STRUCTURE- & PHYSICS- PRESERVING REDUCTIONS OF POWER GRID MODELS

COLIN GRUDZIEN †, DEEPJYOTI DEKA‡, MICHAEL CHERTKOV¶, AND SCOTT N BACKHAUS∥

Abstract. The large size of multiscale, distribution and transmission, power grids hinder fast system-wide estimation and real-time control and optimization of operations. This paper studies graph reduction methods of power grids that are favourable for fast simulations and follow-up applications. We present systematic techniques that reduce the grid size while preserving basic functionality in the reduced grid. The reduction is achieved through the use of iterative aggregation of sub-graphs that include general tree structures, lines, and triangles. An important feature of our reduction algorithms is the efficient use of hash-tables to store the sequential reduction of the original grid in the reduced network. This allows for easy introduction of detailed models into the reduced conceptual network and makes the model backward compatible with varying resolution. The performance of our graph reduction algorithms, and features of the reduced grids, are discussed on a real-word transmission and distribution grid. We produce visualizations of the reduced models through open source libraries and release our reduction algorithms with example code and toy data.

Key words. power grids, networks, graph reduction, visualization

AMS subject classifications. 68Q25, 68R10, 68U05, 94C15

1. Introduction. Power grids consist of the network of transmission and distribution lines connecting generators with end-users, enabling the transfer of electricity. The power grid of North America, in particular, is recognized as the most complicated machine built on earth [1, 30]. Topologically the grid is represented by a large, connected graph with nodes denoting buses (loads and generation) and edges representing lines. These nodes and edges are constructed in distinct formations across the physical scales in the problem of power delivery [17]. The transmission and distribution sub-networks exist in a hierarchical configuration, where the transmission sub-network consists of high voltage lines connecting generators to substations and the distribution sub-network connects substations to end users [10]. System-wide monitoring and control of the grid involves simulation studies carried out by network authorities like independent system operators (ISO) [19, 24]. Simulating grid operations relies on accurate state estimation and optimization with respect to power flow laws, describing interactions across layers of temporal and spatial resolution [27].

Over time, the grid and its dynamical characteristics undergo changes with the introduction of new loads, generators and network components. The increased penetration of the renewable energy, e.g., solar and wind power, has expanded the frontiers of the grid and also made issues regarding grid stability and control of paramount importance [3, 31, 13]. Dynamic forcing from the distribution grid has historically been
much smaller than the transmission components. For example, the amount of inertia and damping in the distribution grid are limited [18]. However, rooftop solar, the internet of things [22], and other resources have cultivated the demand for decentralized resource generation and control in the distribution sub-grid [14, 25, 12, 9]. With this demand comes the need for multiscale, dynamical models of the grid.

Owing to the large size and dense interconnections, control, optimization and dynamical simulation of the detailed grid face implementation issues [5, 4]. Operational demands require reduced order and approximate schemes to improve the efficiency of computations and simulations of grid operations. However, one must ensure that the model reduction schemes are true to the original grid and have comparable dynamic behavior, or approximate the same. This paper analyses system-aware graph reductions of large power grids, to construct conceptual networks amenable for follow-up action and reanalysis. System-awareness here refers to the use of parameters such as nodal voltage, line impedance and bulk nodal inertia; we additionally aim to preserve topological features like nodal degree, presence of graph paths etc. in the graph reduction process. This is necessary for our underlying goal which is to develop graph reduction schemes that preserve qualitative features of the original grid’s dynamic behavior for accurate state estimation and disturbance prediction.

1.1. Contribution. There is extensive research into reduction algorithms for improving the analysis of large networks and reducing the computational burden therein. Community detection approaches use graph based methods to collapse subnetworks into smaller, representative components, e.g. Kannan et. al. analyze criteria for effective clustering approaches in relation to the spectrum of the graph Laplacian [21]; Newman develops reduction methodology in terms of the node-group connectivity measure of modularity [26]. While community detection methods have applications in power systems, these approaches aren’t appropriate for constructing a dynamically consistent reduced order model. Other works, in circuit design, have focused on network reductions which preserve static power flow computations, e.g. Zhou et. al. study block based hierarchical graph reduction schemes for fast solution to power flow for in-chip circuits [35]; Wang provides a deterministic random walk based preprocessing and graph reduction algorithm also aimed at solving the DC power flow problem [32]; Chen & Chen present Krylov-subspace iterative methods for preconditioning [7].

Dorfler & Bullo, in particular, provide a detailed mathematical analysis of the classical Kron reduction [11] for power grid networks. Given a choice of reference nodes, the Kron reduction uses Gaussian elimination to pare down the full network to a reduced model that is electrically equivalent from the perspective of the references. The Kron reduction has been applied extensively in power systems analysis with success in control and optimization problems [11, and references therein]. The selection of reference nodes is unambiguous for transmission networks under a classical, hierarchical distribution design. However, the deployment of decentralized generation and storage in the distribution sub-grid makes the selection of the references problematic: while individual distribution nodes do not provide significant generation, the aggregation of these can strongly impact the optimal power flow and control problem.

Our work is distinguished by graph reductions designed to respect the dynamics and topology of multiscale, distribution and transmission, electric grid networks. Our key contribution is developing a series of invertible graph reduction algorithms and demonstrating the viability of these techniques on a real, electric grid in a US Midwest utility. Our methodology emphasizes two design features: (i) the reductions
are system aware, respecting the network topology and grid parameters, and (ii) the reductions may be partially inverted. System-awareness enforces that the reduced network respects the power flow of the full network, but complex dynamical features can be given increased resolution by inverting the nodal clustering post-facto. Unstructured data, describing the placement of clustered nodes in the reduced model, can be used to parameterize net power flow. However, this unstructured data is insufficient to increase the resolution on a specific cluster. Effective implementation of data structures, tracking sequential reductions, has been an integral component of our work: our design enforces backwards compatibility, with intermediate, partially reduced representations of network features. Implementing these techniques on a real, multiscale electric grid, we present the results of our analysis comparing the graph characteristics of reduced network to the full network. We interactively visualize the reduced network, utilizing graph-topology, rather than geographical location to qualitatively analyze the results.

In section 2, we present our main results, including our reduction algorithms and the analysis of their performance on the real, multiscale network. In section 3 we demonstrate conceptual visualizations the reduced, case study network, and the nodal clustering produced by the algorithms. We detail our use of data structures in Appendix A, explaining how to invert the algorithms to increase the resolution post-facto. Utilizing the reduced network for online, dynamic modelling of multiscale electric grids is discussed in section 4 where we introduce future directions of research. Finally, example code and test data are available as supplementary material online [16] with interactive visualizations available in web browsers [15].

2. Reduction algorithms. This section develops our algorithms for the sequential reduction of a multiscale, distribution and transmission, power grid network — a reference for our notations can be found in Figure 1. Our methodology is designed to achieve the following objectives:

1. the reduced network faithfully represents the original network’s power flow;
2. the full network can be (partially) reconstructed from the reduced model;
3. the reduced network is of a scale that is amenable to interactive visualization.

We focus on off-line methods meeting the above objectives. As an off-line reduction, we use static network characteristics to produce a model which is robust to changes in the production, consumption, and dynamic characteristics of loads and generators. In particular, the same reduced network will be used to represent different load and parameter states including the inertia, damping and frequency control settings. Any such reduction implicitly assumes infrequent updates. These updates occur when topology and parameter estimations (line impedances, transformer settings, and related) undergo a significant change, e.g. in seasonal transitions or following major network modifications. With such schemes, operators infrequently need to reproduce reduced order models (and their associated parameterizations). However, the scheme can be improved with state estimation available through phasor measurement unit
(PMU) technology in real time [29], i.e. on-line. The construction of a robust reduction scheme involving online state estimation will be the subject of future research.

In the remainder of this section we discuss four consecutive sub-steps of our off-line scheme and analyze the performance of the algorithms on a real multiscale network.

2.1. Degree zero reductions. As a preliminary step, we remove nodes of zero nominal voltage and restrict the test network to a single connected component, representing a full regional power grid, as described in the following definition.

**Definition 1.** The set \( N \) is defined to be the set of nodes in the pre-processed network, consisting of a single connected component with all nodes of nominal voltage greater than zero. The edge set for this network is denoted \( E \) and defines a symmetric connectivity matrix. Every node \( b \in N \) may be identified with its row/column number in the connectivity matrix so that we associate \( b \) equivalently with this index. We will denote the undirected line connecting the two nodes \( b_0, b_1 \) by the set \( \{b_0, b_1\} \in E \).

The connected sub-network contains 53,155 nodes, 63,832 edges, 268 PMU devices and 4,332 generators. The distribution of node degrees is given in Figure 4. The mean node degree is 2.40 with standard deviation of 1.61, and a max degree of 40. Our analysis is performed by manipulating the connectivity matrix, but we describe the reduction algorithms at a high level with the node and edge sets, \( N \) and \( E \).

2.2. Degree one reductions. While the physical lines constituting the distribution sub-network form meshed, loopy graphs, the operational topology for load balancing consists of radial tree structures [9]. Operational switching disconnects lines and the meshed topology so that the substations, connected to the transmission network, form roots of disjoint trees in the distribution sub-networks. This structure differs significantly from the transmission network which typically has multiple loops energized at all times to guarantee continuous delivery to the substations [8, 9]. In an operational window where the radial structure is unchanged, the distribution graph structure lends itself to an intuitive representation of the network’s multiscale coupling. We map disjoint distribution trees to their respective roots at substations — in our conceptual network, these terminal roots form super nodes which are used to represent the entire behaviour of the tree. Dynamically this is realized by modelling the net power flow of the entire tree through the terminal super node.

Confining our analysis to the connected network \( N \), we collapse all trees in the network to their root nodes. This reduction is performed unambiguously by recursively mapping each node of degree one into the node with which it shares a line. The recursive step is performed until all nodes in \( N \) are of degree two or greater. Our method is described in Algorithm 1. To post process, and refine the graph structure, our design allows one to invert the collapse of any subset of a tree in the network; we use hashable maps [33] for ease of implementation in this reconstruction. For each terminal super node in which we cluster a tree, we associate a sequence of lists and arrays representing the recursive reduction procedure. This implementation is described comprehensively in Appendix A. We define the following notation.

**Definition 2.** The degree of the node \( b_0 \) is denoted \( \text{deg}(b_0) \). The data structure \( d1H \) is a hashable map, \{"field": "data"\}, where “data” is an ordered list. Any subset of nodes \( t \subset N \) is defined as a tree if it is collapsed to a node under Algorithm 1. The mapping which collapses a tree, or a collection of trees, to the root node \( b_0 \) is associated to the field \( t_{b_0} \) in \( d1H \).

In each loop of Algorithm 1 we collapse the degree one node \( b_0 \) into the connected
Algorithm 1 Degree one reduction

Define: \( d1N := N, d1E := E, d1H := \text{empty hashable map.} \)

\[
\text{while } \exists b_0 \in d1N \text{ with } \deg(b_0) < 2, \text{ do}
\]

Remove \( b_0 \) from \( d1N \) and line \( \{b_0, b_1\} \) from \( d1E \).

if \( t.b_0 \in d1H \), then

Append \( b_1 \) to the end of each array stored under \( t.b_0 \in d1H \).

Append all arrays in \( t.b_0 \in d1H \) to the list \( t.b_1 \in d1H \).

Remove \( t.b_0 \) from \( d1H \).

else

Write array \([b_0, b_1]\) to a list \( t.b_1 \in d1H \).

end if

end while

return \( d1N, d1E, d1H \)

node \( b_1 \). The if statement requires that whenever a list of collapsed trees is associated to \( t.b_0 \in d1H \), we append all associated arrays to the list \( t.b_1 \), and \( b_1 \) is appended to each array denoting the root node. Algorithm 1 reduces the test network to 32,891 nodes and 43,568 edges. The histogram of tree lengths and the distribution of the degrees of the nodes in \( d1N \) are given in the Figure 4. The mean degree of nodes in \( d1N \) is 2.65, with a standard deviation 1.42 and maximal degree of 38. Tree lengths are calculated as the number of nodes aggregated into the super node, including the root node itself. The total number of trees collapsed in \( d1H \) is 9,528 with a mean tree length 3.12 nodes, a standard deviation 2.41 and a max tree length of 36 nodes.

The reductions to the test network via Algorithm 1 are significant, yet in an on-line reduction we may expect a further collapse yet. In our study, the degrees of nodes in the distribution network are defined by the physical lines connecting nodes, irrespective of the operational disconnecting. In practice, however, the operational switching for real power delivery further sparsifies the network and forms additional tree structures that would be collapsed under Algorithm 1. The operational structure typically changes in response to system faults and outages which may occur a few times a day [9]. Therefore, on-line graph reduction faces the additional challenge of efficiently learning the operational topology, based on incomplete information, and constructing a reduced model within the window of the current configuration.

2.3. Degree two reductions. Geographically distant sub-networks that have significant generation or load resources are linked for robustness of power delivery. In case of line failures within one area, the interconnected sub-networks can be configured to balance loads and generation around the failure. In a setting where the intermediate area between these sub-networks has low generation or load, the connection between them is comprised of long range transmission lines, as seen in grids in the USA, China and others [28]. These transmission lines are topologically modelled as string-like line sub-graphs of degree two nodes. Often, the intermediate nodes in the string lack significant generation or load and have negligible impact to network dynamics. The simple dynamical transmission structure of these degree two nodes motivates an intuitive model of the power flow: we replace all nodes in the interior of the string with a “meta-edge” and parameterize the net power flow with line characteristics. Figure 2 visualizes this reduction in which we approximate the string of degree two nodes, \( b_0 \) through \( b_4 \), with a single line connecting \( b_0 \) and \( b_4 \).

Recursively removing degree one nodes from \( N \) as described earlier produces
the network \( d_{1N}, d_{1E} \) comprised of nodes degree two or greater. Our subsequent reduction removes all nodes of degree two by recursively replacing degree two nodes with edges, if the edge does not already appear in \( d_{1E} \). Prohibiting double edges between nodes, our reduction has the additional effect of reducing other tree-like configurations. These structures are discovered when the procedure results in a degree one node in \( d_{1N} \).

**Definition 3.** Let \{\( b_0, b_1, b_2 \)\} \( \subset d_{1N} \) and \( \{b_0, b_1\}, \{b_0, b_2\}, \{b_1, b_2\} \in d_{1E} \) such that \( \text{deg}(b_0) = \text{deg}(b_1) = 1 \), while \( \text{deg}(b_2) \geq 3 \). The set \( \{b_0, b_1, b_2\} \subset d_{1N} \) is defined to be a sparsely connected triangle.

Let \( \{b_0, b_1, b_2\} \) be a sparsely connected triangle as in Figure 3. Removing \( b_0 \), and the edges \( \{b_0, b_1\} \) and \( \{b_0, b_2\} \), lowers the degree of \( b_1 \) to one. In Lemma 1 we demonstrate that a degree one node is produced by Algorithm 2 if and only if a sparsely connected triangle is reduced this way.

**Lemma 1.** Let \( b_0 \in d_{1N} \) be a node of degree two with edges \( \{b_0, b_1\} \) and \( \{b_0, b_2\} \). A degree one node is produced by replacing \( b_0 \) with the edge \( \{b_1, b_2\} \), prohibiting double lines, if and only if \( \{b_0, b_1, b_2\} \) is a sparsely connected triangle.

**Proof.** By construction, every node in \( d_{1N} \) is of degree two or greater. The nodes \( b_1 \) and \( b_2 \) each lose an edge in the reduction, \( \{b_0, b_1\} \) and \( \{b_0, b_2\} \) respectively. However, if \( \{b_1, b_2\} \) isn’t an edge we add this edge to the network. Therefore \( b_1 \) and \( b_2 \) remain the same degree if and only if \( \{b_1, b_2\} \notin d_{1E} \). Suppose removal of the edges \( \{b_0, b_1\} \) and \( \{b_0, b_2\} \) has produced a degree one node. We conclude \( \text{deg}(b_1) = 2 \) or \( \text{deg}(b_2) = 2 \), and \( \{b_1, b_2\} \in d_{1E} \). Without loss of generality suppose \( \text{deg}(b_1) = 2 \). The edges connecting \( b_1 \) are therefore \( \{b_0, b_1\} \) and \( \{b_1, b_2\} \). The network \( d_{1N} \) has a single connected component so we conclude that \( \text{deg}(b_2) \geq 3 \). Indeed, this node must connect the triangular to the rest of the network. The converse statement is obvious.
from the above discussion. By subsequently performing a recursive collapse of degree
one nodes, we may redefine \( d1N \) to consist of nodes at least degree two.

Many structures reduce to a sparsely connected triangle by recursively replacing
nodes with edges — this includes but is not limited to any simple polygon of nodes
\( P \subset d1N \) for which every node but one in \( P \) is of degree two. Any configuration
of nodes that can be reduced to a sparsely connected triangle can thus be collapsed
entirely: the sparsely connected triangle is broken by our routine as in Lemma 1, and
the remaining nodes are mapped into a terminal root node by recursive degree one
reduction. Our analysis leads to Algorithm 2 and Algorithm 3. We describe the data
structures used in these routines in Appendix A and define the following notation.

**Definition 4.** The data structure \( d2H \) is a hashable map \( \{\text{"field" : "data"}\} \),
where "data" is an ordered list. The mapping which takes the node \( b_0 \) to the edge
\( \{b_1, b_2\} \) is associated to the field \( e_{b_1,b_2} \), where we assume \( b_1 < b_2 \). We define any
subset \( gt \subset d1N \) to be a **generalized tree** if it is collapsed to a root node under
Algorithm 2 and Algorithm 3. The mapping which collapses a generalized tree to the
terminal node \( b_0 \) is associated to the field \( t_{b_0} \in d2H \).

**Algorithm 2** Degree two reduction

Define: \( d2N := d1N, d2E := d1E \) and \( d2H := d1H \).

\[ \text{while } \exists b_0 \in d2N \text{ with } \deg(b_0) < 3, \text{ do} \]

\[ \text{Let } b_0 \text{ be connected to } b_1 \text{ and } b_2 \text{ S.T. } b_1 < b_2. \]

\[ \text{Remove } b_0 \text{ from } d2N \text{ and lines } \{b_0, b_1\},\{b_0, b_2\} \text{ from } d2E. \]

\[ \text{if } \{b_1, b_2\} \notin d2E, \text{ then} \]

\[ \text{Write } \{b_1, b_2\} \in d2E. \]

\[ \text{end if} \]

\[ \text{if } \exists b_j \text{ S.T. } e_{b_0,b_j} \text{ or } e_{b_j,b_0} \in d2H, \text{ then} \]

\[ \text{Append list entries in } e_{b_0,b_j} \text{ or } e_{b_j,b_0} \text{ to } e_{b_1,b_2} \in d2H. \]

\[ \text{Remove } e_{b_0,b_j} \text{ or } e_{b_j,b_0} \text{ from } d2H \text{ and append } [b_1, b_0, b_2] \text{ to } e_{b_1,b_2} \in d2H. \]

\[ \text{else} \]

\[ \text{Append array } [b_1, b_0, b_2] \text{ to a list } e_{b_1,b_2} \in d2E. \]

\[ \text{end if} \]

\[ \text{if } t_{b_0} \in d2H, \text{ then} \]

\[ \text{Append the hashable map } \{t_{b_0} : d2H(t_{b_0})\} \text{ to the list } e_{b_1,b_2} \in d2H. \]

\[ \text{Remove } t_{b_0} \text{ from } d2H. \]

\[ \text{end if} \]

\[ \text{Pass } d2N, d2E \text{ and } d2H \text{ to Algorithm 3.} \]

\[ \text{end while} \]

\[ \text{return } d2N, d2E, d2H \]

Algorithm 2 maps nodes to edges and tracks these reductions sequentially in
the hashable map \( d2H \). Whenever \( b_0 \) is mapped to the edge \( \{b_1, b_2\} \), if \( \{b_0, b_1\} \) or
\( \{b_0, b_2\} \in d2H \), we write all preceding mappings to the list \( e_{b_1,b_2} \) when \( \{b_0, b_1\} \)
and \( \{b_0, b_2\} \) are removed. We enforce a similar condition whenever a generalized tree
\( t_{b_0} \in d2H \): these maps are appended, as a hashable map, to the list \( e_{b_1,b_2} \). The
subroutine, Algorithm 3, is a modification of Algorithm 1 which tracks the collapse
of sparsely connected triangles. Knowing that a degree one node is produced under
Algorithm 2 if and only if the routine breaks a sparsely connected triangle, Algorithm 3
stores the list \( e_{b_1,b_2} \) under in the root of the generalized tree subsequently collapsed.
Algorithm 3 Reduce sparsely connected triangle

if ∃ a₀ ∈ d2N with deg(a₀) < 2, then
    while ∃ a₀ ∈ d2N with deg(a₀) < 2, do
        Remove a₀ from d2N and line {a₀, a₁} from d2E.
        if t.a₀ ∈ d2H, then
            Append a₁ to the end of each array in t.a₀ ∈ d2H.
            Append all arrays in t.a₀ ∈ d2H to the list t.a₁ ∈ d2H.
            Remove t.a₀ from d2H.
        else
            Write array [a₀, a₁] to a list t.a₁ ∈ d2H.
        end if
    end while
    Prepend the hash table {e.b₁.b₂ : d2H(e.b₁.b₂)} to the list t.a₁ ∈ d2H.
    Remove e.b₁.b₂ from d2H.
end if
return d2N, d2E, d2H

The root is defined by the final iteration of the degree one reduction. The design and inversion of these data structures is described in detail in Appendix A.

Algorithm 2 and Algorithm 3 reduce the sets d1N, d1E to the sets d2N, d2E with 9716 nodes and 18,700 edges. Figure 4 summarizes this reduction with the histogram of the number of nodes per reduction in d2H and the distribution of the degrees of nodes in d2N. The mean degree of nodes in d2N is 3.85, with a standard deviation 1.76 and a maximal node degree of 38. The total number of collapsed edges in d2H is 9,696, with the mean number of nodes per edge is 3.88, standard deviation 4.24 and max nodes per edge 94. The total number of generalized trees collapsed in d2H is 2,579 with a mean of 4.38 nodes per generalize tree, standard deviation of 3.65 and max nodes per generalized tree 56. We note, generalized trees which have been mapped to edges are considered only as nodes within the meta-edge of their final reduction. Likewise, we do not distinguish meta-edges which have been collapsed into generalized trees from the root super node where their reduction terminates.

2.4. Triangular reductions. The network d2N, d2E is of a scale which permits qualitative analysis. However, further reductions may be necessary to run on-line parameter estimation in an operational window. It is possible to collapse higher degree coherent structures, such as non-sparse triangular configurations, but there is greater subtlety. The degree one and degree two node reductions produce an unambiguous model for net power flow. This is likewise the case for “pure” triangular reductions pictured in Figure 5 where there are three nodes, each of degree three and similar nominal voltage, forming a link between three large connected groups of nodes. In this case we may collapse the three nodes to a single super node of degree three that accounts for the net power flow through the triangle into the other sub-networks. “Pure” triangular configurations are easy to model, but are rare and many other triangular configurations exist throughout the network.

Recursively collapsing generic triangular configurations to super nodes may produce multiple lines between nodes, degree two nodes, degree one nodes and non-unique final reductions, which may strongly bias the distribution of node degrees. To prevent this bias, we permit the collapse of a triangle only if each node in the configuration doesn’t exceed a specified degree. Recursively mapping triangles to nodes, the de-
degrees of nodes will increase and decrease, so that we always refer to the degree of each node in the current iteration of the algorithm. Without this degree threshold, the test network collapses to a single densely connected node with an orbit of small components. Therefore, this threshold is a necessary criteria to produce a reduced model, via recursively collapsing triangles, that faithfully represents the dynamics.

The degree threshold introduces a tuning factor into our algorithm with which we balance the scale of the collapse with preserving the degree distribution for nodes in $d2N$. For any degree threshold $d\text{Thr}$, the maximal degree of a super node produced by collapsing a triangular configuration is given by $3(d\text{Thr} - 1)$. This corresponds to a configuration where all nodes only have common lines within the triangle. Assume each node has the maximum of $d\text{Thr}$ lines and one node gains all lines of the other two nodes. Mapping one node at a time, the first node adds only $d\text{Thr} - 1$ distinct lines to the super node and the second contributes $d\text{Thr} - 2$ distinct lines. We choose $d\text{Thr} = 6, 7$ and $8$, which produce a node of at most degree $15, 18$ and $21$ respectively.

A solely graph based reduction of triangles may, however, combine transmission and distribution nodes in a way which distorts the dynamics of net power flow. For instance, if the “pure” triangle in Figure 5 is formed by two nodes of high and one node of low nominal voltage, a super node produced from this triangle may confer stronger coupling between the three separate sub-networks (groups one, two, and three) than

---

**Fig. 4.** Upper: Histogram of nodes per tree in $d1H$, generalized tree in $d2H$ and meta-edge in $d2H$. Lower: Distribution of the degrees of nodes in $N$, $d1N$ and $d2N$.

**Fig. 5.** The triangle is collapsed to $b_0$, which gains all lines from $b_1$ and $b_2$ avoiding self lines.
actually exists. To prevent non-coherent mixing of transmission and distribution sub-networks, we restrict our reductions only to the nodes in $d2N$ which fall below an additional voltage threshold. We permit a reduction to a triangle if every node in the configuration additionally falls below a specified nominal voltage. We choose voltage thresholds of 110, 138, 230, 345 nominal KV (standard low, medium and high voltages for different transmission grid lines), and for reference, compare results without a voltage threshold. The nodes in $d2N$, and edges in $d2E$, may represent multiple nodes due to reductions performed in Algorithm 1, Algorithm 2 and Algorithm 3. Our analysis leads to Algorithm 4, we introduce the following notation.

**Definition 5.** Let $vThr$ be a specified voltage threshold. Define $nL$ to be a list of nodes in $d2N$ excluding any node(s)

- $b_0$ such that $t.b_0 \in d2H$ contains a node of nominal voltage above $vThr$,
- $b_0, b_1$ such that $e.b_0, b_1 \in d2H$ contains a node of nominal voltage above $vThr$
- or $b_0 \in d2N$ which has a nominal voltage above $vThr$.

The data structure $triH$ is a hashable map \{“field” : “data”\} where “data” is an ordered list. Entries of these lists are hashable maps of the form \{“$b_0$” : lines($b_0$)\} where lines($b_0$) is a list of lines associated to $b_0$ in $d2E$.

**Algorithm 4** Greedy triangular reduction

Define: $triN := d2N$, $triE := d2E$, $triH := \text{empty hashable map.}$

$nL := \text{random permutation of } nL$.

$dThr := \text{degree threshold}$, $K := 0$, $STOP := \text{length}(nL)$.

while $K < \text{STOP}$, do

$K := K + 1$, $b_0 := nL(K)$.

if $\text{deg}(b_0) < dThr$, then

while $\exists$ a triangular configuration defined by \{$b_0, b_1$\}, \{$b_0, b_2$\}, \{$b_1, b_2$\} $\in triE$

where $\text{deg}(b_1), \text{deg}(b_2) < dThr$ and $b_1, b_2 \in nL$, do

for $b_i \in \{$$b_1, b_2$$\}$, do

Append \{“$b_i$” : lines($b_i$)\} to $tri.b_0 \in triH$.

Append all entries in $tri.b_1$ to $tri.b_0 \in triH$.

Remove $tri.b_i$ from $triH$.

for each $b_j$ such that \{$b_i, b_j$\} $\in triE$, do

Write \{$b_0, b_j$\} to $triE$ excluding double and self lines.

Remove \{$b_i, b_j$\} from $triE$.

end for

Remove $b_i$ from $triN$ and from $nL$.

end for

$K := 0$, $nL := \text{random permutation of } nL$, $STOP := \text{length}(nL)$.

end while

end if

end while

for $tri.b_0 \in triH$, do

Append \{“$b_0$” : lines($b_0$)\} to $tri.b_0 \in triH$.

end for

return $triN, triE, triH$

In each iteration of Algorithm 4, we perform a greedy search for permissible triangles connected to a base node $b_0$, i.e. all triangles for which the nodes fall below the specified voltage and degree thresholds. We recursively collapse all such
triangles into $b_0$ by removing the two associated nodes from $\text{triN}$ and connecting all their lines to $b_0$, avoiding double and self lines. We perform this search until there are no permitted triangles which include $b_0$ and start the search again from a new base node. The base node from which we search for triangles is randomized upon each iteration. Thus, for each combination of voltage and degree threshold, we run an ensemble of experiments to find a distribution for our results. We plot the distribution of the degrees of nodes in $\text{triN}$ over 1,000 experiments in Figure 6; for reference we include the degree distribution of nodes in $\text{d2N}$. Note, while the triangular reduction produces nodes of degree at most 21, the reduction may lower the degree of any node if it is connected to at least two nodes in a permissible reduction. In Figure 6, the newly apparent nodes of degree greater than 21 correspond to this phenomena. We likewise plot the distribution of size of the network $\text{triN}$ in Figure 7 with respect to the various threshold settings, demonstrating both the variability of recursive triangular reduction and its sensitivity to the voltage and degree thresholds.

The smallest network produced by Algorithm 4 has 5,560 nodes and 11,079 edges — this is used as a reference for the limit of the reduction, performed without a voltage threshold. The degree distributions are similar across the degree thresholds in each voltage setting. However, we notice sensitivity in the size of the reduced network to the voltage threshold as we pass both from 110 KV to 138 KV, and from 138 KV to 230 KV thresholds respectively. Distributions of the network size are all close and strongly peaked for the 110 KV threshold, indicating that few nodes in the distribution sub-network remain un-clustered after the degree one and degree two steps. The dramatic reductions to network size passing to the 138 KV threshold indicates that the nodes of the distribution sub-network, and the substations connecting these to the transmission network (including super nodes which combine the two), possesses a loopy configuration that can be clustered by the triangular reduction for a significant gain. This is likewise the case passing to the 230 KV threshold, where the loopy structure below the high voltage transmission network can be reduced significantly. The distributions of network size for voltage thresholds above 230 KV are more closely aligned, and are instead distinguished along their degree thresholds.

![Figure 6](image-url)
3. Visualization of the reduced network. The sequential steps in section 2 reduce our test network to a size that is amenable to qualitative analysis. In this section, we discuss methods of graph visualization for the models produced by the degree two reductions and the triangular reductions. Following Wong et. al. [34], we choose to visualize the network based on its graph characteristics as in the GreenGrid visualization package. Rather than visualizing our network by the geographic information, a graph theoretic layout can better represent electrical distance and grid vulnerabilities. The classical force directed layout technique uses a spring and repulsion model where each node is a repelling body and the edges are represented by springs [6, 23]. Initial node positions are chosen randomly and the n-body problem is solved until the positions of nodes stabilize. This pseudo physical model can be utilized to represent electric grid physics by parameterizing spring lengths and node repulsion with the electrical distance of lines and the nominal voltages of nodes [34].

We utilize the JavaScript library vis.js [2] to perform interactive visualizations. The default graph layout uses uniform spring lengths and repulsion parameters, and implementing a parametrization scheme that reflects the electrical distance in the reduced network models is the subject of future work. The underlying ForceAtlas2 model [20] in vis.js is used to resolve the spring repulsion evolution. We produce a conceptual visualization of the clustering performed via Algorithm 4 as follows: (i) first generate node positions for \( \text{triN} \), using no voltage threshold and setting \( \text{dThr} = 8 \), and resolve the ForceAtlas2 model until node positions stabilize; (ii) fix these node locations and assign the initial position for every node in \( \text{d2N} \) as its clustered position in \( \text{triN} \); resolve the ForceAtlas2 model until node positions of \( \text{d2N} \) stabilize. Figure 8 demonstrates a realization of this de-clustering: the left hand plot shows the initial positions for the nodes in \( \text{d2N} \); the middle plot describes an intermediate point in their evolution as node positions are released; the right hand plot visualizes the stabilized \( \text{d2N} \) positions. This de-clustering visualization demonstrates how the degree threshold maintains qualitative graph features during the reduction.

4. Conclusions. Analysis of our test network demonstrates that our reductions meet the goals stated in section 2. Firstly, our graph based approach to network re-
STRUCTURE- & PHYSICS-PRESERVING REDUCTIONS OF POWER GRID MODELS

Fig. 8. Left: ForceAtlas2 initialized for $d_{2N}, d_{2E}$ with the clustered positions in $triN, triE$. Middle: the node positions are propagated by the ForceAtlas2 model. Right: positions stabilize.

| $|N| = 53,155$ | $|d_{1N}| = 32,891$ | $|d_{2N}| = 9,716$ | $|triN| = 5,560$ |
| $|E| = 63,832$ | $|d_{1E}| = 43,568$ | $|d_{2E}| = 18,700$ | $|triE| = 11,079$ |

Fig. 9. Test network reduction summary, $triN, triE$ correspond to the smallest realization.

These graph based reductions can also be used as a preliminary step for the classical Kron reduction on multiscale, transmission and distribution networks. Collapsing nodes and edges as above provides a nodal clustering with physically meaningful coupling between the distribution and transmission sub-networks, while reducing the large size of the distribution sub-network. Classical approaches to solving the optimal power flow and control problems can thereby be applied in a multiscale, power grid model which includes significant generation in the distribution sub-network: the algorithms in section 2 reduce the large multiscale network to one which can be understood in terms of its dynamically relevant super-nodes, used as reference in the subsequent Kron reduction, solving the issue of node selection.

While the reduced network is of a scale that allows additional qualitative analysis, e.g. visualizations and Kron reduction, the graph reduction itself provides insight. Computing the number of generalized trees and meta-edges, along with their associated sizes, helps to identify coherent structures within the full network. Likewise, collapsing triangles within the various voltage and degree thresholds exposes the connectivity of the network across its layers of transmission and distribution. The dramatic reductions when raising the voltage threshold from 110 KV to 138 KV, and likewise 138 KV to 230 KV, quantitatively demonstrates the highly meshed structure of the sub-network falling below these nominal voltages.
Each leaf in the tree $t_{b_0}$ can be re-introduced by its shortest path to $b_0$, described in equation (A.1). The terminal node $b_0$ has at least two lines connecting it to the remaining network.

The ultimate goal of this project is to produce a dynamically faithful, reduced order model, but this still requires significant steps that go beyond the scope of the current work. One component is the parametrization of the reduced order model, from both static and time-varying network data. Additional opportunities also arise from post-processing the reduced model with the Kron reduction. The interactive, graph-based visualizations may also be refined to represent electrical distance with spring and repulsion parameters. These steps will be explored in future research.

Appendix A. Data structures and inverting reductions. Allowing users to refine the reduced network structure is basic to our algorithm design. We expand in detail the data storage of generalized trees, edges and collapsed triangles. The recursion in Algorithm 2 and Algorithm 3 implies that edge and generalized tree data structures can be multilayered, containing multiple levels of sub-edges or sub-trees. Proceeding from the bottom layer to the top, and from right to left within lists, one can recover the reverse sequence of mappings to reconstruct a node. An example interactive visualization is available in web browsers [15], demonstrating the de-clustering performed in Figure 8. We likewise release our reduction scripts and toy data describing the full and reduced network node and edge sets, with voltage information in an arbitrary, per unit representation [16].

A.1. Tree data. Tree reductions are called by a field $t_{b_0}$ where $b_0$ is the terminal node of the collapse in Algorithm 1. Each field returns a list of arrays, each array corresponding to a branch collapsed to the root node $b_0$. The first position of each array describes the end leaf of the branch and each subsequent position describes the shortest path in the network to the terminal node. Figure 10 corresponds to the list

\[
\text{d1H}(t_{b_0}) = \{ [b_1, b_0], [b_2, b_0], [b_5, b_4, b_3, b_0], [b_6, b_4, b_3, b_0] \},
\]

where leaves are reintroduced by following the path described in the array.

A.2. Edge data. Let $b_1$ be a node of degree two, and suppose it is connected to $b_0$ and $b_2$. The basic mapping produced by Algorithm 2 takes $b_1$ to the edge $\{b_0, b_2\}$. We represent this map by the array $[b_0, b_1, b_2]$ where, without loss of generality, we assume that $b_0 < b_2$. Given such a sequence of mappings

\[
\text{d2H}(e_{b_0, b_2}) = \{ [b_0, b_1, b_2], [b_0, b_2, b_3], [b_0, b_3, b_4] \}
\]

we may reconstruct the original string by following the mappings from right to left in the list. Equation (A.2) describes the line of nodes in Figure 2. If $b_3$ is the terminal
node for a generalized tree, Algorithm 2 stores the associated generalized tree data in the list at the point of the reduction. The list is thus given

\[
\mathbf{d}_2^H(e_{b_0 b_4}) = \{[b_0, b_1, b_2], [b_0, b_2, b_3], [b_0, b_3, b_4], \{“t_{b_3}”: “tree data ”\}\}
\]

Generalized trees embedded in an edge are reconstructed by reintroducing the terminal node from the edge data and reconstructing the generalized tree as described in Appendix A.2.1. An edge in \(\mathbf{d}_2^H\) may contain an arbitrary length sequence of edge and tree reductions, possibly multilayered. Each meta-edge may therefore be represented in multiple ways by different orders of mappings, but each map can be inverted sequentially to reconstruct the original network, regardless of the order.

A.2.1. Generalized tree maps. Lemma 1 demonstrates that a sparsely connected triangle is collapsed if and only if Algorithm 2 produces a degree one node. Generalized tree data, therefore, includes the sequence of nodes mapped to edges which precipitate collapse of the triangle. The field \(t_{b_n}\) corresponds to the node, \(b_n\), that the generalized tree has been collapsed to. Suppose as in Figure 3, mapping \(b_0\) to the edge \([b_1, b_2]\) produces a degree one node in \(\mathbf{d}_2^N\). Algorithm 3 collapses degree one nodes recursively until every node is again at least degree two. Let \(b_n\) be the terminal node of this collapse, then Algorithm 3 stores a hashable map as the first entry of \(t_{b_{n}}\in\mathbf{d}_2^H\), followed by the array with the path from \(b_1\) to the terminal node

\[
\mathbf{d}_2^H (t_{b_n}) = \{\{“e_{b_1 b_2}”: [b_1, b_0, b_2]\}, [b_1, b_2, \cdots, b_n]\}\}
\]

In general, the edge data precipitating the collapse of the sparsely connected triangle can be of arbitrary length and contain multiple layers.

A.3. Triangular reductions. Due to the more arbitrary nature of Algorithm 4, we take a simple approach to track the reductions. The field \(\mathbf{tri}_{b_0}\in\mathbf{tri}_H\) corresponds to a list where each entry is a hashable map of the form \(\{“b_j”: \mathbf{lines}(b_j)\}\). The value \(\mathbf{lines}(b_j)\) is the list of lines associated to \(b_j\) in \(\mathbf{d}_2^E\). In this way, one can reintroduce a node from a collapsed triangular configuration by writing the node \(b_j\) into \(\mathbf{tri}_N\) and reconnecting this node with the appropriate edges from \(\mathbf{d}_2^E\), while removing these edges from \(b_0\) if the lines were formed uniquely by joining \(b_j\) to the cluster.

REFERENCES

[1] S. Aggarwal, *Greasing the electric grid, the world’s largest machine*, November 2014, http://www.livescience.com/48893-improving-efficiency-on-the-electric-grid.html. Accessed: 2017-04-06.
[2] Almende B.V., *vis.js*, January 2017, http://visjs.org. Copyright (C) 2010-2016 Almende B.V. License: Apache 2.0.
[3] S. Backhaus and M. Chertkov, *Getting a grip on the electrical grid*, Physics Today, 66 (2013), pp. 42–48.
[4] R. Bent, D. Bienstock, and M. Chertkov, *Synchronization-aware and algorithm-efficient chance constrained optimal power flow*, in Bulk Power System Dynamics and Control-IX Optimization, Security and Control of the Emerging Power Grid (IREP), 2013 IREP Symposium, IEEE, 2013, pp. 1–11.
[5] D. Bienstock, M. Chertkov, and S. Harnett, *Chance-constrained optimal power flow: Risk-aware network control under uncertainty*, SIAM Review, 56 (2014), pp. 461–495.
[6] U. Brandes, *Drawing on physical analogies*, in Drawing Graphs: Methods and Models, M. Kaufmann and D. Wagner, eds., Lecture Notes in Computer Science, Springer, 2001.
[7] T. Chen and C. Chen, *Efficient large-scale power grid analysis based on preconditioned krylov-subspace iterative methods*, in Design Automation Conference, 2001. Proceedings, IEEE, 2001, pp. 559–562.
[8] D. DEKA, S. BACKHAUS, AND M. CHERTKOV, Estimating distribution grid topologies: A graphical learning based approach, in Power Systems Computation Conference (PSCC), 2016, IEEE, 2016, pp. 1–7.
[9] D. DEKA, M. CHERTKOV, and S. BACKHAUS, Structure learning in power distribution networks, IEEE Transactions on Control of Network Systems, (2017).
[10] D. DEKA, S. VISHWANATH, and R. BALDICK, Analytical models for power networks: The case of the western us and ercot grids, IEEE Transactions on Smart Grid, (2016).
[11] F. DORFLER and F. BULLO, Kron reduction of graphs with applications to electrical networks, IEEE Transactions on Circuits and Systems I: Regular Papers, 60 (2013), pp. 150–163.
[12] K. DVJIOTHAM, S. BACKHAUS, and M. CHERTKOV, Distributed control of generation in a transmission grid with a high penetration of renewables, in Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on, IEEE, 2012, pp. 635–640.
[13] K. DVJIOTHAM, M. CHERTKOV, and S. BACKHAUS, Storage sizing and placement through operational and uncertainty-aware simulations, in System Sciences (HICSS), 2014 47th Hawaii International Conference on, IEEE, 2014, pp. 2408–2416.
[14] H. FARRANGI, The path of the smart grid, IEEE power and energy magazine, 8 (2010).
[15] C. GRUDZIEN, Colin grudzien: Research, https://cgrudz.github.io/research/. Accessed: 2017-04-06.
[16] C. GRUDZIEN, cygruds/electric_grid_model_reduction: First release of reduction scripts, June 2017, doi:10.5281/zenodo.815824, https://doi.org/10.5281/zenodo.815824.
[17] P. HINES, S. BLUMSACK, E. COTILLA SANCHEZ, and C. BARROWS, The topological and electrical structure of power grids, in System Sciences (HICSS), 2010 43rd Hawaii International Conference on, IEEE, 2010, pp. 1–10.
[18] R. HOFFMAN, Practical state estimation for electric distribution networks, in Power Systems Conference and Exposition, 2006. PSCE'06. 2006 IEEE PES, IEEE, 2006, pp. 510–517.
[19] M. HUNEAULT and F. GALLANA, A survey of the optimal power flow literature, IEEE transactions on Power Systems, 6 (1991), pp. 762–770.
[20] M. JACOMY, T. VENTURINI, S. HEYMANN, and M. BASTIAN, Forceatlas2, a continuous graph layout algorithm for handy network visualisation designed for the gephi software, PLOS ONE, 9 (2014), pp. 1–12.
[21] R. KANNAN, S. VEMPALA, and A. VETTA, On clusterings: Good, bad and spectral, Journal of the ACM (JACM), 51 (2004), pp. 497–515.
[22] J. KATZ, Iot for electric power: Smart grid was the beginning, http://iot.ieee.org/newsletter/january-2017/iot-for-electric-power-smart-grid-was-the-beginning.html. Accessed: 2017-04-06.
[23] S. KOBOROY, Force-directed drawing algorithms, in Handbook of Graph Drawing and Visualization, R. Tamassia, ed., Discrete Mathematics and Its Applications, CRC Press, 2013.
[24] M. KRANING, E. CHU, J. LAVAEI, S. BOYD, ET AL., Dynamic network energy management via proximal message passing, Foundations and Trends® in Optimization, 1 (2014), pp. 73–126.
[25] J. MEDINA, N. MULLER, and I. ROYTELMAN, Demand response and distribution grid operations: Opportunities and challenges, IEEE Transactions on Smart Grid, 1 (2010), pp. 193–198.
[26] M. NEWMAN, Fast algorithm for detecting community structure in networks, Physical review E, 69 (2004), p. 066133.
[27] T. NISHIKAWA and A. MOTTER, Comparative analysis of existing models for power-grid synchronization, New Journal of Physics, 17 (2015), p. 015012.
[28] J. PAN, R. NUQUI, K. SRIVASTAVA, T. JONSSON, P. HOLMBERG, and Y.-J. HAFNER, Ac grid with embedded vsc-hvdc for secure and efficient power delivery, in Energy 2030 Conference, 2008. ENERGY 2008. IEEE, IEEE, 2008, pp. 1–6.
[29] A. PHADKE, Synchronized phasor measurements—a historical overview, in Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES, vol. 1, IEEE, 2002, pp. 476–479.
[30] SMITHSONIAN MAGAZINE, The largest machine ever built, http://www.smithsonianmag.com/videos/category/history/the-largest-machine-ever-built/. Accessed: 2017-04-06.
[31] J. VON APPEN, M. BRAUN, T. STETZ, K. DIWOLD, and D. GEIBEL, Time in the sun: the challenge of high pv penetration in the german electric grid, IEEE Power and Energy magazine, 11 (2013), pp. 55–64.
[32] J. WANG, Deterministic random walk preconditioning for power grid analysis, in Computer-Aided Design (ICCAD), 2012 IEEE/ACM International Conference on, IEEE, 2012, pp. 392–398.
[33] WIKIPEDIA, Hash table — Wikipedia, the free encyclopedia, 2017, https://en.wikipedia.org/wiki/Hash_table. [Online; accessed 2017-6-8].
[34] P. C. Wong, K. Schneider, P. Mackey, H. Foote, G. Chin Jr, R. Gutthomson, and J. Thomas, *A novel visualization technique for electric power grid analytics*, IEEE Transactions on Visualization and Computer Graphics, 15 (2009), pp. 410–423.

[35] X. Zhao, J. Wang, Z. Feng, and S. Hu, *Power grid analysis with hierarchical support graphs*, in Computer-Aided Design (ICCAD), 2011 IEEE/ACM International Conference on, IEEE, 2011, pp. 543–547.