A Systematic Formulation Tightening Approach for Unit Commitment Problems

Bing Yan, Member, IEEE, Peter B. Luh, Life Fellow, IEEE, Tongxin Zheng, Senior Member, IEEE, Dane Schiro, Member, IEEE, Mikhail A. Bragin, Member, IEEE, Feng Zhao, Member, IEEE, Jinye Zhao, Member, IEEE, and Izudin Lelic, Member, IEEE

Abstract—Unit Commitment is usually formulated as a Mixed Binary Linear Programming (MBLP) problem. When considering a large number of units, state-of-the-art methods such as branch-and-cut may experience difficulties. To address this, an important but much overlooked direction is formulation transformation since if the problem constraints can be transformed to directly delineate the convex hull in the data pre-processing stage, then a solution can be obtained by using linear programming methods without combinatorial difficulties. In the literature, a few tightened formulations for single units with constant ramp rates were reported without presenting how they were derived. In this paper, a systematic approach is developed to tighten formulations in the data pre-processing stage. The idea is to derive vertices of the convex hull without binary requirements. From them, vertices of the original convex hull can be innovatively obtained. These vertices are converted to tightened constraints, which are then parameterized based on unit parameters for general use, tremendously reducing online computational requirements. By analyzing short-time horizons, e.g., two or three hours, tightened formulations for single units with constant and generation-dependent ramp rates are obtained, beyond what is in the literature. Results based on the IEEE 118-bus and Polish 2383-bus systems demonstrate computational efficiency and solution quality benefits of formulation tightening. The approach is general and has great potential for tightening complicated MBLP problems in power systems and beyond.

Index Terms—Unit commitment, mixed binary linear programming, branch-and-cut, formulation tightening.

I. INTRODUCTION

Unit Commitment (UC) is an important problem faced by independent system operators. The problem is to minimize the total commitment and dispatch cost by committing appropriate units while satisfying demand and other constraints [1]. It is usually formulated as a Mixed Binary Linear Programming (MBLP, with binary and continuous variables and a linear structure) problem, and is believed to be NP hard. To solve such problems, industrial state-of-the-practice is to use commercial solvers that are mostly based on branch-and-cut combined with heuristics. In the method, all integrality requirements on binary variables are first relaxed, and the Linear Programming (LP) relaxation problem is solved by using LP methods. If all binary variables have binary values, the solution is optimal to the original problem. If not, valid cuts are added, trying to obtain the convex hull (the smallest convex set that contains all feasible solutions [2]) of the original problem. If successful, the problem can be solved by using LP without combinatorial difficulties. If not, the method relies on time-consuming branching operations. In the solvers, cuts are performed online by using existing types of cuts, and most of them are data dependent. Since the cuts have coefficients in numerical values and cannot be reused, the solvers generate cuts again when solving the problem with other data sets. For problems with a large number of units or complicated units such as combined cycle units with generation dependent ramp rates, the commercial solvers may experience difficulties.

To obtain UC solutions with quantifiable quality fast, most researchers focus on solution methodologies. An important but much overlooked direction is formulation transformation since if problem constraints can be transformed to directly delineate the convex hull (i.e., the formulation is “tight”) in the data pre-processing stage, then a solution can be obtained by using LP methods without combinatorial difficulties [3]. With resulting constraints reused for other data sets, online computational requirements are tremendously reduced. However, this formulation tightening process is fundamentally difficult. Given a problem formulation, it is difficult to obtain the convex hull, and there are no systematic ways to transform constraints. In the literature, a few tightened formulations for single units with constant ramp rates established in the data pre-processing stage were reported without presenting how they were derived as reviewed in Section II. They were shown computationally efficient for overall UC problems. Single-unit formulations were also tightened online in the problem solving process with expensive computations based on optimal LP solutions.

In this paper, a systematic approach is developed to tighten formulations in the data pre-processing stage. Our idea is first to apply existing cuts that are relevant, data independent and easily implementable based on constraint characteristics in Section III. More importantly, tightened constraints are established based on novel integration of “constraint-and-vertex conversion,” “vertex elimination” and “parameterization” in four steps in Section IV. For a unit with given parameters (e.g., minimum/maximum generation levels and ramp rate) in numerical values, the first step is to relax integrality requirements, and generate vertices from constraints. The second step is to eliminate vertices with...
fractional values for binary variables. The remaining vertices will be proved to be vertices of the convex hull to the original MBLP problem. They are converted back to tight constraints in the third step. To make tight constraints reusable, our idea is to covert numerical coefficients to unit parameters in the last step. This parameterization process is done through analyzing constraints and relationships between numerical coefficients and unit parameters, and then verified by checking constraint physical meanings. For practical applications, our idea is to obtain “near-tight” formulations by analyzing short-time horizons, e.g., two and three hours. With units categorized by how long it takes to reach from the minimum generation level to the maximum, tightened constraints are developed for each category and a look-up table is established. Then for each unit, tightened constraints can be identified through table lookup based on unit parameters in the data pre-processing stage.

In Section V, three examples are presented. The first is to obtain tight formulations for a three-hour problem for units with constant ramp rates and a one-hour problem for units with generation dependent ramp rates, to discuss unit categories, and to demonstrate tightness. Resulting tightened constraints are beyond what is in the literature. The second IEEE 118-bus problem is to show impacts of tightening units with generation dependent ramp rates. The last Polish 2383-bus problem is to demonstrate performance of tightening units with constant ramp rates. Results demonstrate great potential for tightening complicated MBLP problems in power systems and beyond.

II. LITERATURE REVIEW

In the literature, most studies focused on single units without system-level constraints in view of complexity. For single units, there are mainly three types of formulations: 1-binary (1-bin) for unit on/off, 2-binary (2-bin) for on/off and start-up, and 3-binary (3-bin) for on/off, start-up and shut-down. Based on these formulations, tightening was performed in the data pre-processing stage or online as reviewed below.

Pre-process. In [4 - 7], new cuts were developed based on restrictions on binary variables to tighten formulations on top of the original constraints. As developing new cuts is not the focus of this paper, it is not elaborated. In terms of rewriting constraints, a 1-bin formulation with start-up/shut-down and minimum (min) up/down time constraints was considered in [8]. New start-up/shut-down constraints were presented based on a 7-hour problem. It was proved that constraints directly delineate the convex hull. With commercial solvers, testing results of 20-32 unit UC problems showed that computational time was significantly reduced as compared to that in [9]. Based on a 3-bin formulation, a new set of tightened ramp rate constraints for the first- and last-operation hour was reported in [3]. Results of 5-unit problems showed that computational time was significantly reduced as compared to those in [10, 11].

With more constraints, short periods were considered. Based on a 3-bin formulation with capacity, ramp rate, and min up/down time constraints, new ramp rate constraints for a two-hour problem were presented [12]. The single-unit formulation is tight when unit parameters satisfy certain conditions. Similar ramp rate and min up/down time constraints were reported for a three-hour problem in [13]. Based on a 2-bin formulation, combined ramp rate and min up/down time constraints were presented for two/three-hour problems under various parameter conditions in [14]. In [12 - 14], under specific assumptions on unit parameters, formulations were proved tight for problems with short-periods, and were shown computationally efficient by using branch-and-cut for overall UC problems.

The above tightened formulations were presented without explaining how they were obtained. Built on [14], assuming a unit is off for certain time, a tight single-unit formulation was derived by dynamic programming to max profits [15], without numerical results. When using branch-and-cut, however, there are no prices and assumptions on units may not be easy to drop.

Ramp rates were constants in the above. For generation-dependent ramp rates, they were modeled as converted ramp time curves in [16], while performance decreases drastically as problem sizes increase. In our previous work, ramp and reserve capability functions were established, and the formulation was improved by convex hull analysis [17]. Results by branch-and-cut show improved performance as compared to [16]. Results by a decomposition and coordination approach show much reduced branching time as compared with branch-and-cut. A few preliminary tightened constraints for units with constant ramp rates were presented in [18], without numerical results.

Tightening of reserve and generation dependent ramp rates was rarely discussed in the literature. To the best of our knowledge, there is no systematic approach in data pre-process.

Online. In [19], with a 3-bin formulation, a LP relaxation problem was solved first. Then cuts were generated as a callback for individual units based on the LP solution if it is infeasible to the unit. These cuts were given to the solver, and the original problem was solved with cuts. Results on 900-unit UC problems showed that computational time was reduced by 19% on average. For online tightening, cuts obtained for one unit cannot be reused for other units, and online computations are expensive as compared to tightening in data pre-process.

III. SINGLE-UNIT FORMULATION AND EXISTING CUTS

Assuming system-level constraints are relaxed, a single-unit UC problem is formulated in Subsection A. Then existing cuts are applied in Subsection B.

A. Formulate single-unit UC problems [1]

For a unit, the main decision variables at each time t include binary commitment decisions on/off x and startup u, and continuous dispatch decision p. For illustration purposes, a 2-bin formulation is adopted in this paper (performance of this 2-bin and a 3-bin formulation in the literature will be compared later in Section V). With different values of x and u, the unit has four statuses at each t, i.e., on, off, start-up and shut-down. Corresponding initial conditions include x(0), u(0) and p(0). Constraints are generation capacity, offer price block, ramp rate, start-up, minimum up/down time, and reserve.

1) Generation capacity

When a unit is online, generation level p should be within its minimum pmin (MW) and maximum pmax (MW); otherwise, p is zero, i.e.,

\[ x(t)P_{\text{min}} \leq p(t) \leq x(t)P_{\text{max}}, \forall t. \]  

(1)

2) Offer price block

Generation cost is usually a piecewise function of p. To maintain linearity, a few offer blocks are considered with constant prices in each block (assume prices are monotonically
non-decreasing). For individual blocks, a new continuous decision variable $p_b(t)$ is needed, and their sum equals $p(t)$, i.e.,

$$p_b(t) \leq P_{mb}^b, \forall b, \forall t; \sum_b p_b(t) = p(t), \forall t,$$

(2-3)

where $P_{mb}^b$ (MW) is the maximum generation of block $b$.

3) Ramp rate

Ramp rate constraints require that the change of generation levels between two consecutive time periods cannot exceed ramp rate $R$ (MW/hour). If the unit cannot reach $P_{mb}^b$ in one hour, it is assumed that $p$ cannot exceed $P_{mb}^b$ plus 30-minute ramp upon starting up or at shutting down following the standard industrial practice, i.e., $P_{mb}^b + R/2 \leq P_{mb}$. Ramp rate constraints are formulated in a linear way below,

$$p(t) - p(t-1) \leq Rx(t-1) + (P_{mb}^b + R/2)(x(t) - x(t-1)), \forall t, (4)$$

$$p(t-1) - p(t) \leq Rx(t) + (P_{mb}^b + R/2)(x(t-1) - x(t)), \forall t. (5)$$

When $P_{mb}^b + R/2 > P_{mb}^b$, the above constraints are not needed.

For units with generation dependent ramp rates, ramp capability functions can be represented by $SOS2$ [17]. With a set of binary SOS2 variables $\alpha_{ub}^{ij}(t)$ and a set of continuous nonnegative weight variables $\alpha_{ub}^{ij}(t)$, ramp-up capability function $R_{ub}^{ij}(t)$ is represented below,

$$R_{ub}^{ij}(t) = R_0 \alpha_{ub}^{ij}(t) + R_1 \alpha_{ub}^{ij}(t) + R_2 \alpha_{ub}^{ij}(t) + \alpha_{ub}^{ij}(t) + 0 \alpha_{ub}^{ij}(t),$$

$$p(t) = 0 \alpha_{ub}^{ij}(t) + P_1 \alpha_{ub}^{ij}(t) + P_2 \alpha_{ub}^{ij}(t) + P_3 \alpha_{ub}^{ij}(t) + P_4 \alpha_{ub}^{ij}(t),$$

$$\alpha_{ub}^{ij}(t) \geq \alpha_{ub}^{ij}(t), 0 \leq \alpha_{ub}^{ij}(t) \leq 1, \alpha_{ub}^{ij}(t) [0, 1], 1 \leq m \leq 5,$$

$$\sum_n \alpha_{ub}^{ij}(t) = 1, \sum_n \alpha_{ub}^{ij}(t) = 2.$$

Non-zero $\alpha_{ub}^{ij}(t)$ must be consecutive in the ordering, and these standard constraints are omitted. Ramp-down $R_{ub}^{bd}(t)$ is modeled in a similar way. Then replace the first $R$ in Eq. (4) and (5) is replaced by $R_{ub}^{bd}(p)$ and $R_{ub}^{bd}(p)$, respectively, and replace the second $R$ by $R_{ub}$. Products of binary and continuous variables are linearized by the standard big M method.

As ramp up/down functions are derived from ramp rates (large blocks), they are related: if $R$ is at the 1st (2nd) block, ramp up/down must be at their 1st (2nd) block [17], i.e.,

$$\alpha_{ub}^{ij}(t) = \alpha_{ub}^{max}(t), \alpha_{ub}^{ij}(t) = \alpha_{ub}^{max}(t).$$

(7-8)

Further relations among $\alpha_{ub}^{ij}$ can be derived as follows [17],

$$\alpha_{ub}^{ij}(t) + \alpha_{ub}^{ij}(t) = 1, \alpha_{ub}^{ij}(t) + \alpha_{ub}^{ij}(t) + \alpha_{ub}^{ij}(t) = 1.$$
implies a new upper bound on \( y \), e.g., \( z = 0 \Rightarrow y \leq y_{\text{max}}(y_{\text{max}} < y_{\text{max}}) \). The idea is to merge \( z \) into \( y \leq y_{\text{max}} \) as follows,

\[
y + (y_{\text{max}} - y_{\text{min}})(1 - z) \leq y_{\text{max}}.
\]  

(19)

According to (19), if \( z = 1 \), \( y \leq y_{\text{max}} \); and if \( z = 0 \), \( y \leq y_{\text{max}} \).

For UC problems, implied bound cuts can be applied to the offer price block constraints as follows,

\[
p(t) + (1 - x(t)) P_{\text{min}, t} \leq P_{\text{max}, t}, \forall b, \forall t.
\]  

(20)

Eq. (20) guarantees that when \( x(t) = 1, p(t) \leq P_{\text{max}, t} \); and when \( x(t) = 0, p(t) = 0 \). Then Eq. (2) is replaced by Eq. (20).

Implied bound cuts can also be applied to the reserve constraints as Eq. (A1) and (A2) in Section A of Appendix, and Eq. (A15) and (A18) are replaced by (A1) and (A2), respectively. Note that the original constraints are replaced by cuts.

2) Mixed-integer rounding cuts \([24]\)

Mixed-integer rounding cuts apply rounding on coefficients of integer variables and the constant of a constraint. Consider a constraint \( z + y \leq b \) with integer variable \( z \) and continuous variable \( y \) (\( z \geq 0, y \geq 0 \)). If constant \( b \) is not an integer, then the convex hull of the LP-relaxed problem (LP-relaxed convex hull) has a non-integer vertex \( (b,0) \). To avoid this, a mixed-integer rounding cut that goes through the two points \((b,0),(b+1,1)\) and \((b+1,1)\) (floor function \( \lfloor \) \) gives the greatest integer \( \leq b \)) of the LP-relaxed convex hull is below,

\[
z - 1 - (b - \lfloor b \rfloor) \leq \lfloor b \rfloor.
\]  

(21)

A general version is expressed as Eq. (A3) in Section A of Appendix. For UC problems, mixed integer rounding cuts can be applied to the offer price block constraints as follows,

\[
\lfloor P_{\text{min}} \rfloor x(t) - \sum_{b} p(t) \leq 0, \forall t.
\]  

(22)

IV. ESTABLISH TIGHT CONSTRAINTS

The above single-unit UC formulation is further tightened in this section. Tightened constraints are established based on novel integration of “constraint-and-vertex conversion,” “vertex elimination” and “parameterization” in Subsection A, and a numerical example is presented in Subsection B. Tightness is proved in Subsection C.

A. Establishment of tight constraints

As mentioned earlier, a few constraints involve initial conditions, and this is problematic since there are many sets of possible initial conditions. This issue is first addressed in IV.A.1. Tightened constraint is established through four steps in IV.A.2. Unit categorization is then discussed in IV.A.3. The overall tightening process is summarized in IV.A.4.

1) Initial conditions

To overcome the difficulties caused by initial conditions, our idea is to treat them as decision variables without specifying their values, and can take any reasonable values. For a one-hour problem with capacity and start up constraints, and given initial conditions, it becomes a two-hour problem with additional ramp rate constraints. Although the convex hull with initial conditions treated as decision variables is generally larger than the one with specific initial values, the tightening process is significantly simplified.

2) Four-step tightening

For a unit with (1) given parameters \((P_{\text{min}}, P_{\text{max}}, R, T_{MU}, \text{and } T_{MD})\) in numerical values and (2) initial conditions \((x(0), u(0))\), \(p(0), T_{MOH}, \text{and } T_{MOB})\) as decision variables, tightened constraints are established through four steps as follows.

Step 1. Constraint-to-vertex conversion. The first step is to relax integrality requirements on binary variables. For the LP relaxed problem, generate vertices of the convex hull from constraints by using algebraic manipulation of unit parameters with algorithms well established based on Gaussian elimination \([25]\). For example, for a problem with two continuous decision variables and three inequality constraints, the vertices of the convex hull are generally the intersections of every two constraints. This conversion is performed by using existing software Porta \([26]\). Given constraints with coefficients in numerical values, Porta outputs vertices in numerical values.

Step 2. Vertex elimination. For clarity of discussion, a vertex with binary values for all binary variables is named a “binary vertex,” and a “fractional vertex” otherwise. All binary vertices are feasible to the original problem, while all fractional ones are infeasible. If all vertices obtained in Step 1 are binary vertices, the formulation is tight. If not, the second step is to eliminate the fractional vertices. The remaining vertices are the vertices of the convex hull to the original MBLP problem as will be proved later in Subsection IV.C.

To illustrate the idea, consider a simple Binary Linear Programming (BLP) problem with two binary variables \(x_1\) and \(x_2\), and \(x_1 + x_2 \geq 0.5\). In Fig. 2(a), constraints are color-coded by blue lines, and the convex hull Conv(P_{BLP}) by red lines. Vertices \(V_{BLP}\) of Conv(P_{BLP}) are represented by solid red dots. For the LP-relaxed problem in Fig. 2(b), constraints are color-coded by blue lines, and they delineate convex hull Conv(P_{RLP}) with vertices \(V_{RLP}\) represented by blue dots. There are two sets of vertices in \(V_{RLP}\). One set \(V_{RLP}^B\) consisting of binary vertices is represented by solid blue dots, and the other set \(V_{RLP}^F\) consisting of fractional vertices is represented by open blue dots. Given \(V_{RLP}\), how to get back to \(V_{BLP}\? The idea is to drop fractional vertices \(V_{RLP}^F\). The remaining binary vertices \(V_{RLP}^B\) in Fig. 2(b) are the same as vertices \(V_{BLP}\) in Fig. 2(a).

![Figure 2(a): Convex hull of a BLP problem with binary variables \(x_1, x_2\) and relaxed problem.](image)

Now consider an MBLP problem with binary variables \(x_1\) and \(x_2\), and continuous variable \(x_3\). In Fig. 3(a), constraints and convex hull Conv(P_{MBLP}) are color-coded by blue and red lines, respectively, and vertices \(V_{MBLP}\) of Conv(P_{MBLP}) are represented by solid red dots. For the LP-relaxed problem in Fig. 3(b), constraints are color-coded by blue lines, and they delineate convex hull Conv(P_{RLP}) with vertices \(V_{RLP}\) represented by blue dots. By dropping fractional vertex \(V_{RLP}^F\) (open blue dots), remaining binary vertices \(V_{RLP}^B\) (solid blue dots) in Fig. 3(b) are the same as vertices \(V_{MBLP}\) (solid red dots) in Fig. 3(a).

Step 3. Vertex-to-constraint conversion. In this step, vertices obtained in Step 2 are converted back to tight constraints by using software Porta as a reverse process of that in Step 1. The resulting formulation should be tight as to be proved in IV.C.

Step 4. Parameterization. The constraints obtained above have coefficients in numerical values. To make them reusable...
to other units, our idea is to covert numerical coefficients to unit parameters. This parameterization process is done by analyzing these constraints and the relationships between numerical coefficients and unit parameters. It is then verified by checking physical meanings of the resulting constraints with coefficients in unit parameters under all possible combinations of binary variables as to be shown in IV.B.

3) Categorization

For different types of units, a few sets of tight formulations are developed through the above four steps. For units, ramp rate is an important physical characteristic, restricting the unit’s up/down movement from one hour to another and determining reserve capability. Thus one way to categorize units is through how long it takes to reach from $p_{min}$ to $p_{max}$. The simplest category is $p_{max} < p_{min} + R/2$ since in this case, the unit can reach $p_{max}$ upon starting up without requiring ramp rate requirements. The second is $p_{min} + R/2 \leq p_{max} < p_{min} + 3R/2$ since the maximum generation that the unit can reach in two hours is $p_{min} + 3R/2$ as shown in Fig. 4. Minimum up/down time requirements are not considered in unit categorization.

![Diagram](image)

Figure 3(a): Convex hull of an MBLP problem with binary $x_1$, $x_2$ and continuous $x_3$

Figure 3(b): Convex hull of its LP-relaxed problem

4) Overall tightening process

The process of generating tight constraints is summarized in Fig. 5. Given a UC problem, a set of tightened constraints for different unit types will be established by using the systematic approach before solving the problem. A look-up table that covers most unit types in the market is established based on unit categories. When solving the UC problem, tightened constraints will be identified based on unit parameters in the data pre-processing stage.

![Flow chart](image)

Figure 5: Flow chart of generating tight constraints

B. Numerical example

To illustrate the idea, consider a two-hour $(t-1, t)$ problem for a unit with $P_{min} = 59$ MW, $P_{max} = 111$ MW, and $R = 19$ MW/hour. Since one coefficient value can be the result of multiple manipulations of unit parameter values, prime values are used for unit parameters. Capacity (Eq. 1), ramp rate (Eq. (4) and (5)) and start up constraints (Eq. (11 - 13)) are denoted as C1 as shown in Fig. 6. Other constraints are not considered for simplicity, and ranges for binary variables ((0, 1)) are not presented for brevity. By constraint-to-vertex conversion, obtain vertices $V_1$, and the last 10 vertices are shown in Fig. 7 (variable sequence: $x(t-1), u(t-1), p(t-1), x(t), u(t), p(t)$). This conversion takes about 1 second by Porta.

![Table](image)

Figure 6: Constraints C1 of a two-hour problem

![Table](image)

Figure 7: Vertices $V_1$ of a two-hour problem

In $V_1$, there are 35 vertices, and 19 are binary vertices. For vertices (32-35), $u(t-1)$ should be 0 as generation levels exceed maximum limit for the first on hour (i.e., $p_{min} + R/2$), and this issue will be addressed later. Keep binary vertices as $V_2$ and convert to constraints C2 (about 1 second by Porta). Compared to C1, there are 4 new constraints as shown in Fig. 8 below.
To describe the parameterization process, take Constraint (2) with three variables in Fig. 8 as an example. In the original formulation in Fig. 6, coefficients before decision variables are linear functions of unit parameters, i.e., $P_{min} = 59$, $P_{max} = 111$, and $R = 19$. Since constraint-and-vertex conversion is based on Gaussian elimination, coefficients of tight constraints are also linear functions of unit parameters. For the value of 222 before $x(t)$ in Constraint (2), it is twice of $P_{max}$. For the value of 85 before $u(t)$, with prime unit parameters, it can be obtained that $85 = 2(111 - 59)$, and the function is $2(P_{max} - P_{min}) - R$. For the value of 2 before $p(t)$, it is just 2, and there is no need of parameterization. The entire constraint is parameterized as follows,

$$p(t) \leq P_{max} x(t) - (P_{max} - P_{min} - R / 2) u(t).$$

(23)

In the above, when $x(t) = u(t) = 0$, $p(t) \leq 0$; when $x(t) = 1$ and $u(t) = 0$, it represents the maximum generation level; and when $x(t) = u(t) = 1$, it represents the maximum generation limit on the first operation hour. With (23), the right-hand side of capacity constraint Eq. (1) can be deleted.

Constraints (10) and (11) in Fig. 8 can be treated as revised ramp rate constraints as compared to Eq. (4) and (5). With prime values for $P_{min}$, $P_{max}$, and $R$, numerical coefficients of constraints (10) and (11) in Fig. 8 are parameterized as follows,

$$p(t) - p(t - 1) \leq (P_{max} + R)x(t) - P_{max}x(t - 1) - (R/2)u(t),$$

$$p(t - 1) - p(t) \leq (P_{max} + R/2)x(t - 1) - (P_{max} - R/2)x(t) - (R/2)u(t).$$

(24-25)

The above constraints are the same as the ramp rate constraints developed in [14].

Physical meanings of Eq. (24) and (25) under all possible combinations of the binary variables are as shown in Table I. It can be seen that these two tight constraints are meaningful under all situations, and no feasible solutions will be cut by adding them.

<table>
<thead>
<tr>
<th>$x(t-1)$</th>
<th>$u(t)$</th>
<th>$p(t)$</th>
<th>$p(t-1)$</th>
<th>$p_{min}$</th>
<th>$p_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>$p(t) - p(t - 1) \leq 0$</td>
<td>$p(t - 1) - p(t) \leq 0$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$p(t) \leq P_{max} + R / 2$</td>
<td>$p(t - 1) \geq P_{max}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$p(t) \geq P_{max}$</td>
<td>$p(t - 1) \leq P_{max} + R / 2$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$p(t) - p(t - 1) \leq R$</td>
<td>$p(t - 1) - p(t) \leq R$</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Constraint (1) in Fig. 8 is parameterized as follows,

$$p(t - 1) - p(t) \leq -(P_{max} + R)x(t) - (P_{max} + R)x(t - 1) - R/2u(t),$$

$$+(P_{max} - P_{min} - R/2)x(t - 1) - u(t), \forall t.$$  

(26)

It can be verified that Eq. (26) is valid for all possible combinations of the three binary variables.

Add Eq. (23) and (26) in C2, apply Eq. (23) to $t - 1$, and replace ramp rate constraints Eq. (4) and (5) by Eq. (24) and (25). Denote the new constraint set as C3, and obtain vertices $V_3$. There are 20 vertices in $V_3$, and 17 binary ones. The issue in $V_1$ with $u(t-1)$ is addressed by applying Eq. (23) to $t - 1$. Keep binary vertices as $V_4$ and $V_4'$ and convert them to constraints C4. Two new tight constraints are obtained and parameterized below,

$$p(t - 1) - p(t) \leq -P_{max} x(t) + (P_{max} + R)x(t - 1) - R/2u(t - 1),$$

$$p(t) \leq (P_{max} + 3R/2)x(t) - Ru(t)$$

$$+(P_{max} - P_{min} - 3R/2)x(t - 1) - u(t - 1)).$$  

(27-28)

When $x(t - 1) = 1$ and $u(t - 1) = 0$, $x(t - 1) = u(t) = 0$, Eq. (28) is $p(t)$ \leq $P_{max} - P_{min} - 3R/2$. Here $P_{max} - P_{min} - 3R/2$ cannot be negative, and Eq. (27) and (28) are valid under all possible combinations of $x$ and $u$, otherwise feasible solutions would be cut off. Eq. (23-28) directly constrain variables at $t - 1$ and $t$, tighten the formulation, but can hardly be obtained manually. Among them, Eq. (23) and Eq. (26-28) are new tightened constraints, beyond what is in the literature. With Eq. (23-28), the above two-hour formulation is tight.

C. Tightness proof

The proof will be conducted in two steps for BLP and MBLP problems, respectively.

Step 1. Tightness of BLP problems

Tightness proof for BLP problems is based established on Theorem 1 introduced below.

**Theorem 1.** For a BLP problem, the set of binary vertices $V_{BLP}$ of its LP-relaxed convex hull $Conv(P_{BLP})$ is the set of vertices $V_{BLP}$ of original BLP convex hull $Conv(P_{BLP})$.

**Proof.** This proof follows by contradiction to prove that the binary vertices in $V_{BLP}$ remain in $V_{BLP}$ and integrality relaxation does not bring new binary vertices to $V_{BLP}$. For a binary vertex $v_1$ in $V_{BLP}$, assume there exists a nonzero vector $d$ (one element is $e$, $e > 0$, and others are 0) such that $v_1 \pm d \in Conv(P_{BLP})$. If the corresponding element of $e$ in $v_1$ is 1, then $1 + e > 1$, and $v_1 + d \notin Conv(P_{BLP})$. If it is 0, then $0 - e < 0$, and $v_1 - d \notin Conv(P_{BLP})$. Similar for $d$ with multiple non-zero elements. By contradiction, $d$ does not exist, thus the binary vertices in $V_{BLP}$ are still vertices of $Conv(P_{BLP})$ [28]. Assume that integrality relaxation brings a new binary vertex $v_2$ to $V_{BLP}$, i.e., $v_2 \in V_{BLP}$ but $v_2 \notin V_{BLP}$. However, $v_2$ is binary and satisfies all constraints, it must be feasible to the original BLP problem. For a $n$-dimensional unit hypercube, its vertex set is $\{0, 1\}^n$. With constraints, binary points still remain as vertices of the truncated hypercube [29], thus $v_2 \in V_{BLP}$. By contradiction, $v_2$ does not exist, thus integrality relaxation does not bring new binary vertices to $V_{BLP}$. Therefore Theorem 1 holds. End.

Based on Theorem 1, constraints converted from $V_{BLP}$ directly delineate $Conv(P_{BLP})$, i.e., the formulation is tight.

Step 2. Tightness of MBLP problems

Tightness proof for MBLP problems is established based on Theorem 2 introduced below.

**Theorem 2.** For an MBLP problem, the set of binary vertices $V_{MBLP}$ of its LP-relaxed convex hull $Conv(P_{MBLP})$ is the set of vertices $V_{MBLP}$ of original MBLP convex hull $V_{MBLP}$.

**Proof.** This proof is based on Theorem 1. For a binary vertex $v_1$ in $V_{MBLP}$, assume there exists a nonzero vector $d'$ such that $v_3 \pm d' \in Conv(P_{MBLP})$. Similar to Theorem 1, $d'$ does not exist, thus the binary vertices in $V_{MBLP}$ are still vertices of $Conv(P_{MBLP})$. Since integrality relaxation has effects on ranges of binary variables only and not on continuous variables, it will not bring new binary vertices in $V_{MBLP}$ following Theorem 1. Therefore Theorem 2 holds. End.
Based on Theorem 2, constraints converted from $V_{RAMBLB}$ directly delineate $Conv(P_{RAMBLB})$, i.e., the formulation is tight.

V. TESTING RESULTS

The above tightening approach has been implemented by using Porta [26], and the UC problems are solved by using commercial solver IBM ILOG CPLEX Optimization Studio V 12.8.0.0 [30], both on a PC with 2.90GHz Intel Core(TM) i7 CPU and 16G RAM. Porta is not embedded in CPLEX. Three examples are presented. The first is to obtain tight formulations for a three-hour problem for units with constant ramp rates and a one-hour problem for units with generation dependent ramp rates, to discuss unit categories, and to demonstrate tightness. The second IEEE 118-bus problem is to show impacts of tightening units with generation dependent ramp rates. The last Polish 2383-bus problem is to demonstrate performance of tightening units with constant ramp rates.

Example 1. Single-unit

a) Constant ramp rates: Three-hour

Add one hour ($t + 1$) to the example in Subsection IV.B. For the unit, $P_{\text{max}} - P_{\text{min}} \geq 5R/2$, it cannot reach $P_{\text{max}}$ in three hours. Minimum up/down time constraints are also added, and $T^{\text{MIN}} = T^{\text{MAX}} = 2$. To consider all possible initial conditions, the following Eq. (29 - 31) are used for minimum up/down time instead of general formulas for this specific example,

$$x(t-1) - x(t) + x(t+1) \geq 0,$$

$$2x(t-1) - x(t) + x(t+1) \leq 2, \quad u(t-1) - x(t) \leq 0.$$ (30-31)

With the standard formulation as C1, after constraint-to-vertex conversion (2 - 3 seconds by Porta), there are 654 vertices in $V_1$, and 45 are binary. Add the tight constraints obtained in the two-hour problem in Subsection IV.B to C1, denote the new constraint set as C2, and obtain vertices $V_2$. There are 314 vertices in $V_2$, with 40 binary ones. Keep binary vertices as $V_3$ and convert them to constraint C3 via vertex-to-constraint conversion (2 - 3 seconds by Porta). There are 17 new constraints. After parameterization, there are 14 tight constraints as three pairs have the same forms applying to different hours. Two of the 14 constraints only contain binary variables and can be generalized as revised minimum up/down time constraints as compared to Eq. (14),

$$\sum_{r=1}^{u_{\text{MIN}}} x(r) = T^{\text{MIN}}, \quad \sum_{r=1}^{u_{\text{MAX}}} u(r) \leq x(t), \quad T^{\text{MAX}} \leq t \leq T,$$

$$\sum_{r=1}^{x_{\text{MAX}}} (1 - x(r)) = T^{\text{MAX}}, \quad \sum_{r=1}^{x_{\text{MIN}}} u(r) \leq 1 - x(t - T^{\text{MAX}}), \quad T^{\text{MAX}} + 1 \leq t \leq T.$$ (32)

The above constraints are the same as the minimum up/down time constraints developed in [13].

It can be verified that the remaining 12 constraints are valid under all possible unit statuses. They further tighten the formulation on top of the tight constraints obtained in the two-hour problem. Take one of 12 as an example as follows,

$$p(t+1) \leq -(P_{\text{MAX}} - P_{\text{MIN}} - 5R/2)u(t-1) + (P_{\text{MAX}} - P_{\text{MIN}} - 5R/2)x(t) - (P_{\text{MAX}} - P_{\text{MIN}} - 3R/2)u(t) + (P_{\text{MAX}} + 5R/2)x(t+1) - 2Ru(t+1), \quad t \in [1, T - 1].$$ (34)

Here, $P_{\text{MAX}} - P_{\text{MIN}} - 5R/2$ has to be non-negative to guarantee that the constraint is valid under all possible unit statuses.

Based on the analysis in IV.A.3, there are 4 unit categories for this three-hour problem. Tightened constraints for the other categories are obtained by repeating the tightening process through three more examples. For all of them, $P_{\text{MIN}} = 59$ MW and $R = 19$ MW/hour, while $P_{\text{MAX}} = 101$ MW, 83 MW, and 67 MW, respectively. The results are summarized in Fig. 9 below.

For Category 4 ($P_{\text{MAX}} \geq P_{\text{MIN}} + 5R/2$), there are 14 new tight constraints beyond Eq. (23 - 28), including Eq. (32) - (34), and Eq. (B1) - (B11) in Section B of Appendix. Among them, Eq. (34) and Eq. (B1-B11) are new tightened constraints, beyond what is in the literature. Category 3 ($P_{\text{MIN}} + 3R/2 \leq P_{\text{MIN}} < P_{\text{MIN}} + 5R/2$), there are 13 tight constraints, excluding Eq. (34) from category 4; Category 2 ($P_{\text{MIN}} + R/2 \leq P_{\text{MIN}} < P_{\text{MIN}} + 3R/2$) there are 7 tight constraints, excluding Eq. (B6)-(B11) from category 3; and Category 1 ($P_{\text{MIN}} < P_{\text{MIN}} + R/2$), there are 2 tight constraints, excluding Eq. (B1)-(B5) from Category 2. All the tightened constraints are still valid for units with minimum up/down times larger than 2. For units with minimum up/down times as 1, tight constraints can be obtained similarly. To demonstrate tightness, 100 Monte Carlo simulation runs are performed. For each unit, a pair of two random variables following $U(0,100)$ are considered: the smaller one is $P_{\text{MIN}}$ and the larger one is $P_{\text{MAX}}$. A third random variable following $U(0,4)$ represents $(P_{\text{MAX}} - P_{\text{MIN}})/R$, and $R$ is calculated correspondingly. The 100 units are categorized into the four types in Fig. 9, with 26, 27, 18, and 29 for each type. With those unit parameters, the corresponding constraints are converted to vertices. Results show that all vertices are binary for all units, demonstrating that the above simplified three-hour single-unit formulation is tight.

b) Generation dependent ramp rates: One-hour

Consider generation dependent ramp rates for the unit in a) with two ramp blocks. For Fig. 1, $P_j$ is 89 MW, $R_j$ and $R_j$ are 19 and 17 MW/hour ($P_j = 70$ MW, $R_j = 103$ MW). Capacity, start up, ramp-up capability, and Eq. (23) are denoted as C1. By constraint-to-vertex conversion, obtain vertices $V_1$. In $V_1$, there are 315 vertices, and 11 are binary. Keep binary vertices as $V_2$ and convert them to the constraint set $C_2$. As compared to C1, there are 7 new constraints as shown in Fig. 10.

$$\begin{align*}
(6) & \quad \pm 59x - 70w_2 - 70w_3 - 70w_4 - 70w_5 + 11u_3 + 11u_5 \leq 0 \\
(7) & \quad -140w_3 + 140w_2 + 140w_3 + 140w_4 + 140w_5 \leq 0 \\
(11) & \quad + w_4 + w_5 + a_4 \leq 0 \\
(12) & \quad - w_4 - w_5 + a_3 \leq 0 \\
(13) & \quad + w_3 + w_4 + w_5 - a_3 - a_5 \leq 0 \\
(14) & \quad - w_3 - w_4 - w_5 + a_4 \leq 0 \\
(15) & \quad - w_3 - w_4 - w_5 + a_4 \leq 0
\end{align*}$$ (35-36)

Figure 9: Tight constraints for four unit categories

Figure 10: New constraints in $C_2$ of a one-hour problem

Constraints (11), and (13) - (15) in Fig. 10 shows relations between $\alpha$ and $\alpha_2$ and (12) shows relations of $x$, $u$ and $\alpha$. Constraints (6) and (7) in Fig. 10 are parameterized below,

$$P_{\text{MIN}} x \leq (\alpha_2 + \alpha_3 + \alpha_4 + \alpha_5)P_j - (P_j - P_{\text{MIN}})(\alpha_2 + \alpha_3),$$

$$(\alpha_2 + \alpha_3 + \alpha_4 + \alpha_5)P_j \leq P_j x - (P_j - P_{\text{MIN}})(\alpha_2 + \alpha_3).$$ (35-36)
For Eq. (35), when \( \alpha_3 = \alpha_3 = 0 \), \( P_{\min} x \leq (\alpha_0 + \alpha_0 + \alpha_0 + \alpha_0) P_3 = p \); and when \( \alpha_3 = 1 \) or \( \alpha_3 = 1 \), \( \alpha_0 + \alpha_0 + \alpha_0 + \alpha_0 \geq 1 \). It can be verified that when \( P_3 \geq P_{\min} + R/2 \), Eq. (36) is valid under all possible combinations of \( x \) and \( u \).

Tightening of generation dependent ramp rates was rarely discussed in the literature, the above tightened constraints are new. They can be extended to other ramp and reserve capability functions, and problems with more blocks.

Example 2. IEEE 118-bus system

This example is based on the IEEE 118-bus system with 54 units [31]. A day-ahead hourly UC problem is considered, and all units have generation-dependent ramp rates with four blocks. The problems are solved by using branch-and-cut with different formulations: (1) standard; (2) adding tightened constraints Eq. (7-10) as \( T_i \); and (3) adding \( T_i \) and tightened constraints Eq. (23-25), (32), (33), (35) and (36) as \( T_i \). In Cplex, optimization stops when computational time reaches the pre-set stop time or the relative mixed-integer programming gap (relative difference between the objectives of the optimal relaxed solution and current integer solution) falls below the pre-set gap. Here the stop time and gap is set as 1800 s and 0.01%, respectively. Results are compared in Table II below. CPU time is the total time on data and model loading, problem solving and solution outputting; and solving time includes root node solving, cutting, and branching. As tightened constraints can be identified based on unit parameters through table lookup quickly in the data-preprocess stage, pre-process time is not included in CPU time.

With the standard formulation, there is no solution after 30 minutes. After adding tightened constraints \( T_1 \), both CPU and solving time is reduced. With \( T_2 \), branching time is further reduced by 75%. To model generation-dependent ramp rates and the corresponding reserve capabilities, many more binary variables and constraints are needed as introduced in Eq. (6), making the problem more complicated as compared with the problem with constant ramp rates. The solving time is much higher than that of the same problem with constant ramp rates, even higher than that of a larger system as to be shown later.

To demonstrate the performance of our tightening approach, another day-ahead hourly UC problem is considered based on the IEEE 118-bus system, where all units have constant ramp rates. Following the testing in [14], a scale factor is randomly generated for each hourly nodal load. Seven instances are created with load scale factors falling in: (a) [0.5, 0.7]; (b) [0.7, 0.9]; (c) [0.9, 1.1]; (d) [1.1, 1.3]; (e) [1.3, 1.5]; (f) [1.5, 1.7]; and (g) [1.7, 1.9]. Time-varying ten-minute spinning reserve requirement is considered, and it is set as 3% of the total load at each hour following [14]. The problems are solved with: (1) the standard UC formulation; and (2) applied cuts and tightened constraints obtained in the two/three-hour problems. The stop time is set as 1800 s, and the stop gap is set as 0.01% following [14]. Cutting and branching time is compared in Fig. 11.

The results show that the cutting time is much reduced after adding cuts and tightened constraints. To compare the results with those in Table 6 of [14], the CPU time and the total number of cuts are shown in Table III below. Since the processor and the version of commercial solver used in the testing could be different as compared with those in [14], the numbers of nodes processed are also provided.

<table>
<thead>
<tr>
<th>Instance (a)</th>
<th>Instance (b)</th>
<th>Instance (c)</th>
<th>Instance (d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
</tr>
<tr>
<td>CPU</td>
<td>3.9</td>
<td>3.68</td>
<td>2.41</td>
</tr>
<tr>
<td>Cuts</td>
<td>751</td>
<td>23</td>
<td>993</td>
</tr>
<tr>
<td>Node</td>
<td>27</td>
<td>96</td>
<td>0</td>
</tr>
</tbody>
</table>

It is shown that the CPU time is reduced by 6% - 33% after adding cuts and tightened constraints, and the total number of cuts is reduced by 92% - 100%. For instances 3 and 4, there are no cuts at all, demonstrating good performance of our approach. As compared with the results in [14], it is shown that the CPU time, and the numbers of cuts and nodes with our tightened formulations are much less.

The above results on units with generation dependent and constant ramp rates demonstrate that our tightening approach is general, and can be used for other problems in power systems.

Example 3. Polish system

This example is based on the Polish 2383-bus system with 327 units [32]. A day-ahead hourly UC problem is considered, and all units have constant ramp rates. To test performance of our tightened formulations for large-scale UC problems, all units are treated dispatchable. Ramp rates of 227 units are reduced by 75% to satisfy \( P_{\max} \geq P_{\min} + 3 R/2 \), and ramp rates of the other 100 units are reduced by 50% to satisfy \( P_{\min} + 3 R/2 \leq P_{\max} < P_{\min} + 5 R/2 \). Minimum up/down times are assumed as 2 or 3 hours. TMSR and TMOR are assumed as 150 MW and 300 MW (as unit ramp rates are reduced), respectively.

Results with different formulations are presented in Table IV below: (1) the standard UC formulation (Std); (2) adding cuts (Cuts); (3) and adding tightened constraints obtained in the two/three-hour problems (Tightened). From (1) to (3), the
process is accumulative. To demonstrate the performance of our tightening approach, the problem with the standard formulation is also solved with aggressive cuts and heuristics provided by Cplex as (4) (Cplex cuts) and (5) (Cplex heuristics). The heuristics are called every 500 nodes. From (4) to (5), the process is accumulative. The standard UC formulation considered here has 2 binary (2-bin) decision variables for unit on/off and start-up, and formulations with 3 binary (3-bin) decision variables for unit on/off, start-up, and shut-down is also discussed in the literature [3, 12, 13, and 19]. For comparison purposes, the above problem is also solved with the 3-bin formulation in [19] (without cuts) as (6) (3-bin).

Comparing (1) to (3), it can be seen that CPU, solving, cutting and branching time is significantly reduced by adding cuts and tightened constraints, while the solution quality is still high. With the standard formulation, a feasible solution with a gap of 0.01% is obtained in 1174 s, while the time on cutting and branching is 665s and 310s. By adding cuts and tightened constraints, a feasible solution which is 0.0007% higher than the above is obtained in 127s, while the cutting and branching time is only 22s and 40s. It demonstrates that tightening single unit formulations also improves the computational efficiency and solution quality when solving the overall UC problems.

Comparing (1), (4) and (5), the results show that CPU time is reduced by adding aggressive cuts and heuristics in Cplex, but still higher than that obtained with our cuts and tightened constraints without adding those aggressive cuts and heuristics. Comparing (1) and (6), the 2-bin formulation has lower CPU time and less numbers of cuts as compared with the 3-bin.

The problem is also solved with different reserve requirements. For the above 6 formulation configurations, cutting and branching time is compared in Fig. 12, and the numbers of cuts are shown in Table VI.

Comparing (1) to (3), it shows that time on branching and cutting is both dramatically reduced by adding cuts and tightened constraints, demonstrating computational efficiency of our approach. In addition, the total numbers of implied bound cuts are reduced by 87% - 89%, implying most of this type of cuts are applied to single units. The total numbers of mixed integer rounding cuts are reduced by 57% - 63%, and the remaining may be related to system-level constraints. The total numbers of other cuts are reduced by 71% - 78%. Results demonstrate great potential of our systematic approach to tighten complicated MBLP problems.

Comparing (1), (4) and (5), the results show that the total cutting and branching time is reduced by 32% - 74% after adding aggressive cuts and heuristics in Cplex, but it is still higher than that obtained with our cuts and tightened constraints and without those aggressive cuts and heuristics. In addition, after adding Cplex aggressive cuts and heuristics to the standard 2-bin formulation, the total numbers of implied bound cuts, integer rounding cuts and other cuts are reduced by -5% ~ 3% (a negative value means that the number of cuts is increased), -5% ~ -2% and 3% ~ 5%, respectively. After adding our cuts and tightened constraints to the standard 2-bin formulation, time on branching and cutting are significantly reduced by 94% - 96%. In addition, the numbers of implied bound cuts, integer rounding cuts and other cuts are dramatically reduced by 88% - 89%, 56% - 64% and 71% - 78%, respectively.

Comparing (1) and (6), it can be seen that the 2-bin performs better in terms of solving time and numbers of cuts. The total cutting and branching time with the standard 2-bin formulation is less than that of the 3-bin by 16% - 45%. In addition, the total numbers of implied bound cuts, integer rounding cuts and other cuts with 2-bin before tightening is also less than those of 3-bin by 56% - 57%, 37% - 39% and 8% - 12%, respectively.

VI. CONCLUSION

In this paper, a systematic approach is developed to tighten single-unit UC formulations in the data pre-processing stage for the first time. Existing cuts are first applied, and then tightened constraints are established based on novel “constraint-and-vertex conversion,” “vertex elimination” and “parameterization” processes. By analyzing problems with short-time horizons, e.g., two or three hours, tightened
formulations for single units with constant and generation-dependent ramp rates are obtained, beyond what is in the literature. For each category, its tightened constraints are developed, and a look-up table is established. Then for each unit, tightened constraints can be identified through table lookup based on unit parameters in the data pre-processing stage. Numerical testing results on IEEE 118-bus and Polish 2383-bus systems demonstrate computational efficiency and solution quality benefits of formulation tightening. The approach is general and has great potential for tightening complicated MBLP problems. We believe that tightened single-unit formulations can be fully exploited by using our latest powerful decomposition and coordination approach [33]. This will fundamentally change how we formulate and solve complicated MBLP problems in power systems and beyond.

According to our knowledge, it is hard to extend the above approach to tighten the entire unit commitment problem with a large number of units considering system-level constraints, e.g., system demand, reserve requirement, and transmission capacity constraints. To tighten those system-level coupling constraints, we have developed another systematic approach [34].

VII. APPENDIX

A. Applied cuts

\[ \begin{align*}
  p(t-1) &\leq (P_{\text{min}} + R/2)(x(t-1) - (P_{\text{min}} - R/2)u(t-1) \\
  + (P_{\text{max}} - P_{\text{min}} - R/2)(x(t) - u(t)), \forall t. \\
  p(t-1) \leq & \quad - (P_{\text{max}} - P_{\text{min}} - R/2)(x(t-1) + R/2u(t-1) \\
  + (P_{\text{max}} - R/2)(x(t)+R/2u(t)), \forall t. \\
  p(t+1) - p(t-1) \leq & \quad - (P_{\text{max}} - P_{\text{min}} + R) x(t+1) \\
  - R/2u(t+1), t \in [1, T]. \\
  - (P_{\text{max}} - P_{\text{min}} + R/2)(x(t)+R/2u(t)), \forall t. \\
  p(t+1) - p(t) \leq & \quad - (P_{\text{max}} - P_{\text{min}} - R/2)(x(t) - u(t)) \\
  + (P_{\text{max}} - P_{\text{min}} - R/2)(x(t-1) - u(t-1)), t \in [1, T]. \\
  p(t) \leq & \quad - (P_{\text{max}} - P_{\text{min}} + R/2)(x(t)-u(t)) \\
  - (P_{\text{max}} - P_{\text{min}} - R/2)(x(t)) \\
  + R(x(t)+u(t)) - u(t+1), t \in [1, T]. \\
  p(t-1) - p(t+1) \leq & \quad (P_{\text{max}} + R/2)x(t-1) - 3R/2u(t-1) \\
  + 3R/2(x(t)-u(t)) - P_{\text{max}}x(t+1), t \in [1, T].
\end{align*} \]  

(B9)  

\[ \begin{align*}
  p(t-1) - p(t+1) \leq & \quad (P_{\text{max}} + R/2)x(t-1) - Ru(t-1) + Rx(t) \\
  - 3R/2u(t) - (P_{\text{max}} - R/2)x(t+1) \\
  - R/2u(t+1), t \in [1, T]. \\
  - (P_{\text{max}} - P_{\text{min}} - R/2)(x(t-1) - R/2u(t) \\
  + (P_{\text{max}} + R)x(t+1) - 3R/2u(t+1)), t \in [1, T].
\end{align*} \]  

(B10)  

(B11)

VIII. REFERENCES


Bing Yan (S’11–M’17) received the B.S. degree from Renmin University of China in 2010, M.S. and Ph.D. degrees from University of Connecticut in 2012 and 2016, respectively. She is currently an assistant professor in the Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology. Her research interests include power system optimization, grid integration of renewables, energy-based operation optimization of distributed energy systems, and scheduling of manufacturing systems.

Peter B. Luh (S’77–M’80–SM’91–F’95) received the B.S. degree from National Taiwan University, the M.S. degree from M.I.T., and the Ph.D. degree from Harvard University. He has been with the University of Connecticut since 1980, and is the Board of Trustees Distinguished Professor and SNET Professor of Communications & Information Technologies. His research interests include smart power systems—smart grid, design of auction methods for electricity markets, effective renewable (wind and solar) integration to the grid, electricity load and price forecasting with demand response, and micro grid. He was the Vice-President of Publication Activities for the IEEE Robotics and Automation Society.

Tongxin Zheng (SM’08) received the B.S. degree in electrical engineering from North China Institute of Electric Power, Baoding, China, in 1993, the M.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 1996, and the Ph.D. degree in electrical engineering from Clemson University, Clemson, SC, USA, in 1999. He is currently a Technical Director at ISO New England. His main research interests include power system operation and electricity market design.

Feng Zhao (M’08) received the B.S. degree in automatic control from Shanghai JiaoTong University, Shanghai, China, in 1998, the M.S. degree in control theory and control engineering from Tsinghua University, Beijing, China, in 2001, and the Ph.D. degree in electrical engineering from the University of Connecticut, Storrs, CT, USA, in 2008. He is currently a Lead Analyst at ISO New England, Holyoke, MA, USA. His research interests include mathematical optimization, power system planning and operations, and economics of electricity markets.

Mikhail A. Bragin (S’11–M’17) received his B.S. and M.S. degrees in mathematics from the Voronezh State University, Russia, in 2004, the M.S. degree in physics and astronomy from the University of Nebraska-Lincoln, USA, in 2006, and the M.S. and Ph.D. degree in electrical and computer engineering from the University of Connecticut, USA, in 2014 and 2016, respectively. He is an assistant research professor in electrical and computer engineering at the University of Connecticut. His research interests include mathematical optimization, including power system optimization, grid integration of renewables (wind and solar), energy-based operation optimization of distributed energy systems, and scheduling of manufacturing systems.

Jinye Zhao (M’11) received the B.S. degree from East China Normal University, Shanghai, China, in 2002, the M.S. degree in mathematics from National University of Singapore, Singapore, in 2004, and the M.S. degree in operations research and statistics and the Ph.D. degree in mathematics from Rensselaer Polytechnic Institute, Troy, NY, USA, in 2007. She is currently a Lead Analyst at ISO New England, Holyoke, MA. Her main interests include game theory, mathematical programming, and electricity market modeling.

Dane A. Schiro (M’14) received the B.S. degree in environmental engineering from the Johns Hopkins University, Baltimore, MD, USA, and the Ph.D. degree in industrial engineering from the University of Illinois, Urbana, IL, USA. He is currently a Senior Analyst at ISO New England, Holyoke, MA, USA. His research interests include the intersection of optimization theory, economics, and operations.

Izudin Lelic (M’98) received the B.S. and M.S. degrees in electrical engineering from University of Tuzla, Bosnia-Herzegovina and University of Belgrade, Serbia. He is currently a Principal Analyst at ISO New England, Holyoke, MA, USA. His research interests include electricity markets and power system analysis and simulations.