Ward Reduction in Unit-Commitment Problems

Bruno Colonetti, Erlon Finardi
Electrical and Electronic Engineering Department, Federal University of Santa Catarina, Florianópolis, Brazil
{colonetti.bruno@posgrad.ufsc.br, erlon.finardi@ufsc.br}

***Abstract*—One of the most challenging aspects of unit commitment (UC) is dealing with the grid, which is commonly represented by thousands of nodes and branches, leading to large optimization models. Because the computational difficulty generally increases with the model size, managing the network representation size can lead to significant savings in running times. Thus, we explore the widely used Ward reduction to reduce the network's size through a process that iteratively removes nodes from the network. We present how, under mild conditions, the resulting reduced network model is equivalent to the original one. We evaluate our approach on 20 UC instances, with up to 13,659 nodes and 18,625 branches. We are able to produce reduced models with as few as 6.8% of the original nodes. Moreover, we show that the network reduction provides average speed-ups from 10% to 36%.**

***Index Terms*—Network reduction; Unit commitment; Ward.**

I. Introduction

Unit commitment (UC) is an optimization model used by Independent System Operators to schedule the generation assets under their supervision to meet the demand while satisfying system-wide constraints. Among such constraints, the most notable, and perhaps most critical, are the transmission constraints. In UC, the network is generally represented through a DC model, in which reactive power and transmission losses are neglected, branch's endpoint nodes' voltage angles difference is assumed close to zero, and node voltage magnitudes are assumed to be unity across all grid [13]. The constraints that define the feasible operating points under this representation dictate that power balances must be satisfied at all network nodes, and branches' capacities must not be violated. Intuitively, as the number of nodes and branches in the network representation increases, so does the difficulty in solving the resulting UC, as more variables and constraints are included in the optimization model. Acknowledging the importance of network representation, researchers have proposed numerous methods to reduce its burden on the UC.

There are two works that are closest to ours: [8] and [10]. In [8], the authors show how nodes connected to the grid through a single branch can be removed from the network without affecting the network representation. The authors show that this unassuming move can lead to significant computational savings in a transmission-constrained UC. On the other hand, [10] is a somewhat more aggressive approach, in which the authors are able to severely reduce the size of networks in the context of transmission-expansion planning problems. As in our work, they also base their node-elimination procedure on the Ward reduction [15] but, different from ours, they rely on (1) a pre-process step to identify candidate nodes to be deleted, and (2) assigning limits to artificial lines added through the Ward reduction based on [6]. The pre-process step (1) consists of solving a so-called multicut problem to distribute nodes among clusters. Originally, the multicut problem is a mixed-integer linear programming problem, but the integrality is dropped and a rounding phase is applied after solving the continuous relaxation in order to obtain integer solutions. In contrast, our strategy does not require solving any additional problem to identify candidate nodes to be deleted. In our strategy, the candidate nodes are by-products of the identification of redundant branch flow bounds. In addition, in [10], authors resort to the procedures of [6] to compute line flow limits to the artificial lines added through the Ward reduction. Again, in our strategy, computing these limits is either entirely unnecessary, or they are obvious by-products of the Ward reduction.

Furthermore, the most common approach to reducing the computational burden of the network in UC is to identify branches' flow limits that are either redundant or are not possibly binding (i.e., they are inactive) in economically meaningful schedules of the generation assets. Once identified, the constraints representing these limits in the optimization model can be removed. As usually only a small fraction of transmission constraints, if any, are binding, identifying the inactive constraints results in significant savings. This is the general reasoning behind many works on reducing network computational burden [9], [11], [5], [1], [16], [18]. However, none of these works leverages the identification of redundant and inactive transmission constraints to apply Ward reduction and further reduce the computational burden of the network representation in UC models.

Another popular strategy to reduce the network burden is node aggregation or clustering. Nonetheless, these strategies are more commonly found in either power flow studies, or longer term studies. For instance, in [2], a node clustering strategy based on electrical distance. Similarly, in [14], [3], [7], node clustering is based on the sensitivities of line flows to nodes' injections. The authors of [12], on the other hand, base their node clustering strategy on node prices. A common feature of node aggregation and clustering is that, in general, the reduced network is not necessarily equivalent to the original, full network. Since we are interested in UC, network equivalence is paramount to obtain from the reduced network schedules and dispatches that are feasible in the full network. Hence, node aggregation and clustering are not in general considered suitable strategies in this context.

In this work, we expand the network reduction proposed

Nomenclature

Sets and indices

$\mathcal{G}_b$ units connected to node b

$\mathcal{L}^-$, $\mathcal{L}^+$ Branches whose from and to node is b, respectively

$g \in \mathcal{G}$ Generating units

$l \in \mathcal{L}$, $b \in \mathcal{B}$ Branches, and network nodes, respectively

$t \in \mathcal{T}$ Periods

Parameters and constants

$\mathbf{B}$ Susceptance matrix

$\mathbf{C}^s$ Unitary load shedding cost in \$/p.u.

$\mathbf{C}_g$, $\overline{\mathbf{C}}_g$ Unitary generation cost in \$/p.u., and start-up cost in \$, respectively

$\mathbf{D}_{b,t}$ Net node load in p.u.

$\mathbf{PTDF}_{l,b}$ Sensitivity of branch flow l w.r.t. positive power injections at node b, nondimensional

$\mathbf{R}_g$ Maximum ramp in p.u./hour, assumed equal to increases and decreases in generation.

$\mathbf{T}_g^{\text{up}}$, $\mathbf{T}_g^{\text{down}}$ Minimum up and down times of unit g in hours

$\mathbf{Y}_l$ Admittance in p.u./rad

$\overline{\mathbf{F}}_l$, $\underline{\mathbf{F}}_l$ Maximum branch flow in p.u. in the positive and negative directions, respectively

$\overline{\mathbf{P}}_g$, $\underline{\mathbf{P}}_g$ Maximum and minimum generation in p.u.

from(l), to(l) Arbitrarily defined from and to nodes of branch l, respectively. Flows in branch l from node from(l) to node to(l) are defined to be positive

Variables

f, θ Branch flow in p.u., and node voltage angle in rad, respectively

p, p^{disp} Total generation and generation above the unit's minimum, respectively, both in p.u.

s Load shedding in p.u.

y, x, d Start-up, shut-down and status binary variables

in [8] and combine it with techniques to identify redundant flow limits to produce a reduced network model. Our objective is to generate equivalent network models with fewer nodes and fewer branches in hope that fewer elements will cause a reduction in computational burden. Our work is a significant extension of [8], where we remove from the original network not only end-of-line nodes but also nodes with more than one branch connection. However, to safely remove those elements without altering the feasible region of the optimization model, i.e., keeping an equivalent network model, we rely on identifying branches whose limits cannot be reached by [1]. This is yet another crucial difference between our work and [8], where such identification is not necessary because only nodes connected to a single branch are removed.

This paper is organized as follows: we introduce the unit-commitment model in Sec. II; we discuss the Ward reduction and introduce our proposal in Sec. III; in Sec. IV, we present our test cases and results. Finally, our concluding remarks are given in Sec. V.

II. Unit commitment

We adopt the position of an ISO who must minimize operation costs while meeting the demand and satisfying the generators' and network's constraints. In the following, we detail each component of the unit-commitment model.

A. Generation model

The statuses of generating units are defined by the logical constraints (1), minimum up (2) and down (3) times.

$$y_{g,t} - x_{g,t} - d_{g,t} + d_{g,t-1} = 0 \quad \forall g \in \mathcal{G}, \forall t \in \mathcal{T}. \tag{1}$$

$$\sum_{i=t-\mathbf{T}_g^{\text{up}}} y_{g,i} \leq d_{g,t} \quad \forall g \in \mathcal{G}, \forall t \in \mathcal{T}. \tag{2}$$

$$\sum_{i=t-\mathbf{T}_g^{\text{down}}} x_{g,i} \leq (1 - d_{g,t}) \quad \forall g \in \mathcal{G}, \forall t \in \mathcal{T}. \tag{3}$$

To ensure that an unit can only inject power into the network if its status is 1, we use constraints (4) and (5). While the first limits the maximum power of the unit above its minimum, the second set of constraints defines that, when operating ($d_{g,t} = 1$), the total power injected into the grid, p, will be at least equal to the unit's minimum power.

$$0 \leq p_{g,t}^{\text{disp}} \leq \left(\overline{\mathbf{P}}_g - \underline{\mathbf{P}}_g\right) \cdot d_{g,t} \quad \forall g \in \mathcal{G}, \forall t \in \mathcal{T}. \tag{4}$$

$$p_{g,t} - \underline{\mathbf{P}}_g \cdot d_{g,t} - p_{g,t}^{\text{disp}} = 0 \quad \forall g \in \mathcal{G}, \forall t \in \mathcal{T}. \tag{5}$$

When either turned on or off, the generation of an unit in the subsequent and prior period, respectively, is set to be equal to its minimum generation. Following the two-variable formulation used here, this requirement translates to setting variable $p_{g,t}^{\text{disp}}$ to zero whenever $y_{g,t} = 1$ and $x_{g,t+1} = 1$. Additionally, when operating, the changes in generation between consecutive periods, both increases and decreases, are constrained by ramp limits. We represent both requirements, start-up and shut-down limits, and ramp up and down limits, through constraints (6).

$$-\mathbf{R}_g \cdot d_{g,t} - \leq p_{g,t}^{\text{disp}} - p_{g,t-1}^{\text{disp}} \leq \mathbf{R}_g \cdot d_{g,t-1} \quad \forall g \in \mathcal{G}, \forall t \in \mathcal{T}. \tag{6}$$

B. Network model

As in this paper we are primarily concerned with the network model size, we implement two of the most widely used DC models in the literature: (1) the B-theta model, and (2) the power transfer distribution factor (PTDF) model [13]. In the B-theta model, the nodes‘ voltage angles are explicitly included in the model. On the other hand, in the PTDF model, the flows in the branches are directly represented as functions of the power injections at the network's nodes through a $|\mathcal{L}|$ by $|\mathcal{B}|$ sensitivity matrix, without explicitly representing the voltage angles. It is important to emphasize that, apart from numerical tolerances, the B-theta and PTDF are equivalent.

Regardless of the model, branch flows are subjected to their lower ($\underline{\mathbf{F}}_{l,t}$) and upper bounds ($\overline{\mathbf{F}}_{l,t}$) (7).

$$\underline{\mathbf{F}}_{l,t} \leq f_{l,t} \leq \overline{\mathbf{F}}_{l,t} \quad \forall l \in \mathcal{L}, \forall t \in \mathcal{T}. \tag{7}$$

1) B-theta model

In the B-theta model, flows are described as functions of the angular difference between its endpoints, as shown in (8).

$$f_{l,t} - \mathbf{Y}_l \cdot \left(\theta_{\text{from}(l),t} - \theta_{\text{to}(l),t}\right) = 0 \quad \forall l \in \mathcal{L}, \forall t \in \mathcal{T}. \tag{8}$$

In this model, we enforce the power balance at each node through constraints (9). Note that we include slack variables s to account for small load shedding — they are heavily penalized in the objective function.

$$\sum_{g\in\mathcal{G}_b} p_{g,t} - \sum_{l\in\mathcal{L}^-} f_{l,t} + \sum_{l\in\mathcal{L}^+} f_{l,t} + s_{b,t} = \mathbf{D}_{b,t} \quad \forall b\in\mathcal{B}, \forall t\in\mathcal{T}. \tag{9}$$

2) PTDF model

Different from the B-theta model, in the PTDF model, it is necessary to include constraints that guarantee that the system's total generation and demand match for each time, as in (10).

$$\sum_{g\in\mathcal{G}} p_{g,t} + \sum_{b\in\mathcal{B}} s_{b,t} = \sum_{b\in\mathcal{B}} \mathbf{D}_{b,t} \quad \forall t\in\mathcal{T}. \tag{10}$$

In this model, the flows in the branches are then expressed as linear functions of the nodes‘ net injections. For an arbitrary node b and time t, its net injection is $\sum_{g\in\mathcal{G}_b} p_{g,t} + s_{b,t} - \mathbf{D}_{b,t}$. With the PTDF model, the flows are thus given as (11).

$$f_{l,t} = \sum_{b\in\mathcal{B}} \mathbf{PTDF}_{l,b} \cdot \left(\sum_{g\in\mathcal{G}_b} p_{g,t} + s_{b,t} - \mathbf{D}_{b,t} \right) \quad \forall l\in\mathcal{L}, \forall t\in\mathcal{T} \tag{11}$$

C. Unit-commitment model

The elements described above are the building blocks of our unit-commitment model stated in (12).

$$\begin{aligned} \min \quad & \sum_{t\in\mathcal{T}} \left(\sum_{g\in\mathcal{G}} \mathbf{C}_g \cdot p_{g,t} + \sum_{g\in\mathcal{G}} \overline{\mathbf{C}}_g \cdot y_{g,t} + \mathbf{C}^s \cdot \sum_{b\in\mathcal{B}} s_{b,t} \right) \\ \text{s.t.} \quad & (1), (2), (3), (4), (5), (6), \\ & (7), \textbf{ and either } (8), (9) \textbf{ or } (11), (10), \\ & (d_{g,t}, y_{g,t}, x_{g,t}) \in \{0,1\}^3 \quad \forall g\in\mathcal{G}, \forall t\in\mathcal{T}, \\ & s_{b,t} \geq 0 \quad \forall b\in\mathcal{B}, \forall t\in\mathcal{T}. \end{aligned} \tag{12}$$

In this model, the unitary cost of generation is $\mathbf{C}_g$, for an arbitrary generator g, and $\overline{\mathbf{C}}_g$ is the start-up cost of g. Additionally, load shedding is penalized with $\mathbf{C}^s$. All power units in this model are p.u..

The model (12) is mixed-integer linear, and its solution depends heavily on the ability of the solver to solve its linear relaxation (the model resulting from dropping the integrality requirements), both in the root node and also later in the nodes of the branch-and-cut tree, where the relaxation is usually solved with a simplex method.

III. Network reduction

In most works related to network reduction, three sets of nodes are defined: internal nodes, boundary nodes, and external nodes. Internal nodes are those to be retained that have no direct connection to external nodes, i.e., there is no branch directly connecting any of them to any of the external nodes. Boundary nodes are also retained but, different from the first set, they do have direct connections to the external nodes. The last set comprises the group of nodes to be removed from the network model. In a Ward reduction, the impact of the external nodes on the retained network is accounted for by distributing the external nodes' power injections among the boundary nodes in addition to adding new branches among them. We briefly reproduce below the steps of the Ward reduction [15].

Firstly, the nodes' power injection are ordered, for convenience, according to the set that the node belongs to (internal, boundary, or external). Then, the net power injections are written in matrix form as functions of the nodes' voltage angles as in (13).

$$\begin{bmatrix} p_{\mathrm{i}} \\ p_{\mathrm{b}} \\ p_{\mathrm{e}} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_i^i & \mathbf{B}_i^b & \mathbf{B}_i^e \\ \mathbf{B}_b^i & \mathbf{B}_b^b & \mathbf{B}_b^e \\ \mathbf{B}_e^i & \mathbf{B}_e^b & \mathbf{B}_e^e \end{bmatrix} \cdot \begin{bmatrix} \theta_{\mathrm{i}} \\ \theta_{\mathrm{b}} \\ \theta_{\mathrm{e}} \end{bmatrix}, \tag{13}$$

where matrices $\mathbf{B}$ are partitions of the nodal susceptance matrix. For instance, matrix $\mathbf{B}_i^b$ is the sensitivity of the power injections at internal nodes w.r.t. to the voltage angles at the boundary nodes. By construction, $\mathbf{B}_i^e = \mathbf{0}$ and $\mathbf{B}_e^i = \mathbf{0}$, and $\mathbf{B}_y^x = \mathbf{B}_x^{y\top}$, for $(x,y) \in \{e,i,b\}^2$. By applying successive row operations to (13), we can eliminate θ_e. We show in (14) the boundary nodes' power injections after eliminating θ_e.

$$p_{\mathrm{b}} = \mathbf{B}_b^i \cdot \theta_{\mathrm{i}} + \left(\mathbf{B}_b^b - \mathbf{B}_b^e \cdot (\mathbf{B}_e^e \cdot)^{-1} \cdot \mathbf{B}_e^b \right) \cdot \theta_{\mathrm{b}} + \mathbf{B}_b^e (\mathbf{B}_e^e)^{-1} \cdot p_{\mathrm{e}} \tag{14}$$

In (14), the power injections at the boundary nodes have been changed by two terms: (1) new branches among them ($-\mathbf{B}_b^e \cdot (\mathbf{B}_e^e \cdot)^{-1} \cdot \mathbf{B}_e^b$), and (2) reassignment of the power injections at external nodes to boundary nodes $\mathbf{B}_b^e (\mathbf{B}_e^e)^{-1} \cdot p_{\mathrm{e}}$.

We discuss in the following the pros and cons of the Ward reduction in the context of UC.

A. Potential benefits of Ward reduction

For the B-theta model, if properly applied, the benefits of Ward reduction are evident: (1) decrease in the number of nodes; and (2) decrease in the number of branches. These decreases are naturally dependent upon applying the Ward reduction only when its beneficial in terms of model size, as we discuss shortly. Nonetheless, the decrease in nodes and branches reflects in a decrease in the number of variables and constraints in the optimization model. On the other hand, applying the Ward reduction when the PTDF model is used may not immediately result in visible benefits. That is because, for the retained lines, the expressions of the branch flows do not change. However, noting that the Ward reduction enables the elimination of nodes, the associated slack variables of the eliminated nodes can also be removed from the model since no more than one slack per node is needed. Thus, if the slacks are also eliminated, the number of nonzero terms in the PTDF flow expressions can also be reduced.

B. Potential shortcomings of Ward reduction

In a UC model, applying the Ward reduction must take into account certain of its shortcomings. These shortcomings are exclusive to the B-theta DC model since in the PTDF model, as flows are explicitly written as functions of the node injections, removing nodes through an exact method and reallocating injections has no impact in the flow expression.

Firstly, although all external nodes are eliminated from the model, depending on the number of boundary nodes, an overwhelmingly large number of new branches can be added to the model. In fact, the possible number of new branches is $\frac{N\cdot(N-1)}{2}$, where N is the number of boundary nodes. In addition to the number of new branches, their associated susceptances might be very low (i.e., branches with high reactance), which can lead to numerical problems. Thus, given the large number of new branches and their possibly high reactances, after applying Ward reduction, one might be faced with the problem of selecting (discarding) the branches to be added (removed) to (from) the model. Naturally, discarding some of these new branches invariably leads to a network representation that is no longer an equivalent.

Secondly, the new branches 'created' after removing the external nodes have no obvious bounds. If any of the branches connected to one of the external nodes has its capacity violated, it is not clear how this information can be conveyed to the reduced network.

Finally, as the injections formerly connected to external nodes are distributed among the boundary nodes, their coefficients in their new nodes, defined by $\mathbf{B}^e_b(\mathbf{B}^e_e)^{-1}$, can be rather small, again introducing numerical problems to the model.

C. Proposed strategy

In face of the pros and cons of the Ward reduction discussed above, in the following, we summarize the main steps of our network-reduction strategy, where we try to maximize the benefits of the network reduction while not unintendedly increasing the computational burden. Evidently, the steps can be repeated for as long as there are nodes to be removed.

1) Identify redundant branches' limits;
2) Remove degree-1 and degree-2 nodes with no injections;
3) Remove degree-1 nodes with injections;
4) Remove degree-2 load nodes with at most one possibly active branch;
5) Remove degree-n nodes with no active branches.

We take the definition of degree-1 nodes from [8]. Degree-1 nodes are those connected to the rest of the network through a single branch. Naturally, degree-2 nodes are those connected by two branches, and, degree-n nodes by n branches.

1) Identify redundant branches' limits

This steps consists in determining which branches' lower and upper bounds cannot be possibly reached by a feasible schedule of the generating units. For branches whose lower and upper bounds cannot be reached in any of the periods of the UC, the branch's limits are deemed redundant. Otherwise, if in at least one of the periods one of the limits cannot be unequivocally classified as redundant, then the branches' limits are deemed 'possibly active'. Note that even though a branch might be deemed possibly active, its bounds are only enforced in periods when they are not redundant. Moreover, this is a flexible step; any algorithm able to identify redundant and inactive flow bounds can be applied. In our implementation, we mostly follow [1] and we only identify redundant flow bounds.

2) Remove degree-1 and degree-2 nodes with no injections

Although no-injection degree-1 nodes make no contribution in a DC model, some of them might still be presented in the network data. More importantly, no-injection degree-2 nodes, i.e., nodes that have no demand and no generation and are connected to exactly two branches can be promptly removed from the network by combining the series reactance of its branches and properly computing the lower and upper bounds of the resulting branch's flow.

3) Remove degree-1 nodes with injections

Akin to [8].

4) Remove degree-2 load nodes with at most one possibly active branch

By defining the degree-2 load node as the external node and its two adjacent nodes as boundary nodes, we can apply Ward reduction to remove it, reallocate any injection to the boundary nodes and obtain the reactance of the new branch. To account for the possibly active bound, we express it as a function of the old bound and the injection of the removed node. Further details are given in sub Sec. III-D.

5) Remove degree-n nodes with no active branches

Again, we define the degree-n node as the external node and all its n adjacent nodes as boundary nodes. In this step, it is crucial to be sure that all branches connected to the target node are redundant. Naturally, if the degree is too high, it might not be beneficial to remove the node, as too many new branches would be added to the model. Thus, we limit the screening of candidate nodes to be removed to at most degree-20 nodes and only remove a node if the net number of new branches added is at most 1. For instance, suppose we have a candidate node to be removed with degree 10. Removing it would add 45 new branches to the model. Normally, however, some of these branches might be paralleled to already existing branches. If the number of new branches paralleled to existing branches is 44, the net result would be that only one new branch is in fact added. In this case, we remove the node.

All steps in the proposed strategy required only one change from the well-known model stated in (12): generators are no longer connected to a single node. Instead, their power outputs are now split according to the coefficients resulting from the Ward reduction applied to the node they were formerly connected to. Thus, (9) and (8) are rewritten, respectively, as (15) and (16) for any $t \in \mathcal{T}$.

$$\sum_{g\in\mathcal{G}_b} \mathbf{K}_{g,b}\cdot p_{g,t} - \sum_{l\in\mathcal{L}^-} f_{l,t} + \sum_{l\in\mathcal{L}^+} f_{l,t} + s_{b,t} = \mathbf{D}^*_{b,t} \;\; \forall b\in\mathcal{B}, \tag{15}$$

$$f_{l,t} = \sum_{b\in\mathcal{B}^r} \mathbf{PTDF}_{l,b}\cdot \left(\sum_{g\in\mathcal{G}_b} \mathbf{K}_{g,b}\cdot p_{g,t} + s_{b,t} - \mathbf{D}^*_{b,t} \right) \;\; \forall l\in\mathcal{L}^r. \tag{16}$$

In (15) and (16), coefficients $\mathbf{K}_{g,b}$ are given by the Ward reduction, as described in (14). In these equations, the demands, now represented by $\mathbf{D}^*$, are the aggregations of demands at each node retained in the network after nodes have been eliminated and their respective demands redistributed among the retained nodes. Different from the generators, the slacks s

are not redistributed among boundary nodes once its former node is eliminated because at most one of them is necessary for each node. Naturally, we have that new set of nodes, $\mathcal{B}^r$, with $\mathcal{B}^r \subset \mathcal{B}$. The same cannot be said about the new set of branches, $\mathcal{L}^r$, which may or may not be a subset of $\mathcal{L}$.

In the following, we show how degree-2 nodes are removed from the grid, even if one of its branches is possibly active, and we illustrate our approach with a 14-node system.

D. Removing degree-2 nodes with one possibly active branch

In Fig. 1, we present a degree-2 node, node 2, with one possibly active branch, branch $f_{2,3}$. In this example, the power injections p account for generation and loads at the respective nodes as well as the net power flows from/to nodes 1, 2 and 3, except through branches $f_{1,2}$ and $f_{2,3}$. Following the usual parlance of network reduction, in this example, 1 and 3 are boundary nodes and 2 is the external one. After removing node 2, a new connection between 1 and 3, $f_{1,3}$, is added. Note that, because bounds on the flow in $f_{2,3}$ need to be enforced, the flows in $f_{1,3}$ must also be constrained to reflect those bounds. Before removing 2, the power balances for these three nodes are defined as follows.

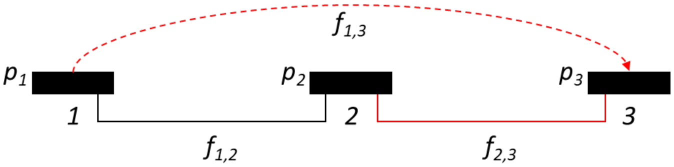


Fig. 1: Two-degree node with one possibly active branch.

$$\begin{bmatrix} p_1 \\ p_3 \\ p_2 \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_{1,2} & 0 & -\mathbf{Y}_{1,2} \\ 0 & \mathbf{Y}_{2,3} & -\mathbf{Y}_{2,3} \\ -\mathbf{Y}_{1,2} & -\mathbf{Y}_{2,3} & \mathbf{Y}_{1,2} + \mathbf{Y}_{2,3} \end{bmatrix} \cdot \begin{bmatrix} \theta_1 \\ \theta_3 \\ \theta_2 \end{bmatrix},$$

where $\mathbf{Y}_y^x$ is the susceptance between nodes x and y, and θ_x is the voltage angle at node x, with $(x, y) \in \{1, 2, 3\}^2$. After removing node 2, we have

$$\begin{bmatrix} p_1 + \left(\frac{\mathbf{Y}_{1,2}}{\mathbf{Y}_{1,2}+\mathbf{Y}_{2,3}}\right) \cdot p_2 \\ p_3 + \left(\frac{\mathbf{Y}_{2,3}}{\mathbf{Y}_{1,2}+\mathbf{Y}_{2,3}}\right) \cdot p_2 \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{Y}_{1,2}\cdot\mathbf{Y}_{2,3}}{\mathbf{Y}_{1,2}+\mathbf{Y}_{2,3}} & -\frac{\mathbf{Y}_{1,2}\cdot\mathbf{Y}_{2,3}}{\mathbf{Y}_{1,2}+\mathbf{Y}_{2,3}} \\ -\frac{\mathbf{Y}_{1,2}\cdot\mathbf{Y}_{2,3}}{\mathbf{Y}_{1,2}+\mathbf{Y}_{2,3}} & \frac{\mathbf{Y}_{1,2}\cdot\mathbf{Y}_{2,3}}{\mathbf{Y}_{1,2}+\mathbf{Y}_{2,3}} \end{bmatrix} \cdot \begin{bmatrix} \theta_1 \\ \theta_3 \end{bmatrix}$$

Thus, the flow in the newly added branch $(1, 3)$ is given by:

$$f_{1,3} = \frac{\mathbf{Y}_{1,2} \cdot \mathbf{Y}_{2,3}}{\mathbf{Y}_{1,2} + \mathbf{Y}_{2,3}} \cdot (\theta_1 - \theta_3).$$

Then, to maintain the balances at nodes 1 and 3 after removing node 2, the flow in $f_{1,3}$ plus the reassigned injections of p_2 to 1 and 3 must be equal to $f_{1,2}$ and $f_{2,3}$:

$$f_{1,2} = f_{1,3} - \frac{\mathbf{Y}_{1,2}}{\mathbf{Y}_{1,2} + \mathbf{Y}_{2,3}} \cdot p_2, \text{ and } f_{2,3} = f_{1,3} + \frac{\mathbf{Y}_{1,3}}{\mathbf{Y}_{1,2} + \mathbf{Y}_{2,3}} \cdot p_2.$$

As only the flow through branch $(2, 3)$ is possibly active, we can use the equations above to enforce the branch's $(2, 3)$ bounds on variables $f_{1,3}$ and p_2, as in the following.

$$\underline{\mathbf{F}}_{2,3} \leq f_{1,3} + \frac{\mathbf{Y}_{1,3}}{\mathbf{Y}_{1,2} + \mathbf{Y}_{2,3}} \cdot p_2 \leq \overline{\mathbf{F}}_{2,3}.$$

Note that, if the injection at node 2 is fixed, then the bounds above can be enforced only on variable $f_{1,3}$ by changing $\underline{\mathbf{F}}_{2,3}$ and $\overline{\mathbf{F}}_{2,3}$, as follows.

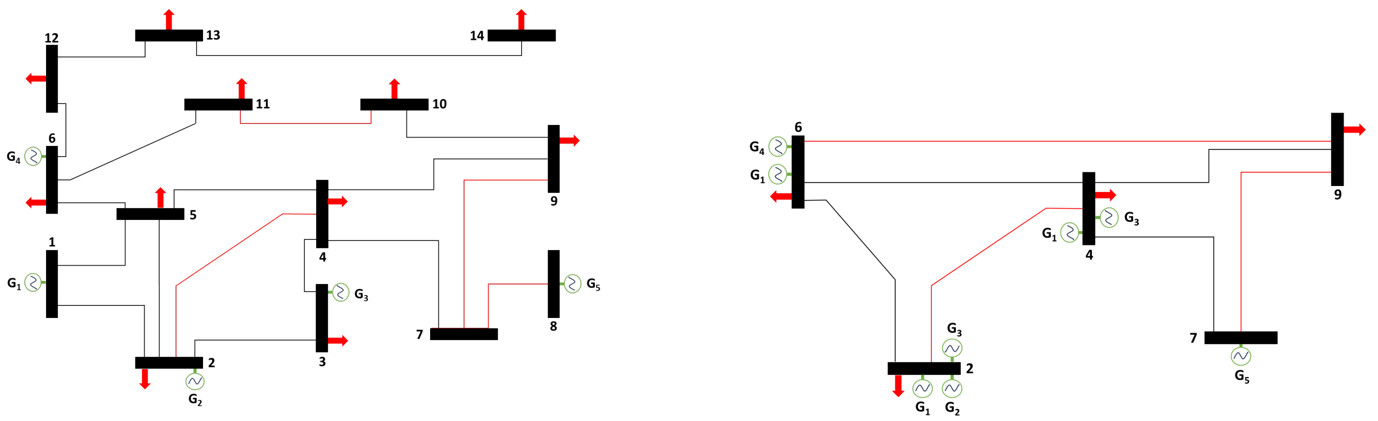
$$\underline{\mathbf{F}}_{2,3} - \frac{\mathbf{Y}_{1,3}}{\mathbf{Y}_{1,2} + \mathbf{Y}_{2,3}} \cdot \mathbf{P}_2 \leq f_{1,3} \leq \overline{\mathbf{F}}_{2,3} - \frac{\mathbf{Y}_{1,3}}{\mathbf{Y}_{1,2} + \mathbf{Y}_{2,3}} \cdot \mathbf{P}_2.$$

In our implementations, we only eliminate degree-2 nodes when at most one of its branches is possibly active and there is no generation connected to it. By doing so, we keep the flow expressions and its bounds in the familiar form shown above, where the bounds are constant although now time dependent. However, note that if a slack variable were associated with a fixed demand, as in (9), neglecting it in the new expression of the flow bounds above results in a relaxation of the upper bound and a tightening of the lower bound (for non-negative susceptances). Thus, if there were load shedding at the node being removed, not accounting for it in the bounds of the new branch can result in an error. Nonetheless, we argue that the slack variables are usually added to compensate for rather small power imbalances and, for any realistic UC, the total value of load shedding should be close to zero.

We further illustrate our reduction strategy with the 14-node system shown in Fig. 2a. In this example, we show how the successive removal of degree-1, degree-2 and degree-3 nodes lead to a significantly smaller network. In Fig. 2a, nodes 14, 13, 12, and 8 are removed as degree-1 nodes. Then, nodes 10, 11, 1, and 3 are removed as degree-2 nodes. Finally, node 5, which is a degree-3 node, is also removed. Furthermore, parallel branches are combined into single branches. Different from the full network representation, generators 1, 3 and 4 have their injections now split among different nodes. For instance, injections from generator 1 are now represented as simultaneous injections at nodes 2, 4 and 6, with the portion of the generator's injection at each of these nodes defined by the coefficients from the Ward reduction, as shown in III-D. Generator 5, on the other hand, is simply moved from node 8 to 7. Lastly, the demands formerly distributed among 11 nodes are now concentrated in 4 nodes, with the distribution determined again by the coefficients from the Ward reduction.

IV. Computational experiments

We assess our strategy on 20 power systems from [17], all of which are UCs with 36-hour planning horizons and hourly temporal resolution. As we are mainly interested in showcasing the benefits of our network-reduction approach, we transform the originally security-constrained UCs of [17] into deterministic UCs by ignoring contingencies. Also, in addition to scaling the unitary generation and start-up costs of each of them, we have modified the cases to induce congestion through tightening branch limits and increasing demand. By doing so, we aim at simulating cases where the system is stressed due to a lack of transmission capacity. Exclusively to the largest power system with 13,659 nodes, we ignore start-up costs to speed up convergence, as in none of our tests we were able to reach acceptable gaps with any of the network models if start-up costs were



(a) Full network.

(b) Reduced network.

Fig. 2: Network with 14 nodes, 11 demands, five generators, and 18 branches. The three possibly active branches are shown in red. Generators are shown as green elements and demands as red arrows.

TABLE I: General characteristics of the power systems with full network representation.

System	Nodes	Branches	Units	Cap. (MW)	Peak load (MW)	Gen. nodes	Load nodes	Possibly active branches
1354pegase	1,354	1,710	260	128,867	86,304	621	260	155
1888rte	1,888	2,308	296	89,842	45,672	939	284	194
1951rte	1,951	2,375	390	96,599	57,977	940	377	139
2383wp	2,383	2,886	323	29,135	16,941	1,817	323	167
2736sp	2,736	3,495	289	27,987	20,233	2,011	217	45
2737sop	2,737	3,497	267	26,290	15,952	2,008	205	45
2746wop	2,746	3,505	443	29,186	19,237	1,971	343	61
2746wp	2,747	3,505	457	31,147	22,705	1,991	346	64
2848rte	2,848	3,442	544	95,253	55,711	1,363	443	204
2868rte	2,868	3,471	596	98,307	51,530	1,391	492	166
2869pegase	2,869	3,968	510	239,856	144,454	1,304	510	82
3012polish	3,012	3,566	377	34,963	27,196	2,257	238	384
3120sp	3,120	3,684	483	35,523	22,591	2,266	328	131
3375wp	3,375	4,068	590	69,245	47,019	2,414	435	120
6468rte	6,468	8,065	1,262	189,477	77,933	3,313	930	119
6470rte	6,470	8,066	1,306	183,396	77,798	3,353	968	212
6495rte	6,495	8,084	1,352	184,479	76,467	3,290	1,008	85
6515rte	6,515	8,104	1,368	187,929	77,908	3,321	1,021	112
9241pegase	9,241	14,207	1,445	546,759	212,692	4,384	1,445	745
13659pegase	13,659	18,625	4,092	966,969	391,461	4,990	4,092	747

TABLE II: General characteristics of the optimization models with full network representation.

System	Binary vars.	B-theta: cont. vars	B-theta: constrs.	B-theta: non-zeros	PTDF: cont. vars.	PTDF: constrs.	PTDF: non-zeros
1354pegase	28,080	174,061	163,018	527,987	63,757	59,372	9,375,315
1888rte	31,968	240,301	218,044	695,001	89,245	74,486	8,824,391
1951rte	42,120	254,017	243,986	776,392	98,281	94,226	11,759,947
2383wp	34,884	297,433	253,131	795,086	107,749	70,123	17,980,729
2736sp	31,212	342,721	281,936	899,892	118,405	59,671	6,339,071
2737sop	28,836	341,641	278,220	887,154	117,217	55,872	6,361,012
2746wop	47,844	353,881	310,025	977,611	128,845	87,805	9,161,646
2746wp	49,356	354,997	313,814	990,161	129,961	91,859	10,061,944
2848rte	58,752	367,525	349,175	1,113,225	141,085	130,820	15,391,458
2868rte	64,368	373,717	362,385	1,153,295	145,513	141,093	18,531,035
2869pegase	55,080	379,837	334,496	1,090,369	133,705	92,322	13,490,334
3012polish	40,716	371,665	315,053	1,053,407	134,857	93,196	49,431,086
3120sp	52,164	389,701	350,295	1,106,391	144,757	111,176	20,520,044
3375wp	63,720	429,661	396,463	1,253,404	161,749	133,717	20,558,969
6468rte	136,296	839,269	796,725	2,586,530	316,081	279,312	44,778,967
6470rte	141,048	843,085	808,263	2,624,885	319,789	295,201	78,771,575
6495rte	146,016	847,945	819,047	2,659,395	323,101	298,647	35,532,156
6515rte	147,744	850,969	822,870	2,671,216	324,685	302,081	43,447,673
9241pegase	156,060	1,264,069	1,108,119	3,714,942	419,941	300,490	343,864,731
13659pegase	441,936	1,948,105	2,085,364	6,827,703	785,881	961,675	523,955,886

included. Our codes are written in Python and available at https://github.com/colonetti/wardUCPSCC2024. To tackle the optimization models, we use Gurobi 10.0.2 [4] through gurobipy without any decomposition. All of Gurobi's parameters are unchanged, except for the following: Method = 2, BarConvTol = 10^{-12}, Heuristics = 0.2, and Cuts = 3. We care to mention that it is likely that better settings can be found, both considering the average performance of Gurobi on the 20-case pool but specially considering its performance on individual cases and network models. Nonetheless, the settings

TABLE III: Optimization results with full representation of the network with B-theta formulation.

System	Root relax. ($)	Root relax. (sec.)	UB ($)	LB ($)	Gap (%)	Time (sec.)	Active bounds	Active branches
1354pegase	118,771	21	120,483	120,456	0.02	812	349	18
1888rte	153,534	29	153,614	153,534	0.05	31	1,533	87
1951rte	350,110	34	350,143	350,123	0.01	557	362	23
2383wp	119,752	77	119,958	119,958	0	3,562	121	7
2736sp	310,037	116	310,091	310,038	0.02	1,091	37	2
2737sop	241,365	108	241,679	241,434	0.1	7,200	98	5
2746wop	269,147	79	269,614	269,462	0.06	5,031	67	4
2746wp	333,944	106	334,209	333,957	0.08	1,785	70	5
2848rte	457,265	60	457,595	457,265	0.07	62	1,580	88
2868rte	752,870	82	753,025	752,870	0.02	87	499	36
2869pegase	399,824	104	400,081	399,914	0.04	336	24	2
3012polish	109,040	81	109,155	109,047	0.1	458	88	5
3120sp	196,647	84	196,796	196,647	0.08	94	87	6
3375wp	478,650	234	478,977	478,663	0.07	1,750	80	6
6468rte	69,700	1,109	69,930	69,711	0.31	7,200	502	21
6470rte	162,592	481	163,293	162,631	0.41	7,278	684	28
6495rte	108,131	1,011	108,317	108,206	0.1	7,205	272	14
6515rte	116,484	362	116,722	116,614	0.09	5,059	378	21
9241pegase	70,642	5,040	94,094	70,658	24.91	7,200	747	37
13659pegase	256,265	2,874	260,357	256,266	1.57	7,200	1,456	65

TABLE IV: Optimization results with full representation of the network with PTDF formulation.

System	Root relax. ($)	Root relax. (sec.)	UB ($)	LB ($)	Gap (%)	Time (sec.)	Active bounds	Active branches
1354pegase	110,535	42	120,484	120,375	0.09	787	345	18
1888rte	151,554	42	153,593	153,540	0.03	89	1,568	86
1951rte	350,092	57	350,347	350,136	0.06	147	367	24
2383wp	118,438	84	119,840	119,779	0.05	206	129	8
2736sp	309,403	33	310,304	310,037	0.09	82	33	2
2737sop	231,640	32	241,590	241,373	0.09	885	83	5
2746wop	258,720	42	269,679	269,418	0.1	213	62	4
2746wp	333,552	52	334,079	333,957	0.04	664	74	5
2848rte	453,105	81	457,525	457,278	0.05	257	1,581	87
2868rte	750,142	94	753,501	752,896	0.08	215	486	36
2869pegase	398,971	59	399,937	399,921	0	338	18	2
3012polish	108,983	276	109,097	109,040	0.05	784	93	7
3120sp	196,411	91	196,698	196,670	0.01	509	82	5
3375wp	468,801	96	478,982	478,661	0.07	1,638	72	8
6468rte	13,560	213	69,781	69,714	0.1	1,011	496	21
6470rte	19,877	319	162,771	162,625	0.09	2,388	701	28
6495rte	36,565	180	108,307	108,201	0.1	1,299	274	14
6515rte	33,242	226	116,709	116,593	0.1	3,875	379	21
9241pegase	55,677	2,004	73,349	70,849	3.41	7,200	648	33
13659pegase	243,744	2,525	-	256,190	-	7,200	-	-

chose perform relatively well on average for all cases. The time limit is set to 2 h for all experiments and the relative gap tolerance ($\frac{\text{Upper bound}-\text{Lower bound}}{\text{Upper bound}}$) is set to 0.1%. Finally, when the PTDF model is used, the flow bounds are added to the optimization model as lazy constraints with the lazy attribute set to 3. (We clarify here that when the PTDF model is used, no flow variable is added to the model and the flow expression (11) is directly plugged into (7) for the possibly active branches.) Lastly, for the PTDF model, we ignore all coefficients in the PTDF matrices whose absolute values are less than 10^{-4}. All experiments are conducted on a single computer with 128 GB of RAM and two Intel Xeon E5-2660 v3 2.60 GHz processors.

A. Results with full network representation

We start by presenting in Tab. I the general structural characteristics of the 20 power systems. This table shows the widely different characteristics of these power systems, in terms of system size, number of generating units, distribution of generation and load over the system, system loading (ratio of peak load and installed capacity), and apparent possible branch congestions. We emphasize two of these characteristics: system loading, and the number of possibly active branches. The first is usually a good indicator that branch capacities might be reached as more power is expected to flow as the demand increases. (Incidentally, with the growing participation of renewable energy, a new phenomenon can be seen: congestions due to increasing amounts of non-dispatchable renewable generation.) Nevertheless, the possibility of congestion is more directly seen by the number of possibly active branches. These are branches whose redundancy could not be unequivocally guaranteed by our implementation of the branch bound redundancy algorithm of [1]. Following this metric, the possibly most congested system is 3012polish with nearly 11% of its branches possibly reaching its limits. Evidently, in practice, considerably less than 11% of the systems' branches will reach its capacity, if any. The characteristics in Tab. I are then reflected on the problem sizes reported in Tab. II. Comparing both tables clearly shows the sharp increase in the number of variables, constraints and non-zeros as the network and the number of units increase. From Tab. II, we underline the model sizes for the six systems with more than 6,000 nodes. As we show in the following, these numbers are reflections of a model whose size and complexity requires running times that are prohibitive for UC.

Finally, in Tab. III and IV, we show the results of solving the UC for the 20 power systems with Gurobi, respectively with formulation B-theta and formulation PTDF. In these tables columns “Root relaxation ($)“ and “Root relaxation (sec.)“ refer to, respectively, the optimal value and the total Gurobi running time of the root relaxation of the branch-and-cut algorithm. Columns UB and LB are respectively the final upper bound and lower bound found by Gurobi. Lastly, column “Active bounds“ refer to the total number of occurrences of active flow bounds over the 36 periods. Similarly, column “Active branches“ presents how many of the system’s branches have one of its bounds active in at least one of the periods. Both columns are used as indicators of the level of congestion of the network. Before diving in the analysis of the results, we urge the reader not to use Tab. III and IV to compare the B-theta and PTDF formulations; any comparison between them could be biased by our implementation and the solver and settings used. A fair comparison would require specialized models and algorithms for these formulations, and it is beyond the scope of this work. Tab. III and IV are, instead, used to show how the size of the network representation may impact on the running time of the solver.

From Tab. III and IV, we see a clear increase in the running times of the root relaxation for both formulations. Naturally, this increase is not only due to the network, as the number of generating units also normally increases with the number of nodes. On the other hand, inspecting the optimization times, we cannot see a clear pattern of increase or decreases in running times because this metric includes the branch-and-cut phase of Gurobi’s algorithm and it is heavily dependent on the combinatorial aspects of the problem. For both formulations, we see that the relative number of active branches is low across all cases, as expected: ranging from 0.05%, for system 2869pegase, to 3.77%, for system 1888rte. Specially for systems 1888rte and 2869pegase itself, many of the active branches are branches that are the only connection of a node to the rest of the network. With the B-theta formulation, Gurobi is able to reach the 0.1% relative gap for 14 of the 20 cases, while it is successful for 18 cases with the PTDF formulation. These tables show a considerable increase in the running times for both formulations when the 6,000-node threshold is passed. For both formulations, Gurobi is not able to reach convergence for systems 9241pegase and 13659pegase.

B. Results with the degree-1 node methodology of [8]

We summarize the results obtained by applying the methodology of [8] in Tab. V. As we can see, the reduction in the number of nodes and branches is significant. The number of retained nodes varies from 45% to 87%, while the retained branches go from 55% to 90%. In the methodology of V, there is a one-to-one reduction in nodes and branches: one branch removed for each node removed. However, the significant reductions in network size are not reflected on Gurobi’s running times: for most cases, there is in fact an increase in running time, with a average increase of 34% for the B-theta formulation and 8.5% for the PTDF formulation. We care to note that these increases in running time are likely due to different algorithmic choices made by Gurobi when solving the reduced model. While a fairer comparison would require making sure that the same algorithmic choices are made for the reduced and full model, this is extremely hard in practice and beyond the scope of this work.

C. Results with the proposed reduction methodology

We present in Tab. VI the main characteristics of the reduced system. With our methodology, the reduction in the number of nodes and branches is even more pronounced: the reduced system with the largest percentage of retained nodes is 9241pegase with only 16.2% of the original nodes and 36.1% of the original number of branches. The system with the largest reduction in the number of nodes is 2848rte for which only 6.8% are needed. On average over the 20 systems, our methodology yields reduced systems with only 10.3% of the original nodes and 26.7% of the original number of branches. It is also interesting to see from this table that the number of nodes with generation and load is also significantly reduced, showing that, although generators now have their outputs split among more nodes, the total number of nodes with injections is still reduced. Similarly, the number of possibly active nodes also decreases due to both branch limits being converted directly to bounds on generation and also to appropriately combining series and parallel branches. Although, as shown in this table, the time taken to reduce the network is considerable, specially for the two largest systems, the benefits from the reduction far out weights these times. Next, we present in Tab. VII the impact of the network reduction on the size of the optimization models. Except from the number of constraints in the PTDF formulation, all other metrics show considerable advantages. For instance, the number of continuous variables in the B-theta formulation decreases, on average, by 76% with the proposed network reduction technique. The number of constraints for this formulation sees a reduction of 56% on average, and the number of non-zeros in the coefficient matrix decreases on average by 49%. The model size reductions are also notable for the PTDF formulation, for which there is an average decrease of 68% in the number of continuous variables, and 74% in the number of non-zeros. In the B-theta formulation, when a node is removed, its associated voltage angle and slack variable are removed. Similarly, when a branch is removed, its flow variable is no longer necessary. The number of non-zeros, on the other hand, is not as easily anticipated because, if a node with generation is removed, its generation is split among all its boundary nodes, thus increasing the number of non-zero terms associated with the generation variable. On the other hand, for the PTDF formulation, removing a node from the network eliminates its slack from the optimization model, and all terms in the flow expressions and power balances where such slack appears are removed. For both formulations, the more compact representation of the network results in remarkable speed-ups in Gurobi’s running time. Tables VIII and IX present the main optimization results for the B-theta and PTDF formulation, respectively. The first benefit we see in Tab. VIII is that, with the reduced network, Gurobi is now able to solve 19 out of the 20 cases with the B-theta formulation. The only case left unsolved is 9241pegase, for which, with our technique, Gurobi is now able to reach a relative gap of 4.46% — against the

TABLE V: Results with the network reduction proposed by [8].

System	Nodes	Branches	B-theta: LB ($)	B-theta: gap (%)	B-theta: (sec.)	PTDF: LB ($)	PTDF: gap (%)	PTDF: (sec.)
1354pegase	730	1,086	120,398	0.08	1,973	120,389	0.08	578
1888rte	886	1,306	153,535	0.06	30	153,538	0.03	82
1951rte	895	1,319	350,110	0.02	36	350,135	0.02	116
2383wp	1,733	2,236	119,953	0.00	4,431	119,792	0.09	194
2736sp	2,395	3,154	310,038	0.03	880	310,040	0.02	60
2737sop	2,396	3,156	241,420	0.10	4,603	241,372	0.09	1,796
2746wop	2,392	3,151	269,444	0.09	4,038	269,497	0.04	947
2746wp	2,393	3,152	333,959	0.07	2,881	333,954	0.08	388
2848rte	1,382	1,976	457,265	0.07	53	457,278	0.01	173
2868rte	1,389	1,992	752,870	0.02	191	752,900	0.06	169
2869pegase	1,989	3,088	399,912	0.01	671	399,919	0.02	261
3012polish	2,301	2,855	109,049	0.10	445	109,040	0.06	555
3120sp	2,385	2,949	196,647	0.08	81	196,669	0.01	198
3375wp	2,511	3,205	478,658	0.10	5,627	478,661	0.07	662
6468rte	3,796	5,393	69,711	0.12	7,200	69,715	0.07	1,498
6470rte	3,783	5,379	162,631	0.17	7,200	162,621	0.09	2,243
6495rte	3,771	5,360	108,217	0.09	7,200	108,198	0.10	970
6515rte	3,773	5,362	116,565	0.19	7,200	116,589	0.10	5,607
9241pegase	7,374	12,340	70,660	19.75	7,200	70,835	1.11	7,200
13659pegase	7,374	12,340	-	-	7,200	256,200	-	7,200

TABLE VI: General characteristics of the power systems with reduced network representation. In this table, ”PAB” stands for ”possibly active branches”, and column ”R (sec.)” is the time in seconds taken to reduce the network, not including the time necessary to identify redundant bounds.

System	Nodes	Branches	PAB	G. nodes	Load nodes	R (sec.)
1354pegase	152	428	139	112	129	1
1888rte	167	467	140	105	154	1
1951rte	186	492	131	121	178	1
2383wp	295	822	167	230	266	3
2736sp	320	1,247	45	233	299	13
2737sop	317	1,254	44	232	293	14
2746wop	324	1,235	61	271	299	13
2746wp	321	1,224	63	270	297	12
2848rte	194	639	120	148	181	5
2868rte	212	644	139	161	194	4
2869pegase	319	1,162	85	269	298	13
3012polish	464	1,064	384	258	410	4
3120sp	300	976	131	239	281	7
3375wp	345	1,123	119	290	322	12
6468rte	484	1,710	119	416	450	30
6470rte	544	1,770	210	444	481	27
6495rte	457	1,654	85	402	419	28
6515rte	472	1,664	111	414	437	27
9241pegase	1,497	5,133	712	1,023	1,241	162
13659pegase	1,471	5,139	708	1,059	1,337	141

24.91% gap with the full representation of the network and B-theta formulation (Tab. III). For the PTDF formulation, the cases successfully solved remains the same but, again, with the reduced network, Gurobi is again able to reach significantly better gaps for the unsolved cases: instead of the 3.41% gap for the 9241pegase with the full network, Gurobi reaches 0.38% with the reduced model; with the full network, Gurobi is not able to find a feasible solution for 13659pegase, on the other hand, with the reduced model, Gurobi finds a solution within a gap of 0.6%. In part, the better convergence rates achieved with the reduced network are results of a linear relaxation that is, according to the evidences shown here, more amenable to the solver. The more compact formulation given by the reduced network, as we have shown in VII, render linear relaxations that are solved, on average, nearly 60% faster for both formulations. Despite the gains in running times, the lower bounds given by the linear relaxation of the reduced model remain practically the same as those obtained with the full representation, which is expected since the reduced model is equivalent to the full representation as long as there is no significant load shedding.

In addition to being able to reach convergence for more cases, the reduced network also result in considerable speed-ups. Comparing Tab. VIII against Tab. III, and Tab. IX against Tab. IV, we see that, with the B-theta formulation, Gurobi reaches the gap tolerance faster with the reduced network for 18 of the cases. The only case for which convergence is reached faster with the full network is 1888rte, which takes 31 seconds with the full network and 151 seconds with the reduced network. The average speed-up for this formulation is 36% if we consider the individual speed-ups for each case; the speed-up computed considering the total time to solve all cases (64,000 seconds for the full network, and 21,225 seconds for the reduced network) is 66%. The same metrics for the PTDF formulation are 10% and 20%, respectively.

Finally, inspecting the bounds obtained with the reduced formulations in Tab. VIII and IX reveals that both upper and lower bounds, for both formulations, are nearly the same as those achieved with the full representation. This is again expected since there is negligible load shedding in all cases. For the convergent cases, the one with the most load shedding is 1951rte, for which there is a total of 30 MWh load shedding in the solution obtained with the full network representation and PTDF formulation, and 10.4 MW for the same formulation and the reduced network. Furthermore, our choice for the PTDF coefficient threshold of 10^{-4} allows for small violations of the branch flows for case 2383wp, which explains why the upper bound obtained with the reduced network for this formulation (Tab. IX) is lower than the lower bound obtained with the full network representation (Tab. IV), in spite of no occurrence of load shedding. Lastly, note that the number of active bounds and branches reported in Tab. VIII and IX are significantly fewer than those in Tab. III and IV. As we mentioned earlier, most of the active bounds and branches in Tab. III and IV are from branches connected to nodes connected to the system by a single branch. In the reduced system, these bounds are converted into constraints on the

TABLE VII: General characteristics of the optimization models with the reduced network representation w.r.t. to the characteristics of the full representation. For instance, if the number of continuous variables is a for the full representation and b for the reduced network, then, in this table, the relative number of continuous variables is $\left(\frac{b}{a}\right)$.

System	B-theta: cont. vars (%)	B-theta: constrs. (%)	B-theta: non-zeros (%)	PTDF: cont. vars. (%)	PTDF: constrs. (%)	PTDF: non-zeros (%)
1354pegase	24	45	51	32	99	28
1888rte	21	41	46	30	98	33
1951rte	23	46	51	35	100	30
2383wp	24	41	46	30	100	24
2736sp	26	40	47	27	100	22
2737sop	25	40	46	26	100	22
2746wop	28	46	52	32	100	26
2746wp	28	46	53	33	100	26
2848rte	21	44	50	32	98	27
2868rte	22	46	51	34	99	28
2869pegase	25	42	49	31	100	27
3012polish	26	42	50	32	100	26
3120sp	23	43	50	30	100	24
3375wp	25	46	52	33	100	24
6468rte	21	44	52	32	100	25
6470rte	23	46	53	33	100	25
6495rte	21	45	53	33	100	26
6515rte	22	45	53	33	100	25
9241pegase	30	45	51	34	99	27
13659pegase	30	56	61	44	100	31

injections of the node removed.

V. Conclusions

In this work, we present a novel application of the Ward reduction to UC models. Our motivation is based on the computational burden introduced by the network representation in UC models. This burden is due to the large number of network elements, and also to the spatial coupling introduced by it. Our main premise, widely used and also supported by evidence, is that a large number of redundant branch flows' bounds can be effectively identified. Based on this premise, we leverage the redundancy of the branch flows' bounds to remove nodes and branches from the network using the well-known Ward reduction. We show that appropriately applying the Ward reduction enables us to obtain a more compact representation of the network that is equivalent under mild conditions. Then, we assess our methodology on a pool of 20 power systems ranging from 1,354 to 13,659 nodes. We show that, with our network reduction, we can represent theses systems with as few as 6.8% of the original nodes. In turn, the network reductions provides speed-ups from 10% to 36% for two of the most widely used formulations for the DC network model.

Data availability

Codes and data are available at https://github.com/colonetti/wardUCPSCC2024.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and in part by Jirau Energia through the Agência Nacional de Energia Elétrica (ANEEL) R&D Project under Grant P&D-06631-0015-2023.

References

[1] Ali Jahanbani Ardakani and François Bouffard. Identification of umbrella constraints in dc-based security-constrained optimal power flow. *IEEE Transactions on Power Systems*, 28(4):3924–3934, 2013.

[2] Wolfgang Biener and Klaus René Garcia Rosas. Grid reduction for energy system analysis. *Electric Power Systems Research*, 185:106349, 2020.

[3] Xu Cheng and Thomas J Overbye. Ptdf-based power system equivalents. *IEEE Transactions on Power Systems*, 20(4):1868–1876, 2005.

[4] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.

[5] Ziming Ma, Haiwang Zhong, Tong Cheng, Junbo Pi, and Fanlin Meng. Redundant and nonbinding transmission constraints identification method combining physical and economic insights of unit commitment. *IEEE Transactions on Power Systems*, 36(4):3487–3495, 2021.

[6] Saurav Mohapatra, Wonhyeok Jang, and Thomas J Overbye. Equivalent line limit calculation for power system equivalent networks. *IEEE Transactions on Power Systems*, 29(5):2338–2346, 2014.

[7] HyungSeon Oh. Aggregation of buses for a network reduction. *IEEE transactions on power systems*, 27(2):705–712, 2012.

[8] James Ostrowski and Jianhui Wang. Network reduction in the transmission-constrained unit commitment problem. *Computers & Industrial Engineering*, 63(3):702–707, 2012.

[9] Salvador Pineda, Juan Miguel Morales, and Asunción Jiménez-Cordero. Data-driven screening of network constraints for unit commitment. *IEEE Transactions on Power Systems*, 35(5):3695–3705, 2020.

[10] Quentin Ploussard, Luis Olmos, and Andres Ramos. An efficient network reduction method for transmission expansion planning using multicut problem and kron reduction. *IEEE Transactions on Power Systems*, 33(6):6120–6130, 2018.

[11] Álvaro Porras, Salvador Pineda, Juan Miguel Morales, and Asunción Jiménez-Cordero. Cost-driven screening of network constraints for the unit commitment problem. *IEEE Transactions on Power Systems*, 38(1):42–51, 2022.

[12] HK Singh and SC Srivastava. A reduced network representation suitable for fast nodal price calculations in electricity markets. In *IEEE Power Engineering Society General Meeting, 2005*, pages 2070–2077. IEEE, 2005.

[13] Brian Stott, Jorge Jardim, and Ongun Alsaç. Dc power flow revisited. *IEEE Transactions on Power Systems*, 24(3):1290–1300, 2009.

[14] Harald G Svendsen. Grid model reduction for large scale renewable energy integration analyses. *Energy Procedia*, 80:349–356, 2015.

[15] J. B. Ward. Equivalent circuits for power-flow studies. *Transactions of the American Institute of Electrical Engineers*, 68(1):373–382, Jul 1949.

[16] Richard Weinhold and Robert Mieth. Fast security-constrained optimal power flow through low-impact and redundancy screening. *IEEE Transactions on Power Systems*, 35(6):4574–4584, 2020.

[17] Alinson S Xavier, Aleksandr M Kazachkov, Ogün Yurdakul, and Feng Qiu. Unitcommitment. jl: A julia/jump optimization package for security-constrained unit commitment (version 0.3). *JuMP Optimization Package for Security-Constrained Unit Commitment (Version 0.3), Zenodo*, 2022.

[18] Álinson S Xavier, Feng Qiu, and Shabbir Ahmed. Learning to solve large-scale security-constrained unit commitment problems. *INFORMS Journal on Computing*, 33(2):739–756, 2021.

TABLE VIII: Optimization results with reduced representation of the network with B-theta formulation.

System	Root relax. ($)	Root relax. (sec.)	UB ($)	LB ($)	Gap (%)	Time (sec.)	Active bounds	Active branches
1354pegase	118,771	10	120,489	120,442	0.04	305	125	7
1888rte	153,535	18	153,562	153,536	0.02	151	27	7
1951rte	350,110	14	350,146	350,116	0.01	101	14	2
2383wp	119,741	23	119,902	119,800	0.09	1,468	130	8
2736sp	310,037	55	310,079	310,038	0.01	541	34	2
2737sop	241,365	32	241,616	241,380	0.1	491	95	4
2746wop	269,147	36	269,564	269,417	0.05	2,816	66	4
2746wp	333,944	45	334,146	333,946	0.06	502	43	4
2848rte	457,265	44	457,594	457,265	0.07	46	72	4
2868rte	752,870	63	753,092	752,870	0.03	65	66	2
2869pegase	399,824	42	399,949	399,908	0.01	318	18	2
3012polish	109,041	43	109,150	109,043	0.1	188	81	5
3120sp	196,647	32	196,797	196,647	0.08	35	17	4
3375wp	478,650	56	478,947	478,658	0.06	505	66	4
6468rte	69,700	128	69,778	69,715	0.09	984	495	21
6470rte	162,592	156	162,786	162,643	0.09	1,215	660	27
6495rte	108,134	149	108,303	108,203	0.09	1,265	273	14
6515rte	116,488	132	116,701	116,613	0.08	1,888	341	19
9241pegase	70,642	429	74,134	70,828	4.46	7,200	701	32
13659pegase	256,265	917	256,406	256,266	0.05	1,141	921	40

TABLE IX: Optimization results with reduced representation of the network with PTDF formulation.

System	Root relax. ($)	Root relax. (sec.)	UB ($)	LB ($)	Gap (%)	Time (sec.)	Active bounds	Active branches
1354pegase	112,442	22	120,504	120,399	0.09	294	127	7
1888rte	153,372	17	153,566	153,542	0.02	62	27	7
1951rte	350,092	22	350,238	350,135	0.03	60	16	2
2383wp	118,438	25	119,831	119,796	0.03	57	129	8
2736sp	309,403	11	310,067	310,044	0.01	104	33	2
2737sop	231,640	10	241,576	241,387	0.08	160	92	4
2746wop	258,720	14	269,718	269,496	0.08	534	58	4
2746wp	333,552	20	334,085	333,957	0.04	525	41	4
2848rte	456,918	45	457,384	457,284	0.02	539	70	4
2868rte	750,142	40	753,086	752,890	0.03	136	59	2
2869pegase	398,971	35	400,024	399,922	0.03	467	22	2
3012polish	108,984	64	109,091	109,054	0.03	717	88	7
3120sp	196,617	33	196,825	196,671	0.08	952	16	3
3375wp	468,801	41	478,856	478,665	0.04	783	68	6
6468rte	13,560	79	69,767	69,713	0.08	969	495	21
6470rte	19,877	116	162,776	162,634	0.09	1,213	666	27
6495rte	36,565	72	108,305	108,198	0.1	823	274	14
6515rte	33,242	77	116,696	116,602	0.08	924	345	20
9241pegase	55,677	586	71,253	70,984	0.38	7,200	720	32
13659pegase	243,768	916	257,810	256,261	0.6	7,200	715	38