d \in D} \min_{y \in Y(x,d)} b^T y
\]

\[
= \min_{x, \eta} c^T x + \eta \\
\text{s.t. } x \in X \\
\eta \geq \min_{y \in Y(x,d_m)} b^T y \forall d \in D
\]

\[
= \min_{x, \eta} c^T x + \eta \\
\text{s.t. } x \in X \\
\eta \geq \min_{y \in Y(x,d_m)} b^T y \forall m \in M
\]

where the third equality follows from the extreme point property discussed in the previous section, and the fourth equality follows from the property of minimization. Here, \(\{d_m\}_{m \in M}\) is the set of extreme points of \(D\), which can be infinite or even uncountable, depending on whether \(D\) is a polyhedron or not. The algorithm is convergent for the general case.

The last reformulation (8) suggests that the overall two-stage robust UC problem can be solved by gradually generating extreme points \(d_m\) and the associated constraints (8c)-(8d). Therefore, the master problem \((MP)\) at iteration \(k\) contains a list of identified extreme points \(M_k \subseteq M\):

\[
(MP) \min_{x, \eta, \{y_m\}_{m \in M_k}} c^T x + \eta \\
\text{s.t. } x \in X \\
\eta \geq b^T y_m \forall m \in M_k \\
y_m \in Y(x,d_m) \forall m \in M_k
\]

The subproblem is to identify the worst-case demand scenario for the commitment solution generated by the master problem, denoted as \(x_k\). That is, the subproblem needs to evaluate \(Q(x_k)\) and compare it with the solution \(\eta_k\) from the master problem. This overall two-level procedure, Algorithm 1, has been independently proposed in several works including [2] for solving the robust UC model, [13] for solving a defender-attacker model, and analyzed rigorously in [14].

**Algorithm 1 Constraint and Column Generation Algorithm**

1: **Initialization:** \(k = 0\) and \(M_0 = \emptyset\)
2: for \(k = 1, 2, 3...\) do
3:   Solve \((MP)\) with \(M_k\). Denote solution as \((x_k, \eta_k)\).
4:   Evaluate \(Q(x_k)\) and denote solution as \(d_k\).
5:   if \(Q(x_k) > \eta_k\) then
6:     Update \(M_{k+1} \leftarrow M_k \cup \{d_k\}\)
7:       \(k \leftarrow k + 1\)
8:       Go to Step 3.
9:   else
10:      Terminate.
11:  end if

As we already mentioned, it is difficult to solve the min-max problem (6). In the following section, we will develop an exact reformulation of (6) to a mixed-integer linear program, and compare it with the simple heuristics of alternating direction method proposed in [10].

B. Exact method for solving the subproblem

After taking the dual of the dispatch problem \(\min_{y \in Y(x,d)} b^T y\), the second-stage problem becomes

\[
\max_{d, \pi} \sum_{i \in N, t \in T} \left[ p_{it}^{\text{max}} \eta_{it}^{\text{max}} - p_{it}^{\text{min}} \eta_{it}^{\text{min}} \right]
\]

\[
+ \left( RD_o x_{it}^o + SD_i \pi_{it}^o \right) \eta_{it}^{RD} + \left( RU_o x_{it}^o + SU_i \pi_{it}^o \right) \pi_{it}^{RU}
\]

\[
+ \sum_{i \in N_t, t \in T} \left( f_{it}^{\text{max}} + \sum_{j \in N_d} \alpha_{ij} d_j \right) \pi_{it}^{f+} + \left( f_{it}^{\text{min}} - \sum_{j \in N_d} \alpha_{ij} d_j \right) \pi_{it}^{f-}
\]

\[
+ \sum_{t \in T} d_t \pi_{it}^{bal}
\]

s.t. \(\pi_{it}^{pmax} - \pi_{it}^{pmin} - \pi_{it}^{RD} - \pi_{it}^{RU} - \pi_{it}^{f+} + \pi_{it}^{f-} = C_{it}, \forall i, t, (9b)\)

\(\pi_{it}^{f-} \leq \pi_{it}^{f+} \leq C_f, \forall t \in \mathcal{T}, (9c)\)

\(\pi_{it}^{bal} - \pi_{it}^{pmax} + \pi_{it}^{pmin} + \pi_{it}^{RD} + \pi_{it}^{RU} + \pi_{it}^{f+} + \pi_{it}^{f-} \leq 0, (9e)\)

where \(C^+, C^-\) are the penalty cost of violating energy balance constraints and \(C_f\) is the penalty cost of violating transmission constraints.

Notice that the objective function involves bilinear terms of the form:

\[
\sum_{j \in N_d, t \in T} d_{ij} \eta_j
\]

where \(\eta_j\) is defined as

\[
\eta_j = \pi_{ij}^{bal} + \sum_{i \in N} \alpha_{ij} \left( \pi_{it}^{f+} - \pi_{it}^{f-} \right).
\]
To reformulate the bilinear term (10), we need to use special structures of the uncertainty set \( \mathcal{D} \). Consider the following uncertainty set \( \mathcal{D}^t \) at each time \( t \):

\[
\mathcal{D}^t = \left\{ \mathbf{d}^t = (d_1^t, \ldots, d_{N_d}^t) : \sum_{j \in N_d} |d_j^t - \bar{d}_j^t| / d_j^t \leq \Delta, \quad d_j^t \in [\bar{d}_j^t - \hat{d}_j, \bar{d}_j^t + \hat{d}_j] \quad \forall j \in N_d \right\}.
\]

(12)

This is the so-called budgeted uncertainty set, which is widely used in the robust optimization literature. There are two types of exact reformulations. The first type uses the KKT conditions of the bilinear program, which is generally applicable to any polyhedral uncertainty sets [14]. However, the observed computation performance is quite slow [10]. The second one assumes the budget parameter \( \Delta \) to be integer valued. In this case, the extreme points of \( \mathcal{D}^t \) have the form of \( \bar{d}_j^t \pm \hat{d}_j^t \). Then, the extreme points of \( \mathcal{D}^t \) can be expressed with binary variables [15]. Bilinear terms involving a binary variable and a continuous variable can be easily linearized with the big-M method.

In this paper, we generalize the second approach by allowing \( \Delta \) to be fractional. We have the following result.

**Proposition 1.** For \( \Delta \leq |N_d| \), the set of extreme points of \( \mathcal{D}^t \) is given by

\[
\left\{ \mathbf{d}^t : \exists \mathbf{u}^t, \mathbf{v}^t, \mathbf{w}^t, \mathbf{p}^t \text{ s.t.} \right. \]

\[
d_j^t = \bar{d}_j^t + \hat{d}_j^t (\bar{v}_j^t - \bar{u}_j^t) + (\Delta - |\Delta|) \hat{d}_j^t (\bar{v}_j^t - \bar{u}_j^t) \quad \forall j \in N_d
\]

\[
\sum_{j \in N_d} u_j^t + v_j^t = |\Delta|
\]

\[
\sum_{j \in N_d} u_j^t + v_j^t = 1
\]

\[
l_j^t + u_j^t + v_j^t + \pi_j^t \leq 1 \quad \forall j \in N_d
\]

\[
u_j^t, u_j^t, v_j^t, \pi_j^t \in \{0, 1\} \quad \forall j \in N_d.
\]

(13a)

(13b)

(13c)

(13d)

(13e)

(13f)

Using this characterization of the extreme points of \( \mathcal{D}^t \), each bilinear term in (10) involves products of \( \bar{v}_j^t n_j^t, u_j^t n_j^t, v_j^t n_j^t, u_j^t n_j^t \), where \( n_j^t \) is defined in (11). These bilinear terms can be linearized. Introduce new continuous variables \( \bar{u}_j^t, \bar{v}_j^t, \bar{p}_j^t, \bar{\rho}_j^t \), we have the following linearized terms:

\[-M_j \bar{u}_j^t \leq \bar{u}_j^t \leq M_j \bar{v}_j^t,
\]

(14a)

\[-M_j (1 - \bar{u}_j^t) \leq n_j^t - \bar{v}_j^t \leq M_j (1 - \bar{u}_j^t),
\]

(14b)

\[-M_j \bar{u}_j^t \leq \bar{p}_j^t \leq M_j \bar{u}_j^t,
\]

(14c)

\[-M_j (1 - \bar{u}_j^t) \leq n_j^t - \bar{p}_j^t \leq M_j (1 - \bar{u}_j^t),
\]

(14d)

where (14a)-(14b) are the linearization of \( \bar{v}_j^t n_j^t \), (14c)-(14d) are the linearization of \( \bar{v}_j^t n_j^t \), and similarly we can linearize \( \bar{u}_j^t n_j^t \) and \( \bar{u}_j^t n_j^t \). Then, (10) is reformulated as

\[
\sum_{j \in N_d, t \in T} \left[ \bar{v}_j^t n_j^t + \bar{p}_j^t (\bar{v}_j^t - \bar{p}_j^t) + (\Delta - |\Delta|) \hat{d}_j^t (\bar{v}_j^t - \bar{p}_j^t) \right].
\]

(15)

Here \( M_j \) is a positive constant big enough to bound the dual variables as in (14). It is important to choose a value of \( M_j \) tight as possible. From (11), (9e), (9d) and (9e) we have that

\[-C^- - C^f \sum_{i \in N_t} \left\lfloor \alpha_i^d \right\rfloor \leq \eta_d^j \leq C^+ + C^f \sum_{i \in N_t} \left\lfloor \alpha_i^d \right\rfloor
\]

must hold. Hence, we can take \( M_j = \max \{C^-, C^+\} + C^f \sum_{i \in N_t} \left\lfloor \alpha_i^d \right\rfloor \) to ensure the formulation is correct.

Using the above linearization, the second-stage problem is equivalent to a mixed-integer linear program. Although solving this MIP can obtain the global optimum, its solution time is usually quite long. In the following, we introduce a simple heuristics that can obtain a feasible solution very quickly. Then we will compare the two methods for the solution times and solution quality.

### C. Simple Heuristics for Solving the Subproblem

An outer approximation algorithm is proposed in [2]. Here, we apply another simple heuristic to solve the disjoint bilinear program, namely the alternating direction (AD) method. It optimizes over \( d \) with \( \pi \) fixed at a given value, then optimizes over \( \pi \) with \( d \) fixed at the optimal solution obtained from the previous iteration, and alternates. Use the compact representation (7), the AD algorithm is outlined below.

**Algorithm 2 Alternating Direction Algorithm**

1: **Initialization:** \( t = 0 \) and \( d_0 \in D \)
2: **repeat**
3: \( \pi_{t+1} = \arg \max_{\pi} \pi^\top C_d + \pi^\top (g + A x) \)
4: \( d_{t+1} = \arg \max_{d \in D} \pi_{t+1}^\top C_d \)
5: \( t \leftarrow t + 1 \)
6: **until** convergence criterion is met.

The heuristic algorithm in general only identifies a KKT point of the bilinear program. Denote the objective value obtained by the AD algorithm as \( Q(x_k) \leq Q(x_k) \). It is possible that \( Q(x_k) \leq Q(x_k) \). In this case, the heuristic method would stop the upper level algorithm and potentially find a suboptimal commitment solution. The above AD algorithm is proposed as a first step heuristics in a cutting plane algorithm for solving general bilinear programs to global optimality [16]. However, the cutting plane algorithm again has the same complexity as the exact method.

### D. Computational Experiments

In this section, we study the performance of the exact method using (13) and (14) and the heuristic method using Algorithm 2. In particular, we want to empirically show how well the exact method performs, how good or bad the heuristics is in solving the second-stage problem, how much it can influence the final solution, and what is the best way to solve the second-stage problem combining the two methods.

We test on the IEEE 30-bus system with nominal demand as is given in [17], [18]. There are 6 generators and 20 loads in the system. The penalty costs \( C^+, C^-, C^f \) are set to be \( $5000/MW. All the codes are implemented in Python and

---

18th Power Systems Computation Conference

Wrocław, Poland – August 18-22, 2014
call CPLEX 12.5. Computation is conducted on a laptop with 2.1GHz CPU and 4GB RAM.

1) Choosing big-M: The value of \( M_j \) gives the upper and lower bounds on the dual variable \( \eta^j \) in (9). Since the primal polytope of the dispatch problem (3) is bounded, by well-known results from LP duality, the dual polytope defined by (9b)-(9e) is unbounded. However, since the primal problem always obtains an optimal solution by adding penalty variables, the dual optimal solution is always bounded. Thus, there exists a finite \( M_j \), and a tight value of \( M_j \) helps speed up the algorithm. Unfortunately, it is not an easy task to pinpoint a tight value for \( M_j \).

For the 30-bus system, we test different values of \( M_j \) at 100, 1000, and 50,000. With \( M_j = 100 \), the exact MIP can be solved to optimality under 1 minute for different levels of budget varying from \( \Delta = 2 \) to 18. However, a closer look at the solution reveals that some dual variables hit the upper or lower bounds, which implies \( M_j = 100 \) is not big enough to bound the optimal dual solution. With \( M_j = 1000 \), the MIP gap cannot be closed below 3% after 10 minutes for \( \Delta \leq 10 \); furthermore, some dual variables still hit the upper and lower bounds. Therefore, \( M_j = 1000 \) is not large enough either, and it already significantly increases the computation time. The bounds on \( \eta^j \) given by (15) amount to 44,950. Therefore, setting \( M_j = 50,000 \) guarantees a valid big-M value. This value might be an over-estimate for \( M_j \), however, it is the only guarantee we have, and the results for \( M_j = 1000 \) suggest that a smaller \( M_j \) may not save much computation time anyway. This tuning procedure is typical for a MIP model with big-M parameters. It is non-trivial to find a tight bound and usually even a tight \( M \) value is already large enough to slow down the convergence of the branch-and-bound procedure.

2) Computation times of solving the second-stage problem: To compare how fast the exact method and the heuristics can solve the second-stage problem, we conduct the following first set of experiments: For budget \( \Delta \in [2, 18] \), we solve the two-stage robust UC using Algorithm 1, where the second-stage problem is solved by the exact method, and then again solved by the heuristics under the same commitment solution.

We observe that CPLEX takes a long time to improve lower bounds when solving the second-stage exact MIP formulation. After 10 min, the optimality gaps are still not much improved from those obtained after 1 min. Therefore, in all the experiments, we terminate the second-stage MIP after 1 min. In comparison, the AD algorithm solving the same set of second-stage problems converges on average within 1.52 seconds.

3) Solution quality: Naturally, the next question is how good the heuristic solutions are. Table I shows the objective values obtained from the exact and heuristic methods for solving the same second-stage problem at each iteration of the two-level algorithm.

For example, with \( \Delta = 2 \), the master problem is solved in 3 iterations by Algorithm 1 with the exact method, i.e., the second-stage problem is also solved 3 times. The objective values obtained by the exact method for these three second-stage problems are [229830, 88633, 88636]; the heuristic AD algorithm obtains [214543, 88636, 88636] for the same set of problems. From the table we can see that the Heuristics obtains the same or better second-stage costs toward the end of the iterations. But in the beginning one or two iterations, the exact method obtains a better second-stage cost (recall the second-stage problem is maximizing.) This property is also observed for other values of \( \Delta \), which suggests that the heuristics works well when the commitment solution is close to optimal, and may get stuck at a bad local solution when the commitment solution is far from optimal, which is the case in the beginning of the master problem iterations.

<table>
<thead>
<tr>
<th>( \Delta )</th>
<th>Exact Method</th>
<th>Heuristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>[229830, 88633, 88636]</td>
<td>[214543, 88636, 88636]</td>
</tr>
<tr>
<td>4</td>
<td>[252040, 90072, 90694, 90707]</td>
<td>[225403, 90707, 90707, 90707]</td>
</tr>
<tr>
<td>6</td>
<td>[264391, 101664, 91835, 91851, 91853]</td>
<td>[231442, 91818, 91863, 91863]</td>
</tr>
<tr>
<td>8</td>
<td>[274884, 116761, 92803]</td>
<td>[236582, 92759, 92804]</td>
</tr>
<tr>
<td>10</td>
<td>[294488, 121328, 93007]</td>
<td>[241288, 93069, 93069]</td>
</tr>
<tr>
<td>12</td>
<td>[292536, 140564, 94450, 94422]</td>
<td>[245136, 94398, 94457, 94422]</td>
</tr>
<tr>
<td>14</td>
<td>[297630, 148011, 94939, 94904]</td>
<td>[247735, 94874, 94939, 94904]</td>
</tr>
<tr>
<td>16</td>
<td>[301445, 155242, 95308, 95273, 95273]</td>
<td>[249608, 95263, 95308, 95273]</td>
</tr>
<tr>
<td>18</td>
<td>[304633, 161472, 95595, 95560]</td>
<td>[251174, 95550, 95595, 95560]</td>
</tr>
</tbody>
</table>

We conduct the second set of experiments, where the second-stage problem is solved using the Heuristics alone. That is, the two-stage problem is solved by Algorithm 1 and Algorithm 2. The extreme points generated by Algorithm 2 can be different from the ones generated by the exact method. Therefore, the commitment solutions obtained would also be different. We observe that the second-stage cost of the final commitment solution obtained using Algorithm 1 with Heuristics is on average 2% lower than that obtained by Algorithm 1 with the exact method. This shows that the solution obtained from Heuristics is not globally optimal, as is expected. However, the gap is quite small.

4) Hybrid method: Combining the results obtained from the first and second sets of experiments, especially the observation that the exact method does well in the initial iterations and the Heuristics can do better in the later iterations, we propose a hybrid method which uses the exact method in the beginning few iterations and then switches to the Heuristics. We conduct the third set of experiments to implement this hybrid approach, where the second-stage problem is solved by the exact methods in the first two iterations of the master problem, then is solved by the Heuristics. For all 9 test cases of \( \Delta = 2, \ldots, 18 \), the hybrid approach converges after 1 iteration of Heuristics, and reaches the same solution as obtained using the exact method.

5) Summary: In this section, we have conducted a detailed study on the performance of the exact method and the Heuristics. The experiments suggest that it is important to obtain good commitment solutions in the beginning iterations of Algorithm 1. Here, it is worth using the exact method to solve the second-stage problem. As the commitment solution is approaching the optimal, it is more economical and does not seem to sacrifice performance to use the Heuristics.

IV. WORST CASE SCENARIO ANALYSIS

Solving the two-stage robust UC model (1) not only produces the robust UC solution, but also identifies the worst-
case net load scenario for the UC solution. These worst case scenarios carry useful information for system security analysis. In this section, we study the worst-case uncertainty scenarios for the budget uncertainty set (12) for different budget levels. The experiments are conducted on the same IEEE 30-bus system, where all of the 20 loads are uncertain with 10% variation, i.e. \( d_j^2 = 0.1d_j^0 \). Table II summarizes the nominal load levels at the peak hour and load distribution factors for the largest 6 loads.

From the previous analysis, we know that the worst-case net load is an extreme point of the uncertainty set \( D \). The question is which extreme point of the uncertainty set will be the worst case. Intuitively, we may expect the largest net load in the uncertainty set to be the worst case, because product cost increases with net load. This is indeed the case when the transmission network is not congested and the ramping limits are not tight. However, if the transmission network is congested, the worst case net load may not be the largest net load any more. In the following, we analyze both cases.

### A. Transmission Network Not Congested

In this case, we have the following characterization of the worst case net load.

**Proposition 2.** Consider a single period dispatch problem. Rank the loads from the largest to the smallest according to their nominal values: \( \overline{d}_1 \geq \cdots \geq \overline{d}_n \). If the transmission network is not congested and there are no ramping limits, then \( d^* \) has the following form:

\[
d^*_j = \begin{cases} 
\overline{d}_j + \hat{d}(j), & \forall j = 1, \ldots, |\Delta|, \\
\overline{d}_j + (\Delta - |\Delta|)\hat{d}(j), & j = 1 + |\Delta|, \\
\overline{d}_j, & \forall j > 1 + |\Delta|.
\end{cases}
\]

Essentially, when there is no transmission congestion and ramping limits, the worst case scenario is the net load whose total deviation from the nominal values exactly reaches the budget, and make the loads with high nominal values deviate to the upper bounds.

For the IEEE 30-bus system, the worst case scenario for \( \Delta = 6 \) is the one where the six loads with the largest nominal values (i.e., the six loads shown in Table II) reach the upper bounds \( \overline{d}_j + \hat{d}_j \), and all other loads stay at the nominal value.

### B. Transmission Network Congested

This is the more interesting case. Here, the worst case net load may not always be the highest load scenario. To demonstrate, we lower each transmission limit in the IEEE 30-bus system by half its original capacity, and increase each nominal load by 30%. The transmission network is congested in this case.

Table III shows the worst case net load scenario for the budget uncertainty set with \( \Delta = 12 \). Each column denotes an hour, from hour 1 to 12. Each row is the bus number of a load. A 1 in a cell means the load at that bus and hour reaches the upper limit, i.e. \( d_j^2 = \overline{d}_j^0 + d_j^0 \); a −1 means it reaches the lower limit, i.e. \( d_j^2 = \overline{d}_j^0 - d_j^0 \); a 0 means the load stays at the nominal value.

For the budget level \( \Delta = 12 \), the worst case scenario at each hour has 12 loads deviating to upper or lower bounds, i.e., the total number of 1 and −1 in a column is 12. The table clearly shows that at several hours, it is the lower level demand that causes the worst case dispatch cost. The reason is that in these hours the lower level demand at certain buses can cause more severe transmission congestion than a demand at the upper bound.

<table>
<thead>
<tr>
<th>Bus</th>
<th>8</th>
<th>7</th>
<th>2</th>
<th>21</th>
<th>12</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d_j^0 ) (MW)</td>
<td>34.88</td>
<td>26.51</td>
<td>25.23</td>
<td>20.35</td>
<td>13.02</td>
<td>12.33</td>
</tr>
<tr>
<td>Factors (%)</td>
<td>15.86</td>
<td>12.05</td>
<td>11.47</td>
<td>9.25</td>
<td>5.92</td>
<td>5.60</td>
</tr>
</tbody>
</table>

### V. Dynamic Uncertainty Sets

The budget uncertainty set (12) is widely used in the robust optimization literature. In fact, most of the recent works on robust UC use this type of uncertainty sets, where the uncertainty is assumed to lie in a hyper-rectangle defined by the nominal value and the intervals of variation. The budget constraint restricts the total variation of uncertainty sources. A main drawback of the budget uncertainty set is that it does not explicitly model uncertainty correlations. For this reason, we call these static uncertainty sets.

In reality, uncertainties such as wind speeds and the resulting available wind power in neighboring wind farms very often exhibit strong temporal and spatial correlations. It is important for a robust optimization model to consider these correlations in the uncertainty sets. In the following, we discuss a new type of uncertainty sets, we call dynamic uncertainty sets, which can model the dynamic relationships between uncertain parameters, and also the interactions between uncertain parameters and decisions. We show the general model first, then show a specific construction for modeling temporal and spatial correlations of available power of wind farms [10].

---

**TABLE II**

**Nominal Load Levels and Distribution Factors**

<table>
<thead>
<tr>
<th>Bus i</th>
<th>8</th>
<th>7</th>
<th>2</th>
<th>21</th>
<th>12</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d_j^0 ) (MW)</td>
<td>34.88</td>
<td>26.51</td>
<td>25.23</td>
<td>20.35</td>
<td>13.02</td>
<td>12.33</td>
</tr>
<tr>
<td>Factors (%)</td>
<td>15.86</td>
<td>12.05</td>
<td>11.47</td>
<td>9.25</td>
<td>5.92</td>
<td>5.60</td>
</tr>
</tbody>
</table>

**TABLE III**

**Worst Case Load Scenario**

<table>
<thead>
<tr>
<th>( i \times t )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d_j^2 )</td>
<td>1</td>
<td>1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Another type of dynamic uncertainty set is proposed in [11], which models the dynamics between uncertainty and decisions for demand response resource.

A. General Definition

Dynamic uncertainty sets can be generally defined as the following:

\[
\mathcal{D}_t(d_{[t-1]}) = \{d_t : \exists u \text{ s.t. } f(d_t, d_{[t-1]}, u) \leq 0\} \tag{16}
\]

where \(d_{[t-1]} = (d_1, d_2, \ldots, d_{t-1})\), and \(f\) is a function that represents the dynamic relationships between \(d_t\), \(d_{[t-1]}\), and the auxiliary variable \(u\).

The key idea here is that the function \(f\) captures the correlations between uncertain resources at different times and locations. A simple example is a dynamic uncertain interval:

\[
\mathcal{D}_t(d_{[t-1]}) = [f(d_{[t-1]}), f(d_{[t-1]})],
\]

where the interval’s upper and lower bounds at time \(t\) depend on the realizations of uncertainty in the previous periods.

B. Modeling Temporal and Spatial Correlations for Wind

In the recent paper [10], the authors propose a construction of dynamic uncertainty sets for modeling wind power’s temporal and spatial correlations between different wind farms.

First, we introduce the dynamic uncertainty set for wind speed. Denote the wind speed vector of multiple wind farms at time \(t\) as \(r_t = (r_{it}, \ldots, r_{in})\), where \(r_{it}\) is the wind speed at wind farm \(i\) and time \(t\) and \(N^w\) is the number of adjacent wind farms. Define the dynamic uncertainty set for \(r_t\) as:

\[
\mathcal{R}_t(r_{[t-L:t-1]}) = \{r_t : \exists \tilde{r}_{[t-L:t]}, u_t \text{ s.t.} \}
\]

\[
\begin{align*}
\tilde{r}_t &= g_t + \tilde{r}_r \quad \forall \tau = t - L, \ldots, t \tag{17a}
\end{align*}
\]

\[
\tilde{r}_t = \sum_{s=1}^{L} A_{t,s} \tilde{r}_{t-s} + B_t u_t \tag{17b}
\]

\[
\begin{align*}
&\sum_{i \in N^w} |u_{it}| \leq \Delta_t \tag{17c}
&|u_{it}| \leq \Gamma_t \quad \forall i \in N^w \tag{17d}
&r_t \geq 0 \tag{17e}
\end{align*}
\]

where vectors \(r_{t-L}, \ldots, r_{t-1}\) are the realizations of wind speeds in periods \(t - L, \ldots, t - 1\). \(N^w\) denote the set of wind farms. Eq. (17a) decomposes wind speed vector \(r_t\) as the sum of a seasonal pattern \(g_t\), such as periodicity of wind speeds over days, which is pre-estimated from the wind data. The residual component \(\tilde{r}_r\) represents the fluctuation of wind speeds from the seasonal pattern \(g_t\).

Eq. (17b) is the key equation that represents a linear dynamic relationship involving the residual \(\tilde{r}_t\) at time \(t\), residuals realized in earlier periods \(t - L\) to \(t - 1\), and an error term \(u_t\). Here, matrices \(A_{t,s}\)’s and \(B_t\) are estimated from a vector autoregressive time series model with time lag \(L\). In particular, \(A_{t,s}\)’s capture the temporal correlation between \(r_t\) and \(r_{t-1}, \ldots, r_{t-L}\), and \(B_t\) specifically captures the spatial relationship of wind speeds at adjacent wind farms at time \(t\). In the time series model, \(B_t\) characterizes the covariance of the estimation error.

Then, the variable \(u_t\) is introduced to control the amount of error by a budget uncertainty set defined in Eq. (17c)-(17d). The meaning of \(\Gamma_t\) is to set how many “standard deviations” we allow the errors to have. The budget constraint (17d) controls the total amount of deviations of the wind speed estimation errors across wind farms. Notice here, the budget uncertainty set is slightly different from the one used in [10]. In particular, we introduce two budget parameters \(\Delta_t\) and \(\Gamma_t\) to control the range of \(u_{it}\) and the total deviation of the \(|u_{it}|\) separately. This gives more freedom to the uncertainty sets.

It is important to also notice that we need Eq. (17e) to avoid negative wind speeds. This constraint further restricts the available range of \(u_{it}\) within the budget uncertainty set. In other words, the projection of the dynamic uncertainty set \(\mathcal{R}_t\) onto the \(u\) variables is different from the budget uncertainty set Eq. (17c)-(17d).

1) Worst Case Wind Scenario: Now, we can ask the same questions as for the budget uncertainty sets, namely, in a robust model using the dynamic uncertainty set, what would the worst case scenarios be? To study this problem, we apply the dynamic uncertainty set Eq. (17) to a two-stage robust economic dispatch (ED) model proposed in [10]. The details of the model can be found there. Here we only give a brief overview of the key components of the model.

The two-stage robust ED model has the same generic structure of a two-stage robust optimization model as shown in Eq. (1). However, the specific definition of the first- and second-stage decisions are different from the two-stage robust UC we discussed earlier in this paper. In particular, the first-stage decision of the robust ED model is the dispatch decision of the current decision period. This dispatch decision is to be implemented right now. The second-stage decision is the dispatch decisions in the future periods. The two-stage robust ED model can be viewed as a generalization of deterministic look-ahead dispatch models (e.g. see [19]).

We build a rolling horizon simulation engine to mimic the real-time dispatch operation. In this engine, the robust ED is solved every 10 min with a look-ahead horizon of 90 min. The rolling horizon simulation runs over a 35-day period. The dynamic uncertainty sets \(A_{t,s}\)’s and \(B_t\) are estimated from real-world wind data collected from [20] and updated after every day in the simulation using the newly available data.

We test on the IEEE 14-bus system with 4 wind farms and 11 loads. We assume the load forecast is accurate so as to treat the wind uncertainty alone. The wind power is dispatchable in the robust ED model, i.e., wind farm \(i\)’s production \(p_{it}^{\text{prod}}\) is a decision variable and satisfies \(p_{it}^{\text{prod}} \leq \overline{p}_{it}^{\text{prod}}\), where \(\overline{p}_{it}^{\text{prod}}\) is the available wind power calculated from wind speed by the power curve of the wind turbines. And \(\overline{p}_{it}^{\text{unc}}\) is the uncertainty in the robust dispatch model, whose uncertainty set is constructed through the uncertainty set of wind speeds (17).

At each time \(t\), the robust ED model has the observed wind speeds at time \(t\) as the input and generates a worst-case available wind power scenario for all the 4 wind farms during the periods in the look-ahead horizon (in our test, 8 periods ahead). We can point out several interesting properties of the
worst-case available wind power scenarios identified by the two-stage robust ED model.

1) Since the wind power can be curtailed in the ED model, high wind scenarios can always be handled by curtailment without additional cost, therefore, the worst case available wind scenario will be the low-wind situation.

2) If we look at the total available wind power summed over all the wind farms, the total wind power of the worst case scenario in time \( t + 1 \) is always lower than that at time \( t \), and so on for the next periods in the look-ahead horizon. That is, the worst-case wind scenarios correspond to an overall wind drop from the previous periods.

3) With a more careful study, we can see that the worst-case wind scenario of a period is typically given by the largest possible wind drop from the previous period allowed by the dynamic uncertainty set. This exactly shows the desired property of the dynamic uncertainty set that the temporal and spatial correlations of the wind speeds consistently restrict the range of wind power drop. When there is transmission congestion, the picture is more complicated. Here, the worst-case wind scenario exploits the congested network and may not correspond to the largest drop in total wind.

4) The worst-case wind power of an individual wind farm at a period may not be always a drop from the previous period, although the total wind power of the worst case drops. In other words, due to the spatial correlation between wind farms, it is possible that some increase of wind power from previous periods can be more costly for dispatch.

Figure 1 plots observed and worst-case wind scenarios over 70 periods with 10 min interval, i.e., about 12 hours in total time. In particular, each blue dot is the available wind power observed at time \( t \). Each red and green dot is the total wind power of the worst-case wind scenario for time \( t + 1 \) obtained from solving the robust ED at time \( t \). The red dots are the results of dynamic uncertainty sets with budget 0.5, whereas the green dots are for budget 1. The following are some interesting observations.

1) The red and green curves in general lie below the blue curve. This demonstrates that the worst-case wind scenarios correspond to wind drop events. The red and green curves basically provide a lower limit on the total available wind power above which the system is guaranteed to operate safely.

2) If we look carefully, we can see sometimes the red curve crosses with and go above the blue curve. At these points, the realized wind is even lower than the worst-case wind identified at the previous period. This could happen because the dynamic uncertainty set only allows a certain amount deviation from the forecast wind, controlled by the budget parameters \( \Gamma_t \) and \( \Delta_t \). If we want to reduce such “surprise” events of realized wind going below the predicted worst case, we can increase the budget. The green curve is obtained from a higher budget, and it does move the green curve further down below the red curve, which makes the green curve cross with the blue curve much less often. This shows the system operating under the green curve is more conservative and secure.

VI. CONCLUSIONS

In this paper, we explore several important questions on the robust optimization approach for the UC and ED problems that have not been addressed extensively in the existing literature. This includes the question of how to efficiently solve the second-stage problem in the robust UC model, which is a source of significant computational challenge for fully solving these adaptive robust optimization models. We also discuss with examples the properties of the worst case net load scenarios produced by the budget and dynamic uncertainty sets. Many interesting questions are still open, such as solving the multistage robust UC models, designing new dynamic uncertainty sets for solar power and demand response resources, and applying the adaptive robust optimization framework to medium and long term power system planning.

REFERENCES


