

On Robust Tie-line Scheduling in Multi-Area Power Systems

Ye Guo Subhonmesh Bose Lang Tong*

Abstract

The tie-line scheduling problem in a multi-area power system seeks to optimize tie-line power flows across areas that are independently operated by different system operators (SOs). In this paper, we leverage the theory of multi-parametric linear programming to propose algorithms for optimal tie-line scheduling respectively within a deterministic and a robust optimization framework. Aided by a coordinator, the proposed methods are proved to converge to the optimal schedule within a finite number of iterations. A key feature of the proposed algorithms, besides their finite step convergence, is that SOs do not reveal their dispatch cost structures, network constraints, or natures of uncertainty sets to the coordinator. The performance of the algorithms is evaluated using several power system examples.

1 Introduction

For historic and technical reasons, different parts of an interconnected power system and their associated assets are dispatched by different system operators (SOs). We call the geographical footprint within an SO's jurisdiction an *area*, and transmission lines that interconnect two different areas as *tie-lines*. Power flows over such tie-lines are generally scheduled 15 – 75 minutes prior to power delivery. The report in [1] indicates that current scheduling techniques often lead to suboptimal tie-line power flows. The economic loss due to inefficient tie-line scheduling is estimated to the tune of \$73 million between the areas controlled by MISO and PJM alone in 2010. Tie-lines often have enough transfer capability to fulfill a significant portion of each area's power consumption [2]. Thus they form important assets of multi-area power systems.

SOs from multiple areas typically cannot aggregate their dispatch cost structures and detailed network constraints to solve a *joint* optimal power flow problem. Therefore, distributed algorithms have been proposed. Prominent examples include [3–5] that adopt the so-called *dual decomposition* approach. These methods are iterative, wherein each SO optimizes the grid assets within its area, given the Lagrange multipliers associated with inter-area constraints. Typically, a coordinator mediates among the SOs and iteratively updates the multipliers. Alternative *primal decomposition* approaches are also proposed in [6–8]. Therein, the primal variables of the optimization problem are iteratively updated, sometimes requiring the SO of one area to reveal part of its cost structure and constraints to the SO of another area or a coordinator.

Traditionally, solution techniques for the tie-line scheduling problem assume that the SOs and/or the coordinator has perfect knowledge of the future demand and supply conditions at the time

*Y. Guo and L. Tong are with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA. (Emails: {yg299,lt35}@cornell.edu). S. Bose is with the Dept. of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign, Urbana, IL, USA. (Email: boses@illinois.edu).

of scheduling. Such assumptions are being increasingly challenged with the rapid adoption of distributed energy resources in the distribution grid and variable renewable generation like wind and solar energy in the bulk power systems. Said differently, one must explicitly account for the uncertainty in demand and supply in the tie-line scheduling problem. To that end, [9, 10] propose to minimize the expected aggregate dispatch cost and [11] propose to minimize the maximum of that cost. In this paper, we adopt the latter paradigm – the *robust* approach.

Our contribution

With the system model in Section 2, we first formulate the deterministic tie-line scheduling problem in Section 3, where we propose an algorithm to solve this deterministic problem that draws from the theory of multiparametric programming [12]. The key feature of our algorithm is that a coordinator can produce the optimal tie-line schedule upon communicating only *finitely many times* with the SO in each area. In contrast to [8], our method does not require SOs to reveal their cost structures nor their constraints to other SOs or to the coordinator. In Section 4, we formulate the robust counterpart of the tie-line scheduling problem. We then propose a technique that alternately uses the algorithm for the deterministic variant and a mixed-integer linear program to solve the robust problem. Again, our technique is proved to converge to the optimal robust tie-line schedule that requires the coordinator to communicate finitely many times with each SO. Also, SOs are not required to reveal the nature and range of the values the uncertain demand and available supply can take. Our proposed framework thus circumvents the substantial communication burden of the method proposed in [11] towards the same problem. We remark that [11] adopts the *column-and-constraint generation* technique described in [13] that requires SOs to reveal part of their network constraints, costs and ranges of demand and available renewable supply to the coordinator. We empirically demonstrate the performance of our algorithm in Section 5 and conclude in Section 6.

2 System model

To formulate the tie-line scheduling problem, we begin by describing the model for multi-area power systems. Throughout, we restrict ourselves to a two-area power system, pictorially represented in Figure 1 for the ease of exposition. The model and the proposed methods can be generalized for tie-line scheduling among more than two areas.

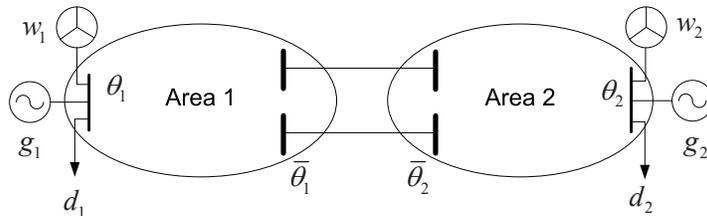


Figure 1: An illustration of a two-area power system.

For the power network in each area, we distinguish between two types of buses: the *internal* buses and the *boundary* buses. The boundary ones in each area are connected to their counterparts in the other area via tie-lines. Internal buses do not share a connection to other areas. Assume that

each internal bus has a dispatchable generator, a renewable generator, and a controllable load¹. Boundary buses do not have any asset that can inject or extract power. Such assumptions are not limiting in that one can derive an equivalent power network in each area that adheres to these assumptions.

Let the power network in area i be comprised of n_i internal buses and \bar{n}_i boundary buses for each $i = 1, 2$. We adopt a linear DC power flow model in this paper.² This approximate model sets all voltage magnitudes to their nominal values, ignores transmission line resistances and shunt reactances, and deems differences among the voltage phase angles across each transmission line to be small. Consequently, the real power injections into the network is a linear map of voltage phase angles (expressed in radians) across the network. To arrive at a mathematical description, denote by $g_i \in \mathbf{R}^{n_i}$, $w_i \in \mathbf{R}^{n_i}$, and $d_i \in \mathbf{R}^{n_i}$ as the vectors of (real) power generations from dispatchable generators, renewable generators, and controllable loads, respectively. Let $\theta_i \in \mathbf{R}^{n_i}$ and $\bar{\theta}_i \in \mathbf{R}^{\bar{n}_i}$ be the vectors of voltage phase angles at internal and boundary buses, respectively. Then, the power flow equations are given by

$$\begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{1\bar{1}} & & & \\ \mathbf{B}_{\bar{1}1} & \mathbf{B}_{\bar{1}\bar{1}} & \mathbf{B}_{\bar{1}\bar{2}} & & \\ & \mathbf{B}_{\bar{2}\bar{1}} & \mathbf{B}_{\bar{2}\bar{2}} & \mathbf{B}_{\bar{2}2} & \\ & & \mathbf{B}_{2\bar{2}} & \mathbf{B}_{22} & \end{pmatrix} \begin{pmatrix} \theta_1 \\ \bar{\theta}_1 \\ \bar{\theta}_2 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} g_1 + w_1 - d_1 \\ 0 \\ 0 \\ g_2 + w_2 - d_2 \end{pmatrix}. \quad (1)$$

Non-zero entries of the coefficient matrix depend on reciprocals of transmission line reactances, the unspecified blocks in that matrix are zeros. Throughout, assume that one of the boundary buses in area 1 is set as the slack bus for the two-area power system. That is, the voltage phase angle at said bus is assumed zero.

Power injections from the supply and demand assets at the internal buses of area i are constrained as

$$\underline{G}_i \leq g_i \leq \bar{G}_i, \quad 0 \leq w_i \leq \bar{W}_i, \quad \underline{D}_i \leq d_i \leq \bar{D}_i. \quad (2)$$

The inequalities are interpreted elementwise. The lower and upper limits on dispatchable generation $\underline{G}_i, \bar{G}_i$ are assumed to be known at the time when tie-line flows are being scheduled. Our assumptions on the available renewable generation \bar{W}_i and the limits on the demands $[\underline{D}_i, \bar{D}_i]$ will vary in the subsequent sections. In Section 3, we assume that these limits are known and provide a distributed algorithm to solve the deterministic tie-line scheduling problem. In Section 4, we formulate the robust counterpart, where these limits are deemed uncertain and vary over a known set. We then describe a distributed algorithm to solve the robust counterpart.

The power transfer capabilities of transmission lines within area i are succinctly represented as

$$\mathbf{H}_i \theta_i + \bar{\mathbf{H}}_i \bar{\theta}_i \leq f_i \quad (3)$$

for each $i = 1, 2$. Here, \mathbf{H}_i and $\bar{\mathbf{H}}_i$ define the branch-bus admittance matrices, and f_i models the respective transmission line capacities. Similarly, the transfer capabilities of tie-lines joining the two areas assume the form

$$\bar{\mathbf{H}}_{12} \bar{\theta}_1 + \bar{\mathbf{H}}_{21} \bar{\theta}_2 \leq f_{12}. \quad (4)$$

¹While we assume that all loads are controllable, uncontrollable load at any node can be easily modeled by letting the limits on the allowable power demand at that node to be equal.

²See [14, 15], and the references therein for solution approaches for a multi-area ACOPF problem.

Again, $\overline{\mathbf{H}}_{12}, \overline{\mathbf{H}}_{21}$ denote the relevant branch-bus admittance matrices and f_{12} models the tie-line capacities.

Finally, we describe the cost model for our two-area power system. For respectively procuring g_i and w_i from dispatchable and renewable generators, and meeting a demand of d_i from controllable loads, let the dispatch cost in area i be given by

$$[P_i^g]^\top g_i + [P_i^w]^\top (\overline{W}_i - w_i) + [P_i^d]^\top (\overline{D}_i - d_i). \quad (5)$$

We use the notation v^\top to denote the transpose of any vector or matrix v . The linear cost structure in the above equation is reminiscent of electricity market practices in many parts of the U.S. today. The second summand models any spillage costs associated with renewable generators. The third models the disutility of not satisfying all demands.

3 The deterministic tie-line scheduling problem

Tie-line flows are typically scheduled ahead of the time of power delivery. The lead time makes the supply and demand conditions uncertain during the scheduling process. Within the framework of our model, the available capacity in renewable supply and lower and upper bounds on power demands, i.e., $\overline{W}_i, \underline{D}_i, \overline{D}_i$, can be uncertain. In this section, we ignore such uncertainty and formulate the deterministic tie-line scheduling problem, wherein we assume perfect knowledge of $\overline{W}_i, \underline{D}_i$ and \overline{D}_i to decide the dispatch in each area and the tie-line flows. Our discussion of the deterministic version will serve as a prelude to its robust counterpart in Section 4.

To simplify exposition, consider the following notation.

$$x_i := (g_i, w_i, d_i, \theta_i)^\top, \quad \xi_i := (\overline{W}_i, \underline{D}_i, \overline{D}_i)^\top, \quad y := (\overline{\theta}_1, \overline{\theta}_2)^\top$$

for $i = 1, 2$. The above notation allows us to succinctly represent the constraints (1) – (3) as

$$\mathbf{A}_i^x x_i + \mathbf{A}_i^\xi \xi_i + \mathbf{A}_i^y y \leq b_i$$

for each $i = 1, 2$ and suitably defined matrices $\mathbf{A}_i^x, \mathbf{A}_i^\xi, \mathbf{A}_i^y$ and vector b_i . Denote by m_i the number of inequality constraints in the above equation. Next, we describe transmission constraints on tie-line power flows in (4) as

$$y \in \mathcal{Y} \subset \mathbf{R}^Y.$$

Without loss of generality, one can restrict \mathcal{Y} to be a polytope³. Finally, the cost of dispatch in area i , as described in (5), can be written as

$$c_i(x_i, \xi_i) := c_i^0 + [c_i^x]^\top x_i + [c_i^\xi]^\top \xi_i$$

for scalar c_i^0 and vectors c_i^x, c_i^ξ .

Equipped with the above notation, we define the deterministic tie-line scheduling problem as follows.

$$\begin{aligned} & \underset{x_1, x_2, y}{\text{minimize}} && [c_1(x_1, \xi_1) + c_2(x_2, \xi_2)], \\ & \text{subject to} && \mathbf{A}_i^x x_i + \mathbf{A}_i^\xi \xi_i + \mathbf{A}_i^y y \leq b_i, \quad i = 1, 2, \\ & && y \in \mathcal{Y}. \end{aligned} \quad (6)$$

³Assuming the power network to be connected, the modulus of the phase angle of any bus can be constrained to lie within the sum of admittance-weighted transmission line capacities connecting that bus to the slack bus.

3.1 Distributed solution via critical region exploration

The structure of the optimization problem in (6) lends itself to a distributed solution architecture that we describe below. Our proposed technique is similar in spirit to the critical region projection method described in [6].⁴ We assume that each area is managed by a system operator (SO), and a *coordinator* mediates between the SOs. Assume that the SO of area i (call it SO_i) knows the dispatch cost c_i and the linear constraint involving x_i, ξ_i, y in (6) in area i , and that SOs and the coordinator all know \mathcal{Y} .

Our algorithm relies on the properties of (6) that we describe next. To that end, notice that (6) can be written as

$$\underset{y \in \mathcal{Y}}{\text{minimize}} \quad J^*(y, \xi_1, \xi_2) := J_1^*(y, \xi_1) + J_2^*(y, \xi_2), \quad (7)$$

where

$$J_i^*(y, \xi_i) := \underset{x_i}{\text{minimum}} \quad c_i(x_i, \xi_i), \quad \text{subject to} \quad \mathbf{A}_i^x x_i + \mathbf{A}_i^\xi \xi_i + \mathbf{A}_i^y y \leq b_i. \quad (8)$$

Assume throughout that all optimization problems parameterized by y is feasible for each $y \in \mathcal{Y}$. Techniques from [11] can be leveraged to shrink \mathcal{Y} appropriately, otherwise. The optimization problem in (8) is a multi-parametric linear program, linearly parameterized in (y, ξ_i) on the right-hand side⁵. Such optimization problems are well-studied in the literature. For example, see [12]. Relevant to our algorithm is the structure of the parametric optimal cost J_i^* . Describing that structure requires an additional notation. We say that a finite collection of polytopes $\{\mathcal{P}^1, \dots, \mathcal{P}^\ell\}$ define a *polyhedral partition* of \mathcal{Y} , if no two polytopes intersect except at their boundaries, and their union equals \mathcal{Y} . With this notation, we now record the properties of J_i^* in the following lemma.

Lemma 1. $J_i^*(y, \xi_i)$ is piecewise affine and convex in $y \in \mathcal{Y}$. Sets over which $J_i^*(\cdot, \xi_i)$ is affine define a polyhedral partition of \mathcal{Y} .

The proof is immediate from [12, Theorem 7.5]. Details are omitted for brevity. We refer to the polytopes in the polyhedral partition of \mathcal{Y} induced by $J_i^*(\cdot, \xi_i)$ as *critical regions*. Recall that the feasible set of (8) is described by a collection of linear inequalities. Essentially, each critical region corresponds to the subset of \mathcal{Y} over which a specific set of these inequality constraints are active – *i.e.*, are met with an equality – at an optimal solution of (8).

A direct consequence of the above lemma is that the *aggregate cost* $J^*(\cdot, \xi_1, \xi_2)$ is also piecewise-affine and convex. Sets over which this cost is affine define a polyhedral partition of \mathcal{Y} . The polytopes of that partition – the critical regions – are precisely the non-empty intersections between the critical regions induced by $J_1^*(\cdot, \xi_1)$ and those by $J_2^*(\cdot, \xi_2)$. The relationship between the critical regions induced by the various piecewise affine functions are illustrated in Figure 2. In what follows, we develop an algorithm wherein the coordinator defines a sequence of points in \mathcal{Y} towards optimizing the aggregate cost. In each step, it relies on the SOs to identify their respective critical regions and the affine descriptions of their optimal costs at these iterates. That is, SO_i can compute the critical region \mathcal{P}_i^y that contains $y \in \mathcal{Y}$ and the affine description $[\alpha_i^y]^\top z + \beta_i^y$ of its optimal dispatch cost $J_i^*(z, \xi_i)$ over $z \in \mathcal{P}_i^y$ by parameterizing the linear program described in (8)⁶. We

⁴The cost structure in [6] is quadratic; the linear cost case does not directly follow from [6].

⁵The problem in (8) reformulated using the so-called epigraph form yields a multi-parametric program that is classically recognized as one linearly parameterized on the right-hand side.

⁶The critical region containing $y \in \mathcal{Y}$ is unique, except when y lies at the boundary of critical regions. In that event, assume that the SO returns one of the critical regions containing y .

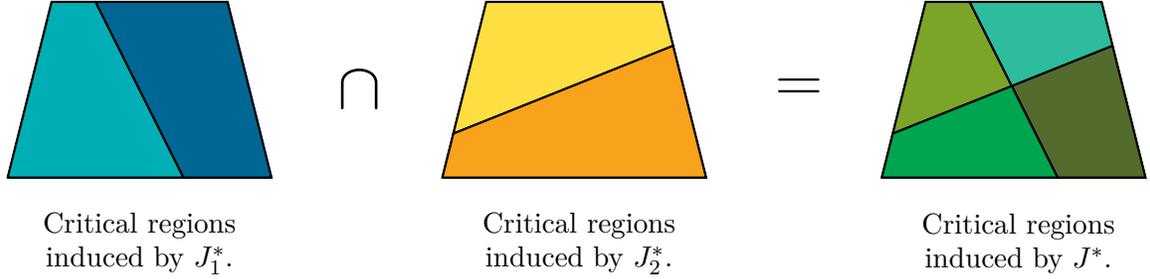


Figure 2: A pictorial representation of the critical regions induced by the areawise parametric optimal costs $J_1^*(\cdot, \xi_1)$, $J_2^*(\cdot, \xi_2)$, and the aggregate cost $J^*(\cdot, \xi_1, \xi_2)$. The trapezoids represent \mathcal{Y} . Differently shaded polytopes indicate different critical regions.

relegate the details of this step to Appendix A to maintain continuity of presentation. For any $y \in \mathcal{Y}$, we assume in the sequel that the coordinator can collect this information from the SOs to construct the critical region \mathcal{P}^y induced by the aggregate cost containing y and its affine description $[\alpha^y]^\top z + \beta^y$ for $z \in \mathcal{P}^y$, where

$$\mathcal{P}^y := \mathcal{P}_1^y \cap \mathcal{P}_2^y, \quad \alpha^y := \alpha_1^y + \alpha_2^y, \quad \beta^y := \beta_1^y + \beta_2^y. \quad (9)$$

In presenting the algorithm, we assume that the coordinator can identify the *lexicographically smallest* optimal solution of a linear program. A vector a is said to be lexicographically smaller than b , if at the first index where they differ, the entry in a is less than that in b . See [16] for details on such linear programming solvers. When a linear program does not have a unique optimizer⁷, such a choice provides a tie-breaking rule. The final piece required to state and analyze the algorithm is an optimality condition that is both necessary and sufficient for a candidate minimizer of (7). Stated geometrically, $y^* \in \mathcal{Y}$ is a minimizer of (7) if and only if

$$0 \in \partial J^*(y^*, \xi_1, \xi_2) + N_{\mathcal{Y}}(y^*). \quad (10)$$

The first set on the right-hand side of (10) is the sub-differential set of the aggregate cost $J^*(\cdot, \xi_1, \xi_2)$ evaluated at y^* ⁸. And, the second set denotes the normal cone to \mathcal{Y} at y^* . The addition stands for a set-sum.

Algorithm 1 delineates the steps for the coordinator to solve the deterministic tie-line scheduling problem. In our algorithm, $\|v^*\|_2$ denotes the Euclidean norm of v^* . If $\mathcal{D} := \{\alpha^1, \dots, \alpha^{\ell_D}\}$ and $N_{\mathcal{Y}}(y^*) := \{z \mid \mathbf{K}^y z \geq 0\}$, then computing the least-square solution v^* amounts to solving the following convex quadratic program.

$$\text{minimize } \frac{1}{2} \|v\|_2^2, \quad \text{subject to } v = \sum_{j=1}^{\ell_D} \eta_j \alpha^j + \zeta, \quad \mathbf{1}^\top \eta = 1, \quad \eta \geq 0, \quad \mathbf{K}^y \zeta \geq 0 \quad (11)$$

over the variables $v \in \mathbf{R}^{\bar{n}_1 + \bar{n}_2}$, $\eta \in \mathbf{R}^{\ell_D}$, and $\zeta \in \mathbf{R}^{\ell_N}$, where $\mathbf{1}$ is a vector of all ones, and $\mathbf{K}^y \in \mathbf{R}^{(\bar{n}_1 + \bar{n}_2) \times \ell_N}$.

⁷A linear program has non-unique optimizers when it is *dual degenerate*. See [16] for details.

⁸We use the sub-differential characterization as opposed to the familiar gradient condition for optimality since $J^*(\cdot, \xi_1, \xi_2)$ is piecewise affine and may not be differentiable everywhere in \mathcal{Y} .

Algorithm 1 Solving the deterministic tie-line scheduling problem.

```
1: Initialize:  
    $y \leftarrow$  any point in  $\mathcal{Y}$ ,  $J^* \leftarrow \infty$ ,  
    $\mathcal{D} \leftarrow$  empty set,  $\varepsilon \leftarrow$  small positive number.  
2: do  
3:   Communicate with the SOs to obtain  $\mathcal{P}^y$  and  $\alpha^y, \beta^y$ .  
4:   Minimize  $[\alpha^y]^\top z + [\beta^y]$  over  $\mathcal{P}^y$ .  
5:    $y^{\text{opt}} \leftarrow$  lexicographically smallest minimizer in step 4.  
6:    $J^{\text{opt}} \leftarrow$  optimal cost in step 4.  
7:   if  $J^{\text{opt}} < J^*$ , then  
8:      $y^* \leftarrow y^{\text{opt}}$ ,  $J^* \leftarrow J^{\text{opt}}$ ,  $\mathcal{D} \leftarrow \{\alpha^y\}$ .  
9:   else  
10:     $\mathcal{D} \leftarrow \mathcal{D} \cup \{\alpha^y\}$ .  
11:  end if  
12:   $v^* \leftarrow \operatorname{argmin}_{v \in \operatorname{conv}(\mathcal{D}) + N_{\mathcal{Y}}(y^*)} \|v\|_2^2$ .  
13:   $y \leftarrow y^{\text{opt}} - \varepsilon v^*$ .  
14: while  $v^* \neq 0$ .
```

3.2 Analysis of the algorithm

The following result characterizes the convergence of Algorithm 1. See Appendix B for its proof.

Theorem 1. *Algorithm 1 terminates after finitely many steps, and y^* at termination optimally solves (7).*

The above result fundamentally relies on the fact that each time the variable y is updated, it belongs to a critical region (induced by the aggregate cost) that the algorithm has not encountered so far. And, there are only finitely many such critical regions. That ensures termination in finitely many steps. Each time the algorithm ventures into a new critical region, we store the optimizer and the optimal cost over that critical region in the variables y^{opt} and J^{opt} . Forcing the linear program to choose the lexicographically smallest optimizer always picks a unique vertex of the critical region as y^{opt} . Unless J^{opt} improves upon the cost at y^* , we ignore the new point y^{opt} . However, the exploration of the new critical region provides a possibly new sub-gradient of the aggregate cost at y^* . The sub-differential set at y^* is given by the convex hull of the sub-gradients of the aggregate cost over all critical regions that y^* is a part of. The set \mathcal{D} we maintain is such that $\operatorname{conv}(\mathcal{D})$ is a *partial* sub-differential set of the aggregate cost at y^* . Notice that

$$\operatorname{conv}(\mathcal{D}) \subseteq \partial J^*(y^*, \xi_1, \xi_2)$$

throughout the algorithm. Therefore, any y^* that meets the termination criterion of the algorithm automatically satisfies (10). As a result, such a y^* is an optimizer of (7).

The proposed technique is attractive in that each SO only needs to communicate finitely many times with the coordinator for the latter to reach an optimal tie-line schedule. Further, each SO_i can compute its optimal dispatch x_i^* by solving (8) with y^* . A closer look at the nature of the communication between the SOs and the coordinator reveals that an SO will not have to reveal the complete cost structure nor a complete description of the constraints within its area to the coordinator.

Remark 1. Algorithm 1 allows the coordinator to minimize

$$F(y) := F_1(y) + F_2(y)$$

in a distributed manner, where $F_i : \mathcal{Y} \rightarrow \mathbf{R}$ satisfies two properties. First, it is piecewise affine and convex. Second, given any $y \in \mathcal{Y}$, SO_i can compute an affine segment containing that y . While we do not explicitly characterize how fast the algorithm converges to its optimum, one can expect the number of steps to convergence to grow with the number of critical regions so induced. However, we do not expect our algorithm to explore all such critical regions on its convergence path.

3.3 A pictorial illustration of the algorithm

To gain more insights into the mechanics of Algorithm 1, consider the example portrayed in Figure 3. The coordinator begins with y^A as the initial value of y . It communicates with SO_i to obtain the critical region induced by J_i^* containing y^A , and the affine description of J_i^* over that critical region. Using the relation in (9), it then computes the critical region \mathcal{P}^A induced by the aggregate cost and the affine description of that cost $[\alpha^A]^\top z + \beta^A$ over that region. For convenience, we use

$$\mathcal{P}^A := \mathcal{P}^{y^A}, \quad \alpha^A := \alpha^{y^A}, \quad \beta^A := \beta^{y^A},$$

and extend the corresponding notation for y^B, \dots, y^E .

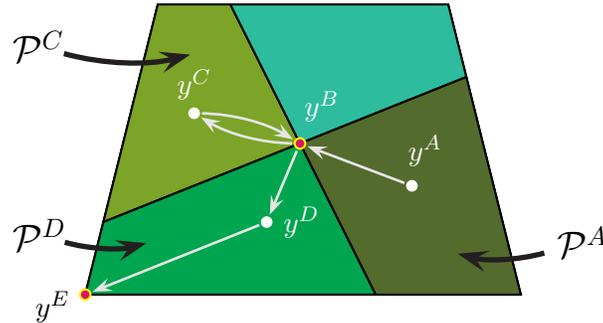


Figure 3: An example to illustrate the iterative process of Algorithm 1.

The coordinator solves a linear program to minimize the affine aggregate cost $[\alpha^A]^\top z + \beta^A$ over $z \in \mathcal{P}^A$, and obtains the lexicographically smallest optimizer y^{opt} . Such an optimizer y^{opt} is always a vertex of \mathcal{P}^A . Identify y^B as that vertex in Figure 3. The optimal cost at y^B is indeed lower than the initial value of $J^* = \infty$, and hence, the coordinator sets $y^* \leftarrow y^B$. It also updates J^* to the aggregate cost at y^B , and the partial sub-differential set to $\mathcal{D} \leftarrow \{\alpha^A\}$.

Next, the coordinator solves the least square problem described in (11) to compute v^* . In so doing, it utilizes $\mathcal{D} = \{\alpha^A\}$, and $\mathbf{K}^y = 0$ that describes the normal cone to \mathcal{Y} at y^B .⁹ Suppose $v^* \neq 0$. The coordinator updates the value of y to y^C , obtained by moving a ‘small’ step of length ε from y^B along $-v^*$. Recall that $y^C \notin \mathcal{P}^A$. The coordinator again communicates with the SOs

⁹The normal cone to \mathcal{Y} at y^B is $\{0\}$ because y^B lies in the interior of \mathcal{Y} .

to obtain the new critical region \mathcal{P}^C induced by the aggregate cost that contains y^C . Again, it obtains the affine description of that cost and optimizes it over \mathcal{P}^C to obtain the new y^{opt} . In the figure, we depict the case when y^{opt} coincides with $y^* = y^B$.

Notice that the optimal cost J^{opt} at y^{opt} is equal to J^* , and hence, the coordinator only updates the partial sub-differential set \mathcal{D} to $\{\alpha^A, \alpha^C\}$. With the updated set of \mathcal{D} , the coordinator solves (11) to obtain v^* . In this example, v^* is again non-zero, and hence, the coordinator moves along a step of length ε along $-v^*$ from y^B to land at y^D . Again, $y^D \notin \{\mathcal{P}^A, \mathcal{P}^C\}$. The coordinator repeats the same steps to optimize the aggregate cost over \mathcal{P}^D to obtain y^E as the new y^{opt} . Two cases can now arise, that we describe separately.

- If the optimal cost J^{opt} at $y^{\text{opt}} = y^E$ does not improve upon the cost J^* at y^B , the coordinator ignores y^E and updates the set \mathcal{D} to $\{\alpha^A, \alpha^C, \alpha^D\}$. It computes v^* with the updated \mathcal{D} . Again, if $v^* \neq 0$, it traverses along $-v^*$ to venture into a yet-unexplored critical region. The process continues till we get $y^* = y^B$ as an optimizer (if $v^* = 0$ at a future iterate), or we encounter the case we describe next.
- If $J^{\text{opt}} < J^*$, then the coordinator sets y^E as the new y^* . It retraces the same steps with this new y^* . Since y^E is a vertex of \mathcal{Y} , one can show that (11) will yield $v^* = 0$, and hence, $y^* = y^E$ will optimize the aggregate cost over \mathcal{Y} .

4 The robust counterpart

The deterministic tie-line scheduling problem was formulated in the last section on the premise that available renewable supply and limits on power demands within each area are known at the time when tie-line schedules are decided. We now alter that assumption and allow these parameters to be uncertain. In particular, we let $\xi_i = (\overline{W}_i, \underline{D}_i, \overline{D}_i)$ assume values in a *box*, described by

$$\Xi_i := \{\xi_i \in \mathbf{R}^{3n_i} \mid \xi_i^L \leq \xi_i \leq \xi_i^U\} \quad (12)$$

for $i = 1, 2$. The robust counterpart of the tie-line scheduling problem is then described by

$$\underset{y \in \mathcal{Y}}{\text{minimize}} \left(\max_{\xi_1 \in \Xi_1} J_1^*(y, \xi_1) + \max_{\xi_2 \in \Xi_2} J_2^*(y, \xi_2) \right). \quad (13)$$

We now develop an algorithm that solves (13) in a distributed fashion. Problem (13) has a *minimax* structure. Therefore, we employ a strategy in Algorithm 2 to alternately minimize the objective function over \mathcal{Y} and maximize it over $\Xi_1 \times \Xi_2$. Thanks to the following lemma, the maximization over $\Xi_1 \times \Xi_2$ can be reformulated into a mixed-integer linear program.

Lemma 2. *Fix $y \in \mathcal{Y}$. Then, maximizing $J_i^*(y, \xi_i)$ over $\xi_i \in \Xi_i$ is equivalent to the following mixed-integer linear program for some $M > 0$.*

$$\begin{aligned} \underset{w_i, \rho, \lambda}{\text{maximize}} \quad & c_i^0 + [c_i^\xi]^\top \xi_i^L + (\mathbf{A}_i^\xi \xi_i^L + \mathbf{A}_i^y y - b_i)^\top \lambda + \mathbf{1}^\top \rho, \\ \text{subject to} \quad & c_i^x + [\mathbf{A}_i^x]^\top \lambda = 0, \\ & \rho \leq M w_i, \\ & \rho \leq M(\mathbf{1} - w_i) + \Delta_i^\xi (c_i^\xi + [\mathbf{A}_i^\xi]^\top \lambda), \\ & w_i \in \{0, 1\}^{n_i}, \rho \in \mathbf{R}^{n_i}, \lambda \in \mathbf{R}_+^{m_i}. \end{aligned} \quad (14)$$

We use the notation Δ_i^ξ to denote a diagonal matrix with $\xi_i^U - \xi_i^L$ as the diagonal. The lemma builds on the fact that $J_i^*(y, \xi_i)$ is convex in ξ_i , and hence, reaches its maximum at a vertex of Ξ_i . The convexity is again a consequence of [12, Theorem 7.5]. Our proof in Appendix C leverages duality theory of linear programming and the so-called *big-M* method adopted in [17, Chapter 2.11] to reformulate the maximization of $J_i^*(y, \cdot)$ over the vertices of Ξ_i into a mixed-integer linear program. An optimal ξ_i^{opt} can be recovered from w_i^* that is optimal in (14) using

$$\xi_i^{\text{opt}} := \xi_i^L + \Delta_i^\xi w_i^*.$$

Next, we present our algorithm for solving the robust counterpart. In the algorithm, the SOs exclusively maintain and update certain variables; we distinguish these from the ones the coordinator maintains.

Algorithm 2 Solving the robust counterpart.

1: **Initialize:**

SO₁: $\mathcal{V}_1 \leftarrow \{\text{a vertex of } \Xi_1\}$,

SO₂: $\mathcal{V}_2 \leftarrow \{\text{a vertex of } \Xi_2\}$.

2: **do**

3: Coordinator uses Algorithm 1 to solve

$$\text{minimize}_{y \in \mathcal{Y}} \left(\max_{\xi_1 \in \mathcal{V}_1} J_1^*(y, \xi_1) + \max_{\xi_2 \in \mathcal{V}_2} J_2^*(y, \xi_2) \right).$$

4: $y^* \leftarrow$ optimizer in step 3.

5: $J^* \leftarrow$ optimal cost in step 3.

6: For $i = 1, 2$, SO _{i} performs:

7: Maximize $J_i^*(y^*, \cdot)$ over Ξ_i using (14).

8: $\xi_i^{\text{opt}} \leftarrow$ optimizer in step 7.

9: $J_i^{\text{opt}} \leftarrow$ optimal cost in step 7.

10: $\mathcal{V}_i \leftarrow \mathcal{V}_i \cup \{\xi_i^{\text{opt}}\}$.

11: **return** J_i^{opt} to the coordinator.

12: **while** $J_1^{\text{opt}} + J_2^{\text{opt}} > J^*$.

We summarize the main property of the above algorithm in the following theorem, whose proof is given in Appendix D¹⁰.

Theorem 2. *Algorithm 2 terminates after finitely many steps, and y^* at termination optimally solves (13).*

Our algorithm to solve the robust counterpart makes use of Algorithm 1 in step 3. The coordinator performs this step with necessary communication with the SOs. However, it remains agnostic to the uncertainty sets Ξ_1 and Ξ_2 throughout. Therefore, our algorithm is such that the SOs in general will not be required to reveal their cost structures, network constraints, nor their uncertainty sets to the coordinator to optimally solve the robust tie-line scheduling problem. Further, Theorems 1 and 2 together guarantee that the coordinator can arrive at the required schedule by

¹⁰The proof is similar to [13, Proposition 2]; we include it for completeness.

communicating with the SOs only finitely many times. These define some of the advantages of the proposed methodology. In the following, we discuss some limitations of our method.

The number of affine segments in the piecewise affine description of $\max_{\xi_i \in \mathcal{V}_i} J_i^*(y, \xi_i)$ increases with the size of the set \mathcal{V}_i . The larger that number, the heavier can be the computational burden on Algorithm 1 in step 3. To partially circumvent this problem, we initialize the sets \mathcal{V}_i with that vertex of Ξ_i that encodes the least available renewable supply and the highest nominal demand. Such a choice captures the intuition that dispatch cost is likely the highest with the least free renewable supply and the highest demand. Our empirical results in the next section corroborate that intuition.

We make use of mixed-integer linear programs in step 7 of the algorithm. While this optimization class encompasses well-known NP-hard problems, solvers in practice have been known to have good empirical performance. Popular techniques for mixed-integer linear programming include branch-and-bound, cutting-plane methods, etc. See [17] for a survey. Providing polynomial-time convergence guarantees for (14) remain challenging; our empirical results in the next section, however, appear encouraging.

5 Numerical Experiments

We report here the results of our implementation of Algorithm 2 on several power system examples. All optimization was solved in IBM ILOG CPLEX Optimization Studio V12.5.0 [18] on a PC with 2.0GHz Intel(R) Core(TM) i7-4510U microprocessor and 8GB RAM.

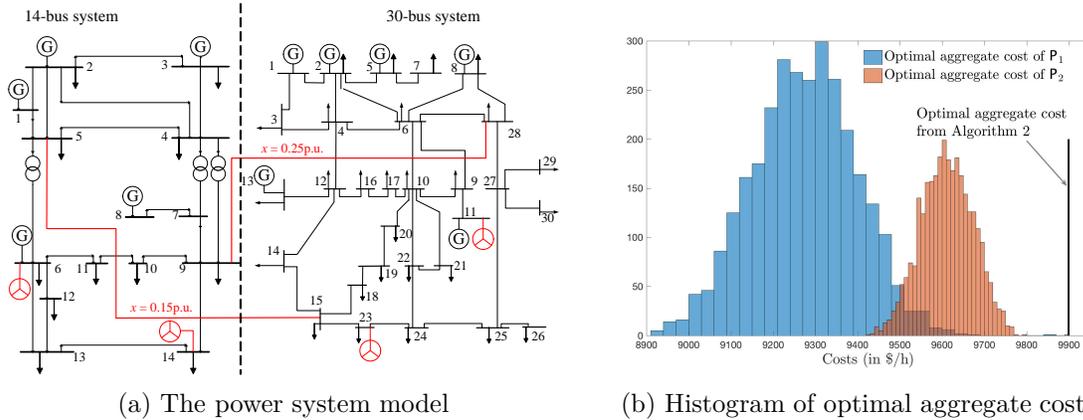


Figure 4: The two-area 44-bus system is portrayed on the left. It shows where the wind generators are added and the parameters for the tie-lines used in our experiments. The figure to the right plots the optimal aggregate costs from P_1 , P_2 over 3000 samples of uncertain variables, and that of Algorithm 2 on this system.

5.1 On a two-area 44-bus power system

Consider the two-area power system shown in Figure 4a, obtained by connecting the IEEE 14 and 30 bus test systems [19]. The networks were augmented with wind generators at various buses. Transmission capacities of all lines were set to 100MW. The available capacity of each wind generator was varied between 15MW and 25MW. The lower limits on all power demands were set to zero, while the upper limits were varied between 98% and 102% of their nominal values. Our

setup had 36 uncertain variables – 32 power demands and 4 available wind generation. Bus 5 in area 1 was the slack bus. From the data in Matpower [20], we chose the linear coefficient in the nominal quadratic cost structure for each conventional generator to define P_i^g in (5). Further, we neglected wind spillage costs by letting $P_i^w = 0$, and defined P_i^d by assuming a constant marginal cost of \$100/MWh for not meeting the highest demands.

Iteration	Step in Algorithm 2	Aggregate cost (in \$/h)	Run-time (in ms)
1	Step 3 to compute y^*	9897.7	113.6
1	Step 7 to compute ξ^{opt}	9910.3	99.6
2	Step 3 to compute y^*	9899.3	93.4
2	Step 7 to compute ξ^{opt}	9899.3	121.5

Table 1: Evolution of aggregate cost of Algorithm 2 for the two-area power system in Figure 4a.

To run Algorithm 2, we initialized \mathcal{V}_i with the scenario that describes the highest power demands and the least available wind generation across all buses. To invoke Algorithm 1 in step 3, we initialized y with a vector of all zeros. When the algorithm encountered the same step in future iterations, it was initialized with the optimal y^* from the last iteration to provide a *warm start*. Algorithm 2 converged in two iterations, *i.e.*, it ended when the cardinality of \mathcal{V}_1 and \mathcal{V}_2 were both two. The trajectory of the optimal cost and the run-times for each step are given in Table 1. In the first iteration, Algorithm 1 in step 3 with $\varepsilon = 10^{-5}$ converged in four iterations¹¹ of its own and explored five critical regions induced by the aggregate cost. A naive search over \mathcal{Y} yielded that the aggregate cost induces at least 126 critical regions. Our simulation indicates that Algorithm 1 only explores a ‘small’ subset of all critical regions.

Step 7 of Algorithm 2 was then solved to obtain ξ_i^{opt} . As Table 1 suggests, the aggregate cost $J_1^{\text{opt}} + J_2^{\text{opt}}$ exceeded J^* obtained earlier in step 3. Thus, the scenario of demand and supply captured in our initial sets \mathcal{V}_1 and \mathcal{V}_2 was *not* the one with maximum aggregate dispatch costs. To accomplish this step, two separate mixed-integer linear programs were solved – one with 13 binary variables (in area 1) and the other with 23 binary variables (in area 2). CPLEX returned the global optimal solutions in 15ms and 77ms, respectively. In the next iteration, step 3 was performed with ξ_i^{opt} added to \mathcal{V}_i , where Algorithm 1 converged in five iterations, exploring only four critical regions. Finally, step 7 yielded $J_1^{\text{opt}} + J_2^{\text{opt}} = J^*$, implying that the obtained y^* defines an optimal robust tie-line schedule.

To further understand the efficacy of our solution technique, we uniformly sampled the set $\Xi_1 \times \Xi_2$ 3000 times. With each sample (ξ_1, ξ_2) , we solved two optimization problems – P_1 and P_2 . Precisely, P_1 is a deterministic tie-line scheduling problem solved with Algorithm 1, and P_2 is the optimal power flow problem in each area with the optimal y^* obtained from Algorithm 2 for the robust counterpart. The histograms of the optimal aggregate costs from P_1 and P_2 are plotted in Figure 4b. The same figure also depicts the optimal cost of the robust tie-line scheduling problem, which naturally equals the maximum among the costs from P_2 . And for each sample, the gap between the optimal costs of P_1 and P_2 captures the cost due to lack of foresight. Figure 4b reveals that such costs can be significant. The median run-time of P_1 was 48.5ms over all samples. The

¹¹The termination condition $v^* = 0$ is replaced by checking that the Euclidean norm of a suitably normalized v^* is less than a threshold.

run-time for the robust problem was 458.2ms – roughly 10 times that median.

5.2 On a three-area 187-bus system test

For this case study, we interconnected the IEEE 30, 39, and 118 bus test systems as shown in Figure 5a. All transmission capacities were set to 100MW. Five wind generators were added to the 118-bus system (at buses 17, 38, 66, 88, and 111), three in the 39-bus system (at buses 3, 19, and 38), and two in the 30-bus system (at buses 11, and 23). Again, we adopted the same possible set of available wind power generations and power demands, as well as the cost structures as in Section 5.1. In total, our robust tie-line scheduling problem modeled 151 uncertain variables. For this multi-area power system, Algorithm 2 converged in the first iteration. The mixed integer programs in step 7 yielded the global optimal solution for each area, taking 62ms, 109ms, and 281ms, respectively. We again sampled the set $\Xi_1 \times \Xi_2 \times \Xi_3$ 3000 times, and solved P_1 . The run-time of Algorithm 2 was 825.3ms, that is roughly 1.8 times the median run-time of P_1 , given by 450.8ms.

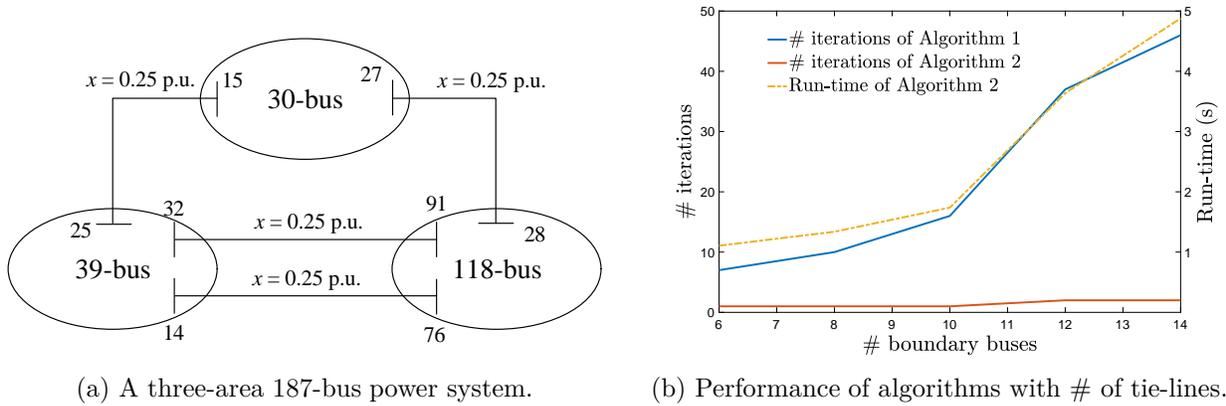


Figure 5: The power system model and how our algorithms perform with variation in number of tie-lines in a three-area 187-bus power system.

We studied how our algorithm scales with the number of boundary buses by adding more tie-lines to the same system. The aggregate iteration count of Algorithm 1 is expected to grow with the number of induced critical regions, that in turn should grow with the boundary bus count. On the other hand, the iteration count of Algorithm 2 largely depends on the initial choice of the scenario encoded in the sets $\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3$, and thus, varies to a lesser extent on the same count. Figure 5b validates these intuitions.

5.3 Summary of results from other case-studies

We compared Algorithm 1 with a dual decomposition based approach proposed in [5]. That algorithm converges asymptotically, while our method converges in finitely many iterations. Table 2 summarizes the comparison.¹² Compared to that in [5], our algorithm clocked lesser number of iterations and lower run-times in our experiments.

¹²We say the method in [5] converges when the power flow over each tie-line as calculated by the areas at its end mismatches by < 0.01 p.u..

Items	Two-area 44-bus system	Three-area 187-bus system
# iterations in Algorithm 1	8	9
# iterations of [5]	23	78
Run-time of Algorithm 1 (ms)	458.2	825.3
Run-time of [5] (ms)	779.8	1227.5

Table 2: Comparison with the method in [5].

Apart from the two systems considered so far, we ran Algorithm 2 on a collection of other multi-area power systems, details of which can be found in Appendix E. The results are summarized in Table 3. Our experiments reveal that Algorithm 2 often converges within 1 – 4 iterations. The run-time of Algorithm 2 grows significantly with the number of uncertain parameters. The 418-bus and the 536-bus systems with 422 and 546 uncertain variables, respectively, corroborate that conclusion. Such growth in run-time is expected because the complexity of (14) grows with the number of binary decision variables that equals the number of uncertain parameters. Run-time of a joint multi-area optimal power flow problem with a sample scenario in the last column provides a reference to compare run-times for the robust one.

# areas	# buses	# uncertain variables	# boundary buses	# iter. in Algorithm 2	Run-time of Algorithm 2 (ms)	Run-time of joint problem (ms)
2	87	91	4	1	719.6	310.0
2	175	179	4	1	871.1	340.5
2	236	240	4	1	1732.6	391.5
2	418	42	10	1	1020.7	455.7
2	418	422	10	4	6124.5	461.4
3	354	360	12	3	4127.4	655.8
3	536	54	12	1	2557.6	699.7
3	536	546	12	3	18359.8	701.2

Table 3: Performance of Algorithm 2 on various multi-area power system examples provided in Appendix E.

6 Conclusion

This work presented an algorithmic framework to solve a tie-line scheduling problem in multi-area power systems. Our method requires a coordinator to communicate with the system operators in each area to arrive at an optimal tie-line schedule. In the deterministic setting, where the demand and supply conditions are assumed known during the scheduling process, our method (Algorithm 1) was proven to converge in finitely many steps. In the case with uncertainty, we proposed a method (Algorithm 2) to solve the robust variant of the tie-line scheduling problem. Again, our method was shown to converge in finitely many steps. Our proposed algorithms do not require the system operator to reveal the dispatch cost structure, network parameters or even the support set of uncertain demand and supply within each area to the coordinator. We empirically

demonstrated the efficacy of our algorithms on various multi-area power system examples. Future directions include extending our decision framework to a multi-period setting considering unit commitment decisions [21], reserve sharing decisions [22], and allowing for asynchronous updates from neighboring SOs [23].

References

- [1] “MISO/PJM interchange transaction optimization overview,” Nov. 2010, [ONLINE], Available: <http://www.pjm.com/~media/committees-groups/stakeholder-meetings/pjm-miso-interchange/20111109/20111109-item-01-interchange-optimization-overview.ashx>.
- [2] M. White and R. Pike, “ISO New England and New York ISO interã regional interchange scheduling: Analysis and options,” *ISO White Paper.*, Jan. 2011.
- [3] B. Kim and R. Baldick, “Coarse-grained distributed optimal power flow,” *IEEE Trans. on Power Syst.*, vol. 12, no. 2, pp. 932–939, May 1997.
- [4] A. Conejo and J. Aguado, “Multi-area coordinated decentralized DC optimal power flow,” *IEEE Trans. on Power Syst.*, vol. 13, no. 4, pp. 1272–1278, Nov. 1998.
- [5] A. Bakirtzis and P. Biskas, “A decentralized solution to the DC-OPF of interconnected power systems,” *IEEE Trans. on Power Syst.*, vol. 18, no. 3, pp. 1007–1013, Aug. 2003.
- [6] Y. Guo, L. Tong, W. Wu, B. Zhang, and H. Sun, “Coordinated multi-area economic dispatch via critical region projection,” *IEEE Trans. on Power Syst.*, vol. 32, no. 5, pp. 3736–3746, Jul. 2017.
- [7] Z. Li, W. Wu, B. Zhang, and B. Wang, “Decentralized multi-area dynamic economic dispatch using modified generalized benders decomposition,” *IEEE Trans. on Power Syst.*, vol. 31, no. 1, pp. 526–538, Jan. 2016.
- [8] F. Zhao, E. Litvinov, and T. Zheng, “A marginal equivalent decomposition method and its application to multi-area optimal power flow problems,” *IEEE Trans. on Power Syst.*, vol. 29, no. 1, pp. 53–61, Jan. 2014.
- [9] A. Ahmadi-Khatir, A. J. Conejo, and R. Cherkaoui, “Multi-area unit scheduling and reserve allocation under wind power uncertainty,” *IEEE Trans. on Power Syst.*, vol. 29, no. 4, pp. 1701–1710, Jul. 2014.
- [10] Y. Ji, T. Zheng, and L. Tong, “Stochastic interchange scheduling in the real-time electricity market,” *IEEE Trans. on Power Syst.*, vol. 32, no. 3, pp. 2017–2027, May 2017.
- [11] Z. Li, W. Wu, M. Shahidehpour, and B. Zhang, “Adaptive robust tie-line scheduling considering wind power uncertainty for interconnected power systems,” *IEEE Trans. on Power Syst.*, vol. 31, no. 4, pp. 2701–2713, Jul. 2016.
- [12] F. Borrelli, *Constrained optimal control of linear and hybrid systems.* Springer, 2003, vol. 290, pp. 31–43.

- [13] B. Zeng and L. Zhao, “Solving two-stage robust optimization problems using a column-and-constraint generation method,” *Operations Research Letters*, vol. 41, no. 5, pp. 457–461, 2013.
- [14] T. Erseghe, “Distributed optimal power flow using ADMM,” *IEEE Trans. on Power Syst.*, vol. 29, no. 5, pp. 2370–2380, Sep. 2014.
- [15] S. MagnÅżsson, P. C. Weeraddana, and C. Fischione, “A distributed approach for the optimal power-flow problem based on ADMM and sequential convex approximations,” *IEEE Trans. on Control of Network Syst.*, vol. 2, no. 3, pp. 238–253, Sep. 2015.
- [16] G. Dantzig, *Linear programming and extensions*. Princeton university press, 2016.
- [17] M. Conforti, G. Cornu ejols, and G. Zambelli, *Integer programming*. Springer, 2014, vol. 271.
- [18] “IBM ILOG CPLEX optimization studio v12.5.1 documentation,” [ONLINE], Available: https://www.ibm.com/support/knowledgecenter/SSSA5P_12.5.1/maps/ic-homepage.html.
- [19] “Power systems test case archive,” [ONLINE], Available: <http://www2.ee.washington.edu/research/pstca/>.
- [20] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, “Matpower: Steady-state operations, planning, and analysis tools for power systems research and education,” *IEEE Trans. on Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.
- [21] A. Lorca and X. A. Sun, “Adaptive robust optimization with dynamic uncertainty sets for multi-period economic dispatch under significant wind,” *IEEE Trans. on Power Syst.*, vol. 30, no. 4, pp. 1702–1713, Jul. 2015.
- [22] M. A. Bucher, S. Chatzivasileiadis, and G. Andersson, “Managing flexibility in multi-area power systems,” *IEEE Trans. on Power Syst.*, vol. 31, no. 2, pp. 1218–1226, Mar. 2016.
- [23] Y. Ji and L. Tong, “Multi-area interchange scheduling under uncertainty,” *IEEE Trans. on Power Syst.*, vol. PP, no. 99, pp. 1–1, 2017.

A How SO_i can compute \mathcal{P}_i^y , α_i^y , β_i^y

With $\xi_i \in \Xi_i$ and $y \in \mathcal{Y}$ fixed, consider the optimization problem described in (8). Suppose the optimal solution $x_i^*(y, \xi_i)$ is unique. We suppress the dependency on (y, ξ_i) for notational convenience. Distinguish between the constraints that are *active* (met with an equality) versus that are *inactive* at optimality with the subscript \mathcal{A} and \mathcal{I} , respectively, as follows.

$$\begin{aligned} [\mathbf{A}_i^x]_{\mathcal{A}} x_i^* + [\mathbf{A}_i^{\xi}]_{\mathcal{A}} \xi_i + [\mathbf{A}_i^y]_{\mathcal{A}} y &= [b_i]_{\mathcal{A}}, \\ [\mathbf{A}_i^x]_{\mathcal{I}} x_i^* + [\mathbf{A}_i^{\xi}]_{\mathcal{I}} \xi_i + [\mathbf{A}_i^y]_{\mathcal{I}} y &< [b_i]_{\mathcal{I}}. \end{aligned}$$

The set of active versus inactive constraints remains the same over the critical region \mathcal{P}_i^y . Assuming $[\mathbf{A}_i^x]_{\mathcal{A}}$ is a square and invertible matrix, the optimal solution x_i^* is unique for each $z \in \mathcal{P}_i^y$, given by

$$x_i^* = [\mathbf{A}_i^x]_{\mathcal{A}}^{-1} \left([b_i]_{\mathcal{A}} - [\mathbf{A}_i^{\xi}]_{\mathcal{A}} \xi_i - [\mathbf{A}_i^y]_{\mathcal{A}} z \right).$$

The inequalities for the inactive constraints, together with the above relation defines the critical region $\mathcal{P}_i^y := \{z \in \mathcal{Y} : \mathbf{D}z \leq d\}$, where

$$\begin{aligned} \mathbf{D} &= -[\mathbf{A}_i^x]_{\mathcal{I}}[\mathbf{A}_i^x]_{\mathcal{A}}^{-1}[\mathbf{A}_i^y]_{\mathcal{A}} + [\mathbf{A}_i^y]_{\mathcal{I}}, \\ d &= [b_i]_{\mathcal{I}} - [\mathbf{A}_i^\xi]_{\mathcal{I}} \xi_i - [\mathbf{A}_i^x]_{\mathcal{I}}[\mathbf{A}_i^x]_{\mathcal{A}}^{-1} \left([b_i]_{\mathcal{A}} - [\mathbf{A}_i^\xi]_{\mathcal{A}} \xi_i \right). \end{aligned}$$

Finally, $J_i^*(y, \xi_i) = c_i(x_i^*, \xi_i)$ yields

$$\begin{aligned} \alpha_i^y &= -[c_i^x]^\top [\mathbf{A}_i^x]_{\mathcal{A}}^{-1} [\mathbf{A}_i^y]_{\mathcal{A}}, \\ \beta_i^y &= c_i^0 + [c_i^\xi]^\top \xi_i + [c_i^x]^\top [\mathbf{A}_i^x]_{\mathcal{A}}^{-1} \left([b_i]_{\mathcal{A}} - [\mathbf{A}_i^\xi]_{\mathcal{A}} \xi_i \right). \end{aligned}$$

The above expressions are derived under the premise that $[\mathbf{A}_i^x]_{\mathcal{A}}$ is invertible. We refer the reader to [12, Sections 7.2.2, 7.2.4] for the procedure in the general case.

B Proof of Theorem 1

After each iteration of Algorithm 1, y^* is a vertex of a critical region induced by the aggregate optimal cost. Also, \mathcal{D} is such that $\text{conv}(\mathcal{D}) \subseteq \partial J^*(y^*, \xi_1, \xi_2)$. Therefore, if the algorithm terminates with $v^* = 0$, then

$$0 \in \text{conv}(\mathcal{D}) + N_{\mathcal{Y}}(y^*) \subseteq \partial J^*(y^*, \xi_1, \xi_2) + N_{\mathcal{Y}}(y^*).$$

That is, y^* optimally solves (7). Next, we argue that the algorithm terminates in finitely many iterations.

Consider the sequence of y^* 's and J^* 's produced by the algorithm. Notice that J^* is a piecewise constant but non-increasing sequence. Further, a change in y^* always accompanies a strict decrease in J^* . Therefore, if y^* changes in an iteration from a certain point, that same point can never become y^* again. Since there are finitely many critical regions with finitely many vertices, it only remains to show that y^* cannot remain constant over infinitely many iterations. Towards that goal, notice that y^* can only belong to a finite number of critical regions. In the rest of the proof, we argue that the variable y computed in step 13 always belongs to a different such critical region containing y^* , unless the algorithm terminates.

At an arbitrary iteration, assume that y has taken values in critical regions $\mathcal{P}^1, \dots, \mathcal{P}^{\ell_D}$ that contain y^* . For convenience, let the optimal aggregate cost be given by $[\alpha^j]^\top z + \beta^j$ for $z \in \mathcal{P}^j$ for each $j = 1, \dots, \ell_D$. Thus, $\mathcal{D} := \{\alpha^1, \dots, \alpha^{\ell_D}\}$. Then, the new value of y is computed as $y^* - \varepsilon v^*$, with v^* as defined in (11). If $v^* = 0$, then the algorithm terminates, proving our claim. Otherwise, assume that $y^* - \varepsilon v^* \in \mathcal{P}^1$, contrary to our hypothesis, implying

$$J^*(y^* - \varepsilon v^*, \xi_1, \xi_2) = [\alpha^1]^\top (y^* - \varepsilon v^*) + \beta^1 = J^*(y^*, \xi_1, \xi_2) - \varepsilon [\alpha^1]^\top v^*.$$

Since, y^* optimizes the aggregate cost over \mathcal{P}^1 , it suffices to show that $[\alpha^1]^\top v^* > 0$ to arrive at a contradiction. For convenience, define the matrix $\boldsymbol{\alpha} := (\alpha^1, \dots, \alpha^{\ell_D})$. We prove more generally that $\boldsymbol{\alpha}^\top v^* > 0$. Associate Lagrange multipliers ϕ, ψ with the equality constraints $v = \boldsymbol{\alpha}\eta + \zeta$, and $\mathbf{1}^\top \eta = 1$, respectively. Also, associate μ_η, μ_ζ with the inequality constraints $\eta \geq 0$ and $\mathbf{K}^y \zeta \geq 0$, respectively. Then, an optimal primal-dual solution pair given by v^*, η^*, ζ^* and $\phi^*, \psi^*, \mu_\eta^*, \mu_\zeta^*$ satisfies the Karush-Kuhn-Tucker (KKT) optimality conditions – comprised of the constraints in (11) and the following

relations.

$$\begin{aligned} v^* - \phi^* &= 0, \quad \boldsymbol{\alpha}^\top \phi^* + \psi^* \mathbf{1} - \mu_\eta^* = 0, \quad \phi^* - [\mathbf{K}^y]^\top \mu_\zeta^* = 0, \\ [\mu_\eta^*]^\top \eta &= 0, \quad [\mu_\zeta^*]^\top \zeta^* = 0, \quad \mu_\eta^* \geq 0, \quad \mu_\zeta^* \geq 0. \end{aligned}$$

Using the KKT conditions, we have

$$\begin{aligned} \|v^*\|_2^2 + \psi^* &= (\phi^*)^\top (\boldsymbol{\alpha} \eta^* + \zeta^*) + \psi^* \\ &= (\boldsymbol{\alpha}^\top \phi^* + \psi^* \mathbf{1} - \mu_\eta^*)^\top \eta^* + (\phi^* - [\mathbf{K}^y]^\top \mu_\zeta^*)^\top \zeta^* \\ &= 0. \end{aligned}$$

Thus, $\psi^* < 0$. Together with the KKT conditions, that yields

$$\boldsymbol{\alpha}^\top v^* = \boldsymbol{\alpha}^\top \phi^* = -\psi^* \mathbf{1} + \mu_\eta^* > 0.$$

C Proof of Lemma 2

Strong duality of the problem described in (8) implies that $J_i^*(y, \xi_i)$ equals the optimum of the following problem.

$$\begin{aligned} \text{maximum}_{\lambda \in \mathbf{R}_+^{m_i}} \quad & c_i^0 + [c_i^\xi]^\top \xi_i + (\mathbf{A}_i^y y + \mathbf{A}_i^\xi \xi_i - b_i)^\top \lambda, \\ \text{subject to} \quad & c_i^x + [\mathbf{A}_i^x]^\top \lambda = 0. \end{aligned}$$

Then, maximizing $J_i^*(y, \xi_i)$ over the vertices of Ξ_i , described by $\{\xi_i^L + \boldsymbol{\Delta}_i^\xi w_i : w_i \in \{0, 1\}^{n_i}\}$, is equivalent to

$$\begin{aligned} \text{maximize} \quad & c_i^0 + [c_i^\xi]^\top \xi_i^L + (\mathbf{A}_i^y y + \mathbf{A}_i^\xi \xi_i^L - b_i)^\top \lambda + \mathbf{1}^\top \rho, \\ \text{subject to} \quad & c_i^x + [\mathbf{A}_i^x]^\top \lambda = 0, \\ & \rho = \text{diag}(w_i) \cdot \boldsymbol{\Delta}_i^\xi (c_i^\xi + [\mathbf{A}_i^\xi]^\top \lambda). \end{aligned}$$

over $w_i \in \{0, 1\}^{n_i}$, $\rho \in \mathbf{R}^{n_i}$, and $\lambda \in \mathbf{R}_+^{m_i}$. Here, $\text{diag}(w_i)$ denotes the diagonal matrix with w_i as the diagonal. Since we maximize $\mathbf{1}^\top \rho$, one can replace the second equality constraint in the above problem with the inequality

$$\rho \leq \text{diag}(w_i) \cdot \boldsymbol{\Delta}_i^\xi (c_i^\xi + [\mathbf{A}_i^\xi]^\top \lambda),$$

that is further equivalent to

$$\rho \leq M w_i, \quad \text{and} \quad \rho \leq M (\mathbf{1} - w_i) + \boldsymbol{\Delta}_i^\xi (c_i^\xi + [\mathbf{A}_i^\xi]^\top \lambda),$$

for a large enough $M > 0$. That completes the proof.

D Proof of Theorem 2

Let J^{rob} denote the optimal aggregate cost of (13). Then, J^* from step 3 and $J_1^{\text{opt}} + J_2^{\text{opt}}$ from step 7 at any iteration of Algorithm 2 satisfy

$$J^* \leq J^{\text{rob}} \leq J_1^{\text{opt}} + J_2^{\text{opt}}.$$

If Algorithm 2 terminates, the termination condition implies that the above inequalities are all equalities. In that event, y^* optimally solves (13).

To argue the finite-time convergence, notice that at least one among \mathcal{V}_1 and \mathcal{V}_2 increases in cardinality unless the termination condition is satisfied. The rest follows from the fact that Ξ_1 and Ξ_2 have finitely many vertices.

E Power system details for additional simulations

The multi-area power systems considered in Section 5.3 are given in Figure 6. Tie-line capacities were set to 100MW and their reactances were set to $0.25p.u.$ Capacity limits on the transmission lines within each area were set to their respective nominal values in Matpower [20] wherever present, and to 100MW, otherwise. For all two-area tests, two wind generators were installed in the two areas at buses 6 and 14 in area 1 and buses 11 and 23 in area 2. For the three-area tests, we replicated the placements described in Section 5.2. Power demands and available wind generations were varied the same way as in Sections 5.1 and 5.2.

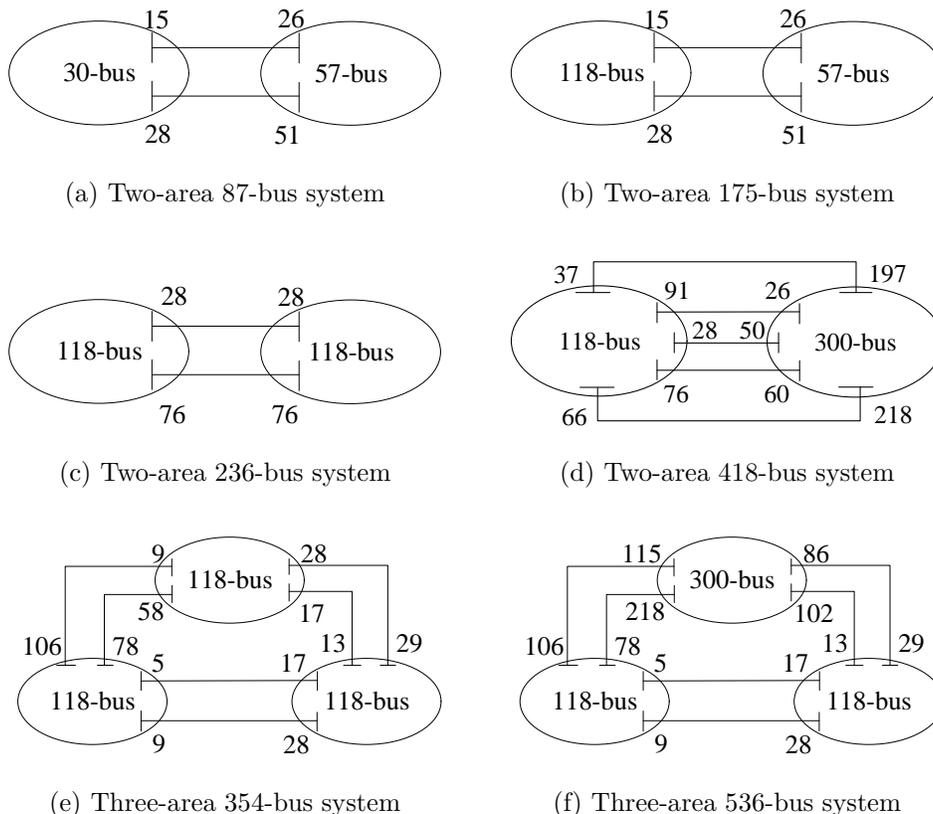


Figure 6: Additional power system examples considered for numerical experiments.