Tree-Partitioning as an Emergency Measure to Contain Cascading Line Failures

SCHOLARONE™
Manuscripts

Tree-Partitioning as an Emergency Measure to Contain Cascading Line Failures

Janusz W. Bialek, *Fellow, IEEE* , Vahid Vahidinasab, *Senior Member, IEEE*

Abstract—**This paper proposes to replace controlled islanding, which is a defense mechanism against cascading failures, by tree partitioning whereby some of the tie-lines connecting the clusters are still connected in such a way that the cluster-level graph forms a tree. Tree-partitioning prevents line failures from spreading between clusters, similarly as for islanding, but keeps the clusters connected. That results in three main advantages. Power transfers between the clusters can still take place, helping to balance each cluster and limiting any necessary load shedding. Fewer lines are cut, which reduces the shock to the system. There is no need to re-synchronize the clusters after the emergency. This paper offers a simple graph-theoretic justification for tree-partitioning, rather than one based on the spectral analysis of network Laplacian proposed in the literature. It also proposes a two-stage methodology, which utilizes spectral clustering, for splitting a network into tree-connected clusters. Test results performed on the 118 node IEEE test network have confirmed the usefulness of the methodology.**

Index Terms—**power systems, blackout prevention, clustering**

I. INTRODUCTION

Large-area power blackouts, when millions of customers lose supply, are very rare but with very significant social and economic consequences. Many different reasons could cause such blackouts but one of the most common modes is a cascading blackout when tripping one transmission line results in a domino-like spreading of consequent line trips. Among the many examples, one can mention the blackouts in Canada/US in 2003, Italy in 2003 and Europe in 2006 [1].

One particular defense mechanism against cascading failures is *Controlled* Islanding (CI). When a large-area blackout is imminent, CI splits the network into a number of self-sustained islands to prevent the proliferation of outages. A good overview of different techniques can be found in [2]. The clusters of nodes identified to form islands should have minimal power imbalance [3][4], CI should cause a minimal change in the power flow pattern [5][6], minimize dynamic stability problems [2], [7] - [14] and consider resynchronization requirements [15]. The conflicting requirements often require trade-offs [12]. The question of *when* to island is still open but some solutions have been proposed [16][17].

Despite many publications on CI over the last two decades, there has been little reported take-up by the industry. The main

reason seems to be the drastic nature of CI. Firstly, islanding goes against the fundamental instincts of System Operators who always try to keep the system together. Secondly, islands would inevitably have power imbalances and would therefore require balancing actions such as generation/load shedding resulting in a potentially significant additional loss of supply. Thirdly, CI would require resynchronization of islands which may not be easy and may take some time. Fourthly, CI requires opening many transmission lines and dealing with the resulting, potentially severe, static and dynamic problems. To summarize, there is a fear that the medicine may be worse than the disease. In a quest to find a less drastic way to contain a cascading blackout, we have utilized insights provided in [18][19] which analyzed spectral properties of the *tree-partitioned network*, i.e. a network divided into regions (clusters) connected in such a way that the cluster-level graph forms a tree. Papers [18][19] proved that Tree-Partitioning (TP) *localizes* line failures so that failures in one cluster do not affect power flows in other clusters. Therefore, this paper's main idea is to replace CI by TP as an emergency measure to contain cascading line failures so that they do not propagate to other clusters.

Essentially, the main difference between CI and TP is that for the former, all the tie-lines connecting the clusters are cut so the clusters are islanded. For TP, some of the tie-lines, referred to as *bridges*, remain in operation so the clusters are still connected. Both achieve localization of line failures but TP has the following advantages stemming from keeping the clusters connected. Firstly, power transfers between clusters can still take place and therefore there is no need for power balancing actions such as generation/load shedding. Secondly, fewer tie-lines are cut so the shock to the system is less and therefore system stability and congestion are less affected. And thirdly, there is no need for resynchronization when returning to normal operation.

This paper makes the following original contributions. Firstly, in section II. , we offer a *simple graph-theoretical proof* of the line failure localization property of TP, rather than a more complicated proof utilizing spectral properties of the network graph [18][19]. We also show that failure localization due to *cut vertices* is a special case of TP. Secondly, in section III. , we propose *tree partitioning as an emergency measure to replace controlled islanding*. Thirdly, we address a vital question, which has not been dealt with in [18][19], of *how to split a network into tree-connected clusters*. We suggest a two-stage approach. In the first step, section IV. , we suggest using

---
Janusz W. Bialek is with Newcastle University, UK, and with Skolkovo Institute of Science and Technology (Skoltech), Russia (Janusz.bialek@newcastle.ac.uk). Vahid Vahidinasab is with Nottingham Trent University, UK (vahid.vahidinasab@ntu.ac.uk).

spectral clustering [8][9][13][20] to select clusters of tightly connected nodes in both static and dynamic sense. Then, in section V. , we examine two approaches to select which of the tie-lines linking the clusters should be retained as bridges. Application of Prim's algorithm [23], *minimizes the overall impact* of cutting tie-lines while the *impact on congestion* can be minimized by evaluating all possible spanning trees of the cluster-level graph.

Finally, in section VII. , we show test results examining the performance of the proposed methodology on the 118 node IEEE test network.

In this paper, we concentrate on the analysis of TP using the static DC network model as the first step in our investigation. We discuss the effects of using the AC network model and the dynamic effects at a high level in section VI. Full examination of those issues would require an extensive further investigation which is beyond the scope of this paper.

II. GRAPH THEORY AND TREE-PARTITIONING

A. Graph representation of power networks

The power network can be naturally represented as a graph: vertices represent buses (nodes), and edges (links) represent electrical connections (lines or transformers). Without losing generality, we will only consider *simple graphs*, where no self loops (i.e. edges beginning and ending at the same vertex) and no multiple edges between the same vertices are allowed.

A graph is described by the vertex-edge *incidence matrix C*

$$C_{ie} = \begin{cases} 1 & \text{if node } i \text{ is the source vertex of edge } e \\ -1 & \text{if node } i \text{ is the target vertex of edge } e \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The topological structure of the graph expressed by (1) does not capture the functional information about the power grid, which could be provided using edge weights. For partitioning purposes, it makes sense to select the weight equal to the power flow in an edge $w_{ij} = |P_{ij}|$ where P_{ij} is the real power flow from node i to node j (assuming the DC network model). A small power flow indicates that disconnecting the line will result in a small shock to the system. In other words, edge weights can be interpreted as a penalty for cutting the corresponding line, but also as a measure of the connection strength, as nodes connected by lines with high power flows are more likely to be clustered together.

Laplacian of a graph is a square matrix $L = CWC^T$ where W is the diagonal matrix of edge weights. The elements of L are

$$L_{ij} = \begin{cases} \sum_{k=1}^{N} w_{ik}, & \text{if } i = j, \ k \neq i \\ -w_{ij}, & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

and the *degree* of vertex i is equal to $|L_{ii}|$.

When $W = B$, where B is the diagonal matrix of line susceptances, the Laplacian matrix has a clear power engineering interpretation as the nodal susceptance matrix.

Eigenvalues of the Laplacian matrix are nonnegative real numbers.

A *tree* is an acyclic graph. Disconnecting any edge in a tree separates the tree into two disjoint trees. A tree with N vertices will have $(N$-1) edges.

B. Graph-theoretical justification of failure localization in tree-partitioned networks

Consider Fig. 1(a) which shows a network consisting of a number of clusters denoted as *a, b, c, d, e, f*. The network is tree-partitioned when its cluster-level graph, Fig. 1(b), forms a tree. We will refer to the edges in the cluster-level tree as *bridges*. The comprehensive theory behind TP has been developed in [18][19] using spectral representation of the network graph and proved that internal line failures in a cluster could not affect power flows in other clusters. Hence, any cascading line failures in a cluster cannot propagate to other clusters, i.e. they are *localized*.
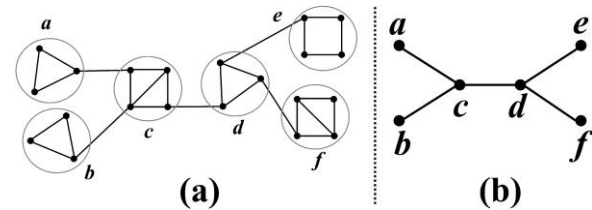

Fig. 1 A tree-partitioned network: (a) full graph with clusters encircled; (b) cluster-level graph.

This paper argues that the above conclusions can be deduced in a simpler way using fundamental algebraic graph theory. First, consider a simple tree consisting of two clusters connected by a bridge. The only way one cluster can affect power flows inside the other is through the bridge flow. The bridge flow depends only on each cluster's power imbalance as the net export from one cluster must be equal to the net import by the other one, assuming a lossless network. Hence, if both clusters' power imbalances are constant, any line trip inside one cluster cannot affect the bridge flow and therefore cannot affect power flows inside the other cluster.

The above reasoning can be generalized to any number of tree-connected clusters. To prove it in a formal way, let us consider a cluster-level graph of a tree- partitioned network, as in Fig. 1(b), with N vertices (clusters) and $(N$-1) edges (bridges). Kirchhoff's Current Law (KCL) applied to the graph can be expressed as

$$p' = C' f' \quad (3)$$

where C' is the $N \times (N-1)$ incidence matrix, p' is the vector of N cluster imbalances, and f' is the vector of $(N$-1) bridge flows. We assume the network to be balanced so the sum of all cluster imbalances is zero. As the graph is a tree, matrix C' has full column rank equal to $(N$-1). Hence, matrix $(C'^T C')$ is invertible and bridge flows f' are unique and equal to:

$$f' = (C'^T C')^{-1} C'^T p' \quad (4)$$

Equation (4) shows that bridge flows depend only on the tree topology and cluster imbalances but do not depend on each cluster's internal connections or any network impedances. Note that for practical applications, solving (3) does not require using pseudoinverse matrix $(C'^T C')^{-1}$. Bridge flows can be calculated in a recursive way using tree logic and starting from the leaf clusters of the tree – see an example in section VII. A.

Now power flows inside each cluster can be calculated by treating bridge flows f' as additional injections to the clusters at the bridges' terminal nodes. Using the DC network model, we get:

$$p = Cf \qquad (5)$$

$$f = BC^T\theta \qquad (6)$$

where f is the vector of line flows in the cluster (excluding the bridges), C is node-line incidence matrix of internal connections in the cluster (excluding the bridges), θ is the vector of nodal angles, B is the matrix of line susceptances in the cluster (excluding the bridges), and p is the extended vector of nodal injections in the cluster. Vector p includes, apart from the "normal" injections (i.e. nodal generations/demands), also additional injections at the terminal nodes of the bridges which are equal to the appropriate elements of the bridge flows f' calculated in (4). Solving (5)-(6) gives a unique solution for cluster power flows f and voltage angles θ (after assuming one angle as the reference).

Equations (5)-(6) show that the internal cluster power flows f are functions of the extended cluster injections p (that also include the bridge flows f'), cluster topology (incidence matrix C), and line susceptances B. On the other hand, (4) shows that bridge flows f' depend only on cluster power imbalances p' and the tree topology expressed by C'. Hence, *if both the tree topology and cluster power imbalances stay constant,* power flows inside each cluster depend only on its internal incidence and susceptance matrices. Therefore, line failures in a cluster affect only power flows in that particular cluster but not in any other cluster.

C. Localization of line failures due to cut vertices

Consider now *cut vertices* which, when removed, separate a graph into two or more disjoint graphs – see an example of a "butterfly" graph in Fig. 2(a). As proved in [19], cut vertices also localize line failures and we will now generalize our proof from section II. B. to prove the same.

Let us conduct a thought experiment splitting the cut vertex (bus) a in Fig. 2(a) into three auxiliary vertices (buses) a_0, a_1, a_2 connected by two auxiliary lines a_0-a_1, a_0-a_2, of zero impedance as in Fig. 2(b). We assume that the original injection at bus a is connected to bus a_0.

Hence we have transformed a network with a cut vertex in Fig. 2(a) into a tree-partitioned network in Fig. 2(b) in which the auxiliary lines a_1-a_0, a_2-a_0, become bridges linking three clusters (b, c, a_1), (a_0), and (d, e, a_2). And, as proved in section II. B. , a tree-partitioned network localizes line failures.
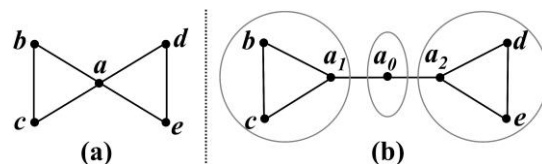

Fig. 2: (a) a graph with cut vertex a; (b) inserting zero-impedance lines creates a tree-partitioned network.

To summarize, we have proved in this section that tree-partitioned networks localize line failures and that cut vertices are just a special case of tree-partitioned networks. We have employed only fundamental algebraic graph theory without resorting to a complex spectral representation of the network graph as in [18][19].

D. Comments

The fundamental condition for the line failure localization property of TP is that the power imbalances in each cluster must stay constant and that bridges stay intact. However, practice shows that line failures may lead to generator trips hence changing power imbalances. On the other hand a bridge failure separates the network into two disjoint parts which, in effect, is equivalent to islanding and results in the two parts losing power balance. Similarly, line failures in a cluster may lead to islanding of a part of the cluster so that the network is separated into two disjoint parts. If the islanded part contains loads and generators that are not in power balance, the remaining part of the cluster, which is connected to the rest of the system, has a changed power balance which again violates the localization condition of TP. Any changes in the power balance trigger a frequency control process. Those effects cannot be studied using static DC or AC power flows as they require modelling of frequency response which is discussed in Section VI.

Next, let us discuss if it would make sense to leave more than one bridge between clusters. That would make the network topology more robust as a failure of one of the bridges would still keep the clusters connected. However note that while the cluster-level graph would still be radial, it would not be a tree as there would be a loop due to the parallel bridges. Hence, the failure localization properties of TP would generally no longer hold. To consider that question further, let us assume that there are two bridges linking two clusters and that the terminal nodes of the two bridges are electrically distant from each other. If there are internal line trips in one of the clusters, the total power flowing over the two bridges stays constant, as the cluster-level network is radial (although no longer a tree), but there may be a shift of power transfer from one bridge to the other due to the cluster's changed topology. This will change two elements of the extended vector of nodal injections p in (5) that correspond to the two bridge lines. A change in vector p will change power flows f in the other cluster which may potentially lead to overloads and cascading failures. This confirms that for failure localization, clusters must be connected by one bridge only. However, if the two terminal nodes of the bridges are electrically close, the shift in bridge power flows will be small. Hence, it will not cause a significant change in the other cluster's internal power flows so line failures will be in practice localized. Quantification of that issue would warrant a separate

III. TREE-PARTITIONING AS AN EMERGENCY MEASURE TO CONTAIN CASCADING FAILURES

In this section we suggest how Tree-Partitioning (TP) can replace Controlled Islanding (CI) as an emergency measure to contain cascading line failures. We do not address the question of *when* to apply it [16][17]. The high-level algorithm defense mechanism will work as follows:

1. Using the latest information about the system, identify clusters of closely-connected nodes in both static and dynamic sense - see section IV. Generally, each cluster will have a power imbalance.
2. Find a cluster-level spanning tree, i.e. identify which tie-lines connecting the clusters should be kept connected as bridges and which tie-lines should be disconnected– see section V.
3. Knowing the cluster-level tree topology and cluster imbalances, use (4) to calculate bridge flows.
4. Treating bridge flows as additional cluster injections at bridges' terminal nodes, calculate post-partitioned line flows inside each cluster using (5)-(6).
5. Assess the impact of tree-partitioning on congestion and stability and take appropriate remedial actions if necessary. Note that the strategies proposed in steps 1 and 2 should ensure that the need for remedial actions is minimized.

Let us now compare TP to CI. Both CI and TP localize line failures so that they cannot propagate outside their cluster but for TP, some of the tie-lines (bridges) remain in operation so the clusters are still connected. This results in the following advantages of TP. Firstly, CI usually creates power imbalances in each island and will therefore require balancing actions such as generation/load shedding. When the network is tree-partitioned, power transfers between the clusters can still take place to keep the clusters in balance thus avoiding generation/load shedding. Secondly, fewer lines are cut so that the shock to the system is less and therefore system stability and network congestion will be less affected. Thirdly, there is no need to re-synchronize clusters when returning to normal operation. Admittedly, re-connecting the disconnected tie-lines could cause problems due to possibly large angle differences between their terminal nodes, but it is less of a problem than resynchronization of islands.

One of the issues for CI is finding an optimal trade-off between minimizing generator non-coherency and minimizing power imbalance of the islands [12]. TP solves that problem as there is no need to worry about power balances of clusters.

IV. HOW TO SELECT CLUSTERS?

While [18][19] provided a theoretical background for TP, it did not address the vital issue of how to partition a network into clusters forming a tree. In this paper we suggest a two-stage approach. In this section, we present the first step, i.e. selection of optimal clusters, while in section V. we present a

methodology of choosing which tie-lines linking the clusters should be retained as bridges.

There are many methodologies available in the islanding literature, briefly reviewed in Introduction, about the selection of nodes and generators to form clusters. Generally, any methodology could be used for that purpose as long as cluster nodes and generators are tightly connected both in a static and dynamic sense and weakly connected to other clusters. However, we believe that spectral clustering provides perhaps the most suitable framework as it can be theoretically justified using *Cheeger inequality* as discussed below.

To measure the quality of clustering [20], we define *expansion* $\phi(S)$ of cluster S as

$$\phi(S) = \frac{\partial(S)}{vol(S)} \qquad (7)$$

where $\partial(S)$ is size of the cluster *boundary*, i.e. the sum of weights of the edges (tie-lines) linking S with other clusters, and $vol(S)$ is the cluster *volume* defined as the sum of degrees of its vertices. The best partition into k clusters is quantified by minimizing the maximum expansion among all the clusters. It means that we try to find clusters of nodes that are internally strongly connected with each other (high volume) with weak interconnections between them (low boundary). This has the effect of minimizing disruption to the system when tie-lines linking the clusters are cut. The problem is NP-hard and spectral clustering provides an approximate solution to the problem.

In a graph with N vertices, we use the first k eigenvectors of the Laplacian, for some $2 \le k << N$, to give geometric coordinates to the N vertices in $\Re^k$. Namely, these coordinates are the N rows of the $N \times k$ matrix whose columns are the k eigenvectors of the smallest k eigenvalues. The resulting data points are then clustered using a standard clustering algorithm. Spectral clustering can be justified by *Cheeger inequality* [20] which indicates that the smaller the k-th eigenvalue, the closer the approximate solution is to the optimal one.

Dendrogram produced by the *hierarchical spectral clustering* methodology [20] reveals the internal structure of a network and makes it possible to partition a network into almost any number of clusters, although with varying effectiveness. Best partitions are associated with large eigengaps, i.e. differences between two consecutive eigenvalues. Fig. 3 shows eigengaps of 118 node IEEE test network while Fig. 4 shows the network divided into 5 clusters.

Clustering procedure should take into account not only power flows but also system stability. To take into account transient stability, each cluster should include only coherent generators so that they do not lose synchronism following any disturbances. To achive that, spectral clustering can be executed in two steps [8][9]. In the first step, a graph of the Kron-reduced network is constructed such that it contains only generator nodes and its edge weights are the synchronizing power coefficients that describe dynamic coupling between the generators. To satisfy the generator coherency constraint, the generators are grouped using spectral clustering which will result in each clusters containing generators strongly linked to

each other (high synchronizing power coefficients) and with weak links (small synchronizing power coefficients) between the clusters. In the second step, constrained spectral clustering is performed on the full static DC network model to identify clusters of network nodes that are tightly internally connected and with minimal power flows between the clusters. To make sure that the clusters contain only coherent generators, pair-wise constraints obtained from the first step are enforced in the clustering process.

Spectral clustering can be further generalized to take voltage and frequency stability into account [13]. We will not discuss those issues in detail here and refer instead to the literature.

V. SELECTING THE OPTIMAL SPANNING TREE

Clustering will generally result in a cluster-level graph that is meshed, i.e. it contains loops. Let us now define the *quotient graph* consisting of N vertices corresponding to the clusters and L edges corresponding to tie-lines linking the clusters. Fig. 5 shows an example of the quotient graph of 118 node IEEE test network divided into 5 clusters with 17 tie-lines linking them. TP requires finding a spanning tree of the quotient graph. The spanning tree is defined as a subgraph that links every node in the graph without any cycles. The retained links are bridges and the remaining tie-lies linking the clusters have to be disconnected. An optimum spanning tree depends on the chosen criterion of optimality.

A. Maximum-weight spanning tree

The first criterion we consider is to minimize a shock to the system when tie-lines are cut. To do that we will utilize the weights (i.e. power flows) associated with the tie-lines, hence we will try to cut lightly-loaded lines and preserve the lines with heavy power flows. To achieve that we utilize the Prim's algorithm [23] that finds a minimum-weight spanning tree for a weighted undirected graph. As we want to choose such edges that maximize, rather than minimize, power flows in the retained lines (bridges), we have modified the Prim's algorithm so that it finds the maximum-weight spanning tree.

The maximum-weight algorithm may informally be described as follows:
1. Find an edge with the heaviest weight. The edge with its terminal vertices will form the starting tree.
2. Of the edges that connect the tree to vertices not yet in the tree, find the maximum-weight edge, and transfer it to the tree together with its terminal vertex.
3. Repeat step 2 until all vertices are in the tree.

Note that we do not recalculate the flows after each step as we are interested in the overall impact of TP by comparing the flows before and after TP. Hence each step is conceptual rather than physical. Section VII. A. shows an example of the algorithm using the quotient tree of 118 node IEEE test network divided into five clusters.

B. Minimizing load shedding to relieve congestion

Maximum-weight tree minimizes the overall shock to the system created by cutting less-loaded tie-lines as that should minimize the impact on congestion and stability. However,

congestion depends on individual line flows and limits rather than total changes in all flows. Hence, let us consider an alternative algorithm that selects such a spanning tree that the sum of line overloads caused by TP is minimized. This will therefore also minimize load shedding required to relieve congestion.

The brute-force approach to do it would be to simply check the overloads for every possible spanning tree and chose the one that minimizes the overloads. The first question is how many trees would we have to check. Kirchhoff's Matrix Tree Theorem says that the number of spanning trees can be computed in polynomial time and it is equal to any cofactor of the unweighted Laplacian matrix L. An example in section VII. B. shows that there are 420 possible spanning trees for the quotient graph shown in Fig. 5 so we would have to run 420 power flows.

In order to examine if it is possible to speed up the calculations, we have considered the application of Line Outage Distribution Factors (LODFs) [22] instead of running a power flow for each spanning tree. *LODF* is defined as the share of a disconnected line flow that will show up on another line. However, the problem is that LODFs are valid for a linear DC network model so the results are reasonably accurate only for small disturbances while TP requires many tie-lines to be disconnected. For the IEEE 118 node system considered, creating a spanning tree involves disconnecting 13 tie-lines and our calculations showed that applying LODFs was highly inaccurate and cannot be applied in practice. The optimal tree chosen using LODFs was supposed to result in the overload of 11 MW. We have checked that result using the AC power flow and found out that the actual overload value was 165 MW. This proves that using LODFs to assess the effect of TP on congestion is highly inaccurate.

Another possibility of speeding up calculations would be by eliminating irrelevant trees in the search algorithm but that issue woud require further research.

Once an optimal spanning tree has been identified, congestion relief actions have to be undertaken which might involve generation/load shedding that would limit power transfers between clusters. We do not deal in detail with that issue in this paper, as congestion relief tools are well-known, but note that the *Unified Controlled* proposed in [21] would automatically minimize load shedding in a tree-partitioned network.

VI. AC NETWORK MODELLING AND DYNAMICS

TP's localization of line failures has been proved in this paper using the static DC network model that neglected transmission losses, reactive power, and voltage changes. Also, the effect of generation failures, which often accompany line failures, has not been considered. A full investigation of those issues is beyond the scope of this paper but in this section we offer a high-level discussion.

A. The effects of using AC network model

When using the AC network model, we have to consider the effects of transmission losses, reactive powers and changing voltages. Firstly, let us consider the effects of considering

transmission losses. A line failure in a cluster will generally increase transmission losses in that cluster and a power flow program will cover those increased losses from the slack bus. Hence, if the slack bus is located in the cluster where the line failure occurred, the additional losses would be covered internally inside the cluster and there would be no effect on bridge flows. However if the slack bus is located in another cluster, the slack bus would have to export power over the bridges to the cluster where the line failed to cover the increased losses. This, of course, would change the bridge flows meaning that the localization feature of TP would not hold. However, the slack bus just is a mathematical trick to make sure that the non-linear power equations are balanced. In reality, if there is a deficit of power due to changed losses, that deficit will cause the frequency to drop which will activate the frequency control process. In the new steady-state, the allocation of increased transmission losses between individual generators will depend on the Automatic Generation Control (AGC) strategy. AGC could be simulated in a static AC power flow by assuming a *distributed slack bus*, with distribution factors corresponding to individual generators' participation factors in the AGC. The conclusion is, therefore, that line failures will in practice cause some changes in the bridge flows due to increased losses. Still, those changes are likely to be small and of no great practical significance as transmission losses tend to be small.

Next, let us consider the effect of reactive powers. Assuming the DC network model, we have used real power flows as edge weights and when considering congestion. Under the AC network model, we could use apparent power for those purposes. It is certainly justified when considering congestion as it is the current (proportional to the apparent power), rather than just the real power, that heats a line. It is more debatable whether apparent power should be considered as line weights. The modified Prim's algorithm minimizes interruption caused by the disconnection of tie-lines. One can argue that a disruption of real power flows is more important than a disruption of reactive power flows. That issue warrants a further investigation beyond the scope of this paper.

Next, let us consider the voltage effects. Any line failure will cause a reconfiguration of power flows and depressed voltage profiles. That latter effect is not only of secondary importance when compared to the former, but it is also mostly local and would therefore affect mainly the "sick" cluster where lines failed. Hence, localization of flow reconfiguration by TP would also localize voltage drops, especially when clusters have been chosen such that they are strongly internally connected with weak connections to other clusters – see section IV. If depressed voltages in the "sick" cluster threatened voltage stability in the whole system, under-voltage protection would open the connecting bridges, islanding the "sick" cluster.

B. Dynamic effects

First, let us consider the dynamic effects of line failures, concentrating on transient stability. We assume that the clusters are selected such that they contain only coherent generators, as discussed in section IV. Hence, generators in a tree-partitioned network should stay synchronized within each cluster following large disturbances, but the clusters can lose synchronism with each other causing the bridges to trip. Hence, in the worst possible case, when system dynamics cause all the bridges to trip, the result will be islanding of clusters and therefore the same as for CI. However, at least some of the bridges could survive so the end result would still be better than outright CI.

Next, let us now consider the effect of a changed cluster power balance due to line failures causing generation failures or islanding of a part of the cluster. In CI, any power balance change will affect only the island where it occurs, worsening its local power balance and most likely requiring additional load shedding - but the other islands are unaffected. Under TP, a changed cluster power balance would trigger a system-wide frequency response with all the system generators responding to a frequency drop. Suppose a power balance change in a cluster is less than the total system frequency reserve. In that case, the system will recover without load shedding so the result would be better than for islanding. However, if the power balance change in a cluster were higher than the system frequency reserve, it would cause an uncontrolled frequency drop, resulting in a possible system-wide under-frequency load shedding. Whether or not the outcome would be better than a localized load shedding in an island under CI would depend on the situation at hand. That question would require a further detailed investigation beyond the scope of this paper.

To summarize the discussion, TP localizes line failures but in extreme situations the effects of voltages, dynamics and power balance changes may require reverting to CI. Hence, one way of taking advantage of both CI and TP would be a two-step defense whereby initially the network is tree-partitioned in order to keep the system together and limit load shedding. However, if and when system dynamics, voltage effects and/or power balance changes threaten the stability of the system, bridges are tripped to separate the clusters, i.e. CI is executed. Analysis of such a two-step defense is beyond the scope of this paper.

VII. SIMULATION RESULTS

This section reports simulation results performed on the IEEE 118-bus network model available in the MATPOWER library [24]. Due to space constraints and also to simplify the considerations, we will show the results using the static DC network model.

A. Identifying clusters and bridges

First the DC power flow problem was solved to calculate all the network flows. Then spectral clustering was undertaken utilizing MATLAB function "spectralcluster", with power flows used as weights. Fig. 3 shows the absolute and relative eigengaps with the latter indicating that the best clustering can be obtained when the network is divided into two or five clusters. As dividing the network into two clusters would not achieve a good localization of disturbances, we decided to use five clusters shown in Fig. 4. The clusters contain 49, 15, 17, 23, and 14 nodes, respectively.
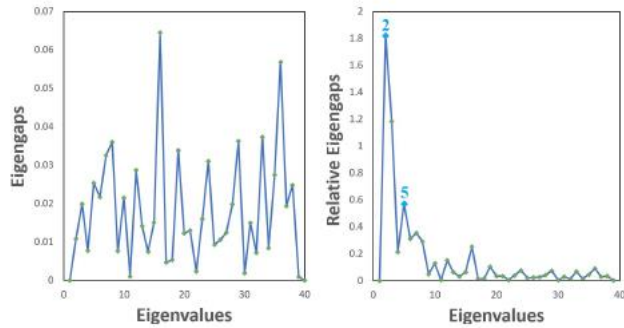
Fig. 3 Absolute and relative eigengaps for 118 node system

The *quotient graph*, i.e. a graph whose vertices are the clusters of the full network and whose edges are the tie-lines, is shown in Fig. 5. Note that the quotient graph is meshed. Power imbalances of each cluster (net export "E" or import "I" in MW) are indicated together with terminal nodes for each tie-line. The numbers on top of tie-lines show their power flows in MW.
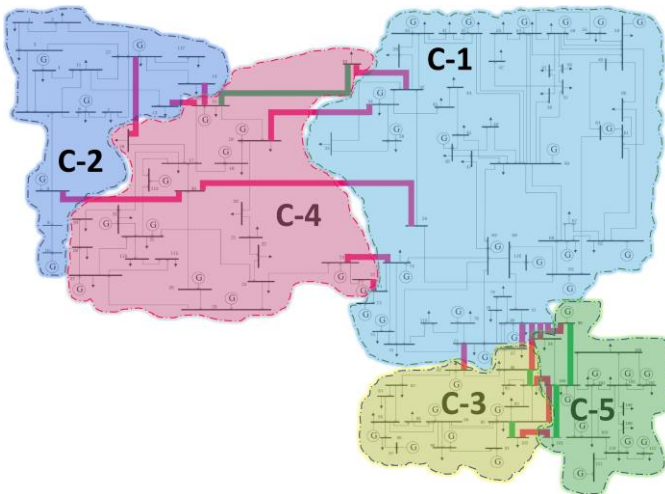

Fig. 4 IEEE 118 node network divided into 5 clusters. Seventeen tie-lines linking the clusters are shown in red. Five congested lines considered in section VII. B. are shown in green.

The modified Prim's algorithm to determine the maximum-weight spanning tree of the quotient graph from Fig. 5 results in the following steps
1. Choose the edge with the highest line flow (90 MW in line 8-30) as the starting edge. So, the starting tree consists of vertices 2 and 4 and the bridge line 8-30.
2. The next heaviest edge connected to the tree is 30-38 carrying 68 MW. Add it to the tree together with vertex 1. The tree consists of vertices 1, 2, and 4.
3. The next heaviest edge connected to the tree is 80-98 carrying 23 MW. Add it to the tree together with vertex 5.
4. The next heaviest edge connected to the tree is 101-102 carrying 39 MW. Add it to the tree together with vertex 3. This completes the algorithm as the tree contains all the vertices.

The resulting maximum-weight spanning tree shown in Fig. 6. The total amount of power flows in the disconnected tie-lines is (1+5+6) + (11+13 +1+7) + 14 + (3+30) + (18+ 22+15) = 146 MW.
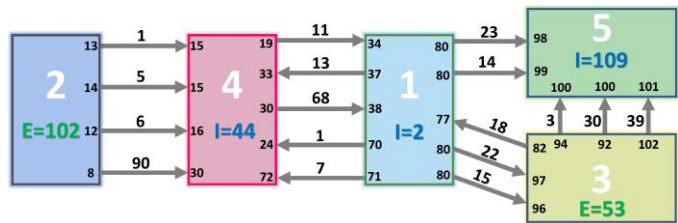

Fig. 5 The quotient graph of IEEE 118 node network.

Bridge flows in Fig. 6 can now be calculated using (4). Cluster power imbalances are $p' = [-2, 102, 53, -44, -109]^T$. Numbering the bridges consecutively in Fig. 5 clockwise starting from the left gives $C' = [0 -1 \ 1 \ 0; 1 \ 0 \ 0 \ 0; 0 \ 0 \ 0 \ 1; -1 \ 1 \ 0 \ 0; 0 \ 0 \ -1 \ -1]$. Utilizing (4) gives bridge flows $f' = [102, 58, 56, 53]^T$ shown in Fig. 6.
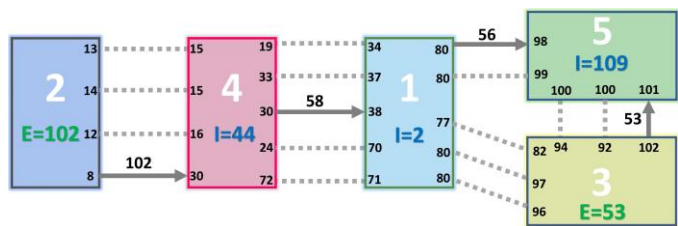

Fig. 6 The maximum-weight spanning tree of the quotient graph.

Note that, as there is a unique path for power to flow along the tree from net exporters to importers, bridge flows can be calculated without using (4) but recursively starting from the leaf clusters. Let us start from leaf cluster 2. It exports 102 MW so the flow in bridge 2-4 is 102 MW. Cluster 4 net-imports 44 MW so the balance (102-44) = 58 is flowing in bridge 4-1. Cluster 1 net-imports 2 MW so the balance (58-2) = 56 MW flows in bridge 1-5. Cluster 5 net-imports 109 MW so the balance (109-56)= 53 must flow in bridge 3-5. This concludes the calculations as the export of cluster 3 is 53 MW.

TP has resulted in a changed pattern of flows with some flows increasing and some decreasing as shown in the heat map in Fig. 7. The vast majority of lines did not increase their flows, with many even reducing them, as shown in blue and green. There were some small increases in loading (shown in yellow) in clusters 3 and 4. Also, five internal lines in clusters 3, 4, and 5 and two tie-lines linking cluster 2-4 and 3-5 (shown in amber) had some increases in loading. Three internal lines changed the flow direction. Only one line (shown in brown) linking clusters 5 and 1 increased its load more significantly. The small impact of TP on the network flows is the result of applying spectral clustering, which minimizes the flows in the tie-lines of the quotient graph, and the modified Prim's algorithm that minimizes the interrupted tie-line flows. Therefore the overall change in network power flows is minimized.
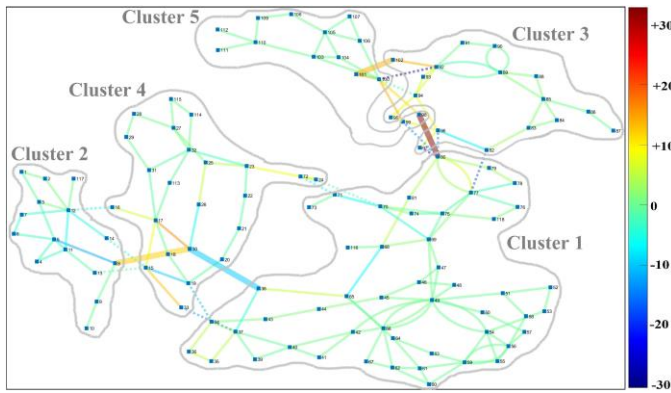
Fig. 7 Heat map showing color-coded changes in power flows following tree-partitioning. The network diagram looks different than that in Fig. 4 as it was produced automatically by MATLAB.

B. Selecting a spanning tree that minimizes congestion

Generally, TP will change power flows in the network which may cause network congestion. To investigate that possibility, we have assumed that lines **(15-33)**, **(94-96)**, **(99-100)**, **(100-101)**, **(92-102)**, shown in green in Fig. 4 operated at their limits prior to TP so any increase in their flows would cause congestion. The reason for choosing those particular lines was that they experienced high increases in power flows due to TP. Choosing the maximum-weight tree shown in Fig. 6 resulted in the total value of overloads equal to 62 MW.

Now, let us instead choose a tree that minimizes load shedding necessary to relieve congestion caused by TP - see section V. B. To do that we have to examine all the possible spanning trees and their number is equal to the cofactor of the unweighted Laplacian matrix L. For the quotient graph shown in Fig. 5 , the Laplacian is:

$$L = \begin{pmatrix} 10 & 0 & -3 & -5 & -2 \\ 0 & 4 & 0 & -4 & 0 \\ -3 & 0 & 6 & 0 & -3 \\ -5 & -4 & 0 & 9 & 0 \\ -2 & 0 & -3 & 0 & 5 \end{pmatrix} \quad (8)$$

and all its cofactors are equal to 420.

To evaluate all spanning trees of the quotient graph shown in Fig. 5 we utilized MATLAB function "generateSpanningTrees(A)". Calculating DC power flow for each spanning tree and evaluating the impact on congestion resulted in choosing the optimal spanning tree shown in Fig. 8 which uses different lines as bridges compared to the maximum-weight spanning tree of Fig. 6. The total value of overloads is reduced to 20 MW from 62 MW for the maximum-weight tree. Note that the bridge flows in Fig. 8 and Fig. 6 are the same as the tree incidence matrix is the same – see (4).

The price we pay for minimizing congestion is increasing the overall shock to the system as quantified by the sum of disconnected tie-line power flows. Comparing Fig. 5 with Fig. 8 gives the total of disconnected power flows equal to (1+5+90) + (13+68+1+7) + 23 + (30+39) + (15+22+18) = 332 MW which is obviously much more than that for the maximum-weight spanning tree equal to 146 MW.
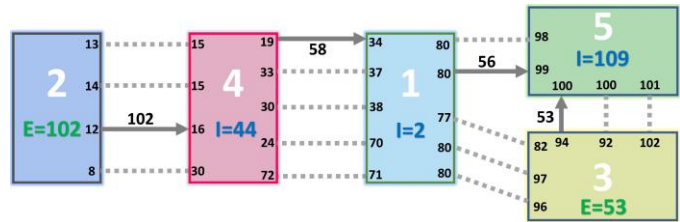

Fig. 8 The optimal spanning tree that minimizes the sum of overloads in IEEE 118 node network.

C. Comparison between CI and TP

Here we provide quantification of some of the advantages of TP over CI which have been compared theoretically in section III. CI requires extensive load/generation shedding to keep the islands in power balance. After being islanded, clusters 2 and 3 would have to shed 102 and 53 MW of generation, respectively, while clusters 1, 4, and 5 would have to shed 2, 44, and 109 MW of load, respectively. Hence the total load shed would be 155 MW. Under TP, the clusters are still connected so power transfers to make up the power balance in each cluster can still take place and no load shedding is necessary.

Obviously both CI and TP could cause network congestion, so some additional load shedding might be necessary. However, it might be expected that congestion due to TP would be less as TP results in a smaller flow reconfiguration – see the next paragraph.

Now let us consider a shock to the system from flow reconfiguration which could result in congestion and stability problems. For CI, all the tie-lines linking the clusters, see Fig. 5, would have to be cut so the total interrupted power over all the tie-lines would be 366 MW. For TP, some of the tie-lines would be left connected as bridges, see Fig. 6, and the total power cut on tie-lines would be only 146 MW, see section VII. A.

D. Execution times

The computer used in this study was a DELL laptop with Intel Core i7 processor and 16GB RAM. Execution time for spectral clustering was 0.18 seconds, Prim's algorithm 0.02 seconds, calculating all possible spanning trees 0.3 seconds, and running DC power flows for all the trees to find a tree that minimizes congestion 8.8 seconds. Clearly the execution times are not excessive and suggest that the methodology can be implemented in real-time even for large systems. The most time-consuming step was running the power flows to find a tree that minimizes congestion. The algorithm could be sped up by omitting irrelevant trees but that issue would have to be investigated further.

VIII. CONCLUSIONS

This paper argues that controlled islanding (CI) to stop a cascading blackout is a drastic move which the industry is finding hard to adopt in practice. We argue that tree-partitioning (TP), which also limits cascading of line failures to within a cluster, is less drastic and has the following advantages. Firstly,

the clusters are still interconnected so that power transfers between them can still take place helping to balance each cluster and limiting any necessary load shedding. Secondly, fewer lines are cut so that the shock to the system is less and therefore system stability and congestion are less affected. Thirdly, there is no need to re-synchronize the clusters after the emergency.

This paper has made the following original contributions. Firstly, we have proposed TP as an emergency measure to stop cascading line failures. Secondly, we have offered a graph-theoretical justification for TP which is much simpler than the one based on the spectral representation of the network graph previously proposed in the literature. Thirdly, we have addressed a vital question of how to split a network into tree-connected clusters. We have suggested a two-stage approach. In the first step, we have used spectral clustering to identify closely-connected clusters of nodes in both static and dynamic sense. Then we have examined two approaches to select which of the tie-lines linking the clusters should be retained as bridges. Application of Prim's algorithm minimizes the overall impact of cutting tie-lines while the impact on congestion, and therefore also load shedding required, can be minimized by evaluating power flows for each possible spanning tree.

In this paper, we have concentrated on analyzing TP using the static DC network model as the first step in our investigation. The effects of the AC model and system dynamics have been analyzed only at a high level and require further research.

One way of taking advantage of both CI and TP would be a two-step defense whereby initially a network is tree-partitioned into clusters to keep the system together and limit load shedding that would be necessary for CI. However, when system dynamics, voltage effects and/or generation failures threaten the system stability, CI is executed to separate the clusters.

Test results examining the performance of the proposed methodology on the 118 node IEEE test network have confirmed the usefulness of the method.

IX. ACKNOWLEDGEMENTS

X. REFERENCES

[1] J. W. Bialek, "Why has it happened again? Comparison between the UCTE blackout in 2006 and the blackouts of 2003," *2007 IEEE Lausanne Power Tech*, Lausanne, 2007.

[2] S. Liu *et al.*, "Robust System Separation Strategy Considering Online Wide-Area Coherency Identification and Uncertainties of Renewable Energy Sources," *IEEE Trans. Power Systems*, vol. 35, no. 5, Sept. 2020

[3] K. Sun, D.-Z. Zheng and Q. Lu, "Splitting strategies for islanding operation of large-scale power systems using OBDD-based methods," in *IEEE Trans. Power Systems*, vol. 18, no. 2, pp. 912-923, May 2003

[4] C.Wang, B. Zhang, Z. Hao, J. Shu, P. Li, and Z. Bo, "A novel real-time searching method for power system splitting boundary," *IEEE Trans. Power Syst.*, vol. 25, Nov. 2010.

[5] L. Hao, G. W. Rosenwald, J. Jung, and C. C. Liu, "Strategic power infrastructure defense," *Proc. IEEE*, vol. 93, no. 5, pp. 918–933, May 2005

[6] A. Peiravi and R. Ildarabadi, "A fast algorithm for intentional islanding of power systems using the multilevel kernel k-means approach," *J. Appl. Sci.*, vol. 9, no. 12, pp. 2247–2255, 2009

[7] H. You, V. Vittal and X. Wang, "Slow coherency-based islanding," *IEEE Transactions on Power Systems*, vol. 19, Feb. 2004

[8] J. Quiròs-Tortos, R. Sanchez-Garcia, J. Brodzki, J. Bialek, and V. Terzija, "Constrained spectral clustering-based methodology for intentional controlled islanding of large-scale power systems," *IET Generation, Transmission & Distribution*, vol. 9, no. 1, 2014.

[9] L. Ding, F. M. Gonzalez-Longatt, P. Wall, and V. Terzija, "Two-step spectral clustering controlled islanding algorithm," *IEEE Transactions on Power Systems*, vol. 28, no. 1, pp. 75–84, 2013.

[10] S. Kamali, T. Amraee and M. Fotuhi-Firuzabad, "Controlled Islanding for Enhancing Grid Resilience against Power System Blackout," in *IEEE Transactions on Power Delivery*, early access, 2020.

[11] A. Kyriacou, P. Demetriou, C. Panayiotou and E. Kyriakides, "Controlled Islanding Solution for Large-Scale Power Systems," in *IEEE Transactions on Power Systems,* vol. 33, March 2018.

[12] Z. Liu, A. Clark, L. Bushnell, D. S. Kirschen and R. Poovendran, "Controlled Islanding via Weak Submodularity," in *IEEE Transactions on Power Systems*, vol. 34, May 2019.

[13] M. Dabbaghjamanesh et al., "A Novel Two-Stage Multi-Layer Constrained Spectral Clustering Strategy for Intentional Islanding of Power Grids," *IEEE Transactions on Power Delivery*, vol. 35, April 2020.

[14] S. Kamali, T. Amraee and M. Khorsand, "Intentional power system islanding under cascading outages using energy function method," *IET Generation, Transmission & Distribution*, vol. 14, 2020

[15] P. Demetriou, M. Asprou and E. Kyriakides, "A Real-Time Controlled Islanding and Restoration Scheme Based on Estimated States," *IEEE Transactions on Power Systems*, vol. 34, no. 1, Jan. 2019.

[16] H. Shao, S. Norris, Z. Lin and J. Bialek, "Determination of when to island by analyzing dynamic characteristics in cascading outages," 2013 IEEE Grenoble Conference, Grenoble, 2013,.

[17] P. Demetriou, J. Quiro's-Torto's, and E. Kyriakides, "When to island for blackout prevention," *IEEE Systems Journal*, vol. 13, no. 3, pp. 3326–3336, 2018.

[18] L. Guo, C. Liang, S. H. Low "Monotonicity properties and spectral characterization of power redistribution in cascading failures" 55ᵗʰ Annual Allerton Conference, UIUC, Illinois, 2017

[19] L. Guo, C. Liang, A. Zocca, S. H. Low and A. Wierman, "Failure Localization in Power Systems via Tree Partitions," *2018 IEEE Conference on Decision and Control (CDC)*, Miami Beach, FL, 2018.

[20] R. J. Sanchez-Garcia, M. Fennelly, S. Norris, N. Wright, G. Niblo, J. Brodzki, and J. W. Bialek, "Hierarchical spectral clustering of power grids," *IEEE Transactions on Power Systems*, vol. 29, 2014.

[21] L. Guo, C. Liang, A. Zocca, S. H. Low and A. Wierman, "Less is More: Real-time Failure Localization in Power Systems," *2019 IEEE 58th Conference on Decision and Control (CDC)*, Nice, France, 2019.

[22] A. Wood, B. Wollenberg *Power Generation, Operation, and Control* Wiley-Interscience, 1996

[23] Prim, R. C. "Shortest connection networks and some generalizations", *Bell System Technical Journal*, 36 (6), Nov. 1957

[24] R. D. Zimmerman, C. E. Murillo-Sanchez, *MATPOWER (Version 7.0)* [Software], 2020, Available: https://matpower.org

XI. BIOGRAPHIES

Janusz W. Bialek (F'11) received M.Eng. (1977) and Ph.D. (1981) from the Warsaw University of Technology, Poland. He held Chair positions at University of Edinburgh (2003–2009) and Durham University (2009–2014). Currently he is Professor of Power and Energy Systems at the Newcastle University, UK, and Full Professor at Skolkovo Institute of Science and Technology (Skoltech), Moscow. His interests are in smart grids, energy system integration, power system dynamics, power system economics, and preventing electricity blackouts.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Vahid Vahidinasab (SM'17) received PhD degree in electrical engineering from Iran University of Science and Technology, Tehran, Iran, in 2010. He was an Assistant Professor at the Shahid Beheshti University, Iran, and a Senior Research Associate of power and energy systems at the Newcastle University, UK, and managed the inteGRIDy as an EU Horizon 2020 project. He also worked with the EPSRC Active Building Centre (ABC). Currently, he is a Senior Lecturer (Assistant Professor) at the Nottingham Trent University, UK. His research interests are smart grids, energy systems integration, power systems operation as well as and energy markets.