Condensed Grid Topology Detection for a Fully Transparent Distribution Management System

Monika Ruh¹,², Andreas Borer², Göran Andersson¹
¹Power Systems Laboratory, ETH Zurich, Zurich, Switzerland
Email: andersson@eeh.ee.ethz.ch
²Rittmeyer AG, Baar, Switzerland
Email: monika.ruh@rittmeyer.com, andreas.borer@rittmeyer.com

Abstract—Fully integrated distribution management systems (DMSs) do not only require a transparent data architecture. They also have to be able to handle purely radial grid topologies as well as ring grid topologies. Thus, a fully integrated DMS needs an algorithm that examines the grid topology to detect if rings are existing. This paper will present such a topology detection algorithm that acts as an important pre-routine for power flow analysis.

Keywords: Distribution management system, distribution grid, transparent data architecture, grid topology detection, ring detection, application-oriented calculation tools.

1 INTRODUCTION

With the advent of distributed generation during the last years, distribution grids are no longer purely passive load systems, which makes the operation of them more complex. For their control and monitoring, distribution management systems (DMSs) have become more important. Hence, the general aim of the project the authors are working on is the development of a DMS that will meet the new challenges.

To explore the innovation potential of existing DMSs, a state-of-the-art survey has been carried out. As described in detail in [1], the major conclusion of this survey was that there is a need for a fully integrated DMS which has a transparent data architecture. The aim is thus to develop a fully integrated DMS that ensures data consistency and data correctness for its application functions.

As power flow analysis needs information about the actual grid topology, a fully integrated DMS has to guarantee that this information is provided without the risk of data inconsistency or data incorrectness. For this purpose, a special algorithm is developed that examines the actual grid topology to detect all existing rings.

This paper is organized as follows. Section 2 briefly describes the transparency matrix and the condensing algorithm, which are part of a concept for a fully transparent DMS. Section 3 describes the requirements of a power flow calculation method for a fully transparent DMS and gives the motivation for the condensed grid topology detection algorithm. Section 4 elaborates the principle and the procedure of the condensed grid topology detection algorithm. The final Section 5 contains the conclusions and the outlook.

2 A FULLY TRANSPARENT DMS

2.1 The Transparency Matrix

Since grid models are very important in modern DMSs providing grid analysis functions, a so called transparency matrix has been introduced with the idea that the grid model is completely represented in matrix form. For this purpose, the upper part of the transparency matrix contains a block matrix, which is a special incidence matrix representing the grid model. Besides this special incidence matrix, the transparency matrix contains attribute rows, in which the attributes of the data objects of the distribution grid control system are placed.

Fig. 1 visualizes the idea of the transparency matrix for a medium-voltage substation: On the left side of the figure, the transparency matrix of Station A is mapped showing three typical attribute rows. On the right side of the figure, Station A itself is schematically pictured.

The concept of this transparency matrix is elaborated in detail in [2].

2.2 The Condensing Algorithm

As the grid model represented by the transparency matrix is too detailed for many grid analysis functions, an efficient link between the transparency matrix and the simplified node-branch-grid model is needed. The filterability of the transparency matrix allows to establish such an efficient link by filtering, or rather condensing, the transparency matrix into the desired matrix form for application-oriented algorithms.

Hence, this algorithm is denominated as condensing algorithm, and the matrix it derives is called condensed transparency matrix. In [3], the procedure of this condensing algorithm is described.

3 POWER FLOW CALCULATION FOR A FULLY TRANSPARENT DMS

3.1 The Power Flow Calculation Method

The aim of the mentioned project is to develop a fully transparent DMS that does not only provide SCADA functions but also application-oriented calculation tools. As power flow calculation is one of the most important grid analysis functions, the first application the authors are developing for a fully transparent DMS is a power flow calculation method.
Due to its intended purpose, such a power flow calculation method has to have the capability to handle a large variety of distribution grids. It has to be able to determine the unknown quantities for any actual topology of a typical distribution grid. Even though distribution grids are normally operated as radial tree structures, there could be grid situations where rings are existing and power flow analysis is required. Hence, a power flow calculation method for a fully transparent DMS has to be able to handle ring situations as well.

Since tests on radial grid structures showed good conversion for a first version of the power flow calculation method based on the Gauss-Seidel iterative method (see for instance [4]), the intention was to enhance this developed method such that it can be used for meshed grid structures too. The principle approach for ring situations is to “virtually” break open all existing rings. The current which actually would flow over such a “virtually” broken grid branch can then be substituted by two equivalent currents at both ends of this grid branch. This idea of partitioning the current into two so called equivalent injection currents is an approach quite often used to determine power flow quantities in meshed distribution grids (see for instance [5]).

Probably, future tests will show that for meshed grid topologies a method based on Newton-Raphson (e.g. see [4]) has better convergence performance than the developed one. However, in all cases, the power flow calculation method needs information about the actual topology of the distribution grid to analyze. Especially, the knowledge if rings are currently existing is required.

### 3.2 Motivation for the Condensed Grid Topology Detection Algorithm

In order to guarantee that the needed information is provided without the risk of data inconsistency or data incorrectness (see [6]), an appropriate linking procedure between the condensed transparency matrix, which represents the node-branch-grid model, and the power flow calculation method for the fully transparent DMS is needed.

The objective is thus to develop an algorithm which acts as such a linking procedure. More precisely, this algorithm has to work as an important pre-routine that not only checks if rings are currently existing but also examines the condensed transparency matrix more extensively so to gain other relevant information about the condensed grid topology. In fact, the algorithm has to detect the condensed grid topology and therefore is named **condensed grid topology detection algorithm**.

### 4 THE CONDENSED GRID TOPOLOGY DETECTION ALGORITHM

#### 4.1 Detecting of the Condensed Grid Structure

That the algorithm can fulfill its assigned tasks, a suitable strategy for traversing through the condensed grid topology has to be determined. As it is the intention to use, if possible, the same traversing strategy as is used in the first version of the power flow calculation method described in Subsection 3.1, it has to be considered how the unknown quantities like complex voltages and complex currents are associated to grid nodes and grid branches.

For this reason, a radial grid structure with one infeed serves as example to demonstrate the sequence in which the complex currents have to be calculated. In Fig. 2, this exemplary grid structure is pictured together with the branch currents and the complex power values at the nodes. The network is energized by Infeed 1 being
connected to Node A, which is at the “top” of the tree structure. It is common to denominate these nodes adjacent to an infeed as source nodes, like done in [8]. Consequently, Node A is referred to as a source node. Each of the two outgoing branches leading away from Node A constitutes together with its successor branches and successor nodes a so called distribution feeder. In nodes that are adjacent to only one branch, one arm of such a distribution feeder is ending, why [8] denotes them as end nodes. Thus, Node E, Node F, Node I and Node J are called end nodes.

Source nodes and end nodes have an important meaning during power flow calculations in distribution grids: Depending on which power flow variables are known and which ones have to be calculated or rather updated, the determination of a specific power flow variable starts or ends at one of these nodes. For instance, assuming that for each node feeding a load, its complex load current is given, the complex line currents can be calculated by starting at the end nodes using the given complex load currents and proceeding towards the source node. On the path up towards the source node, at each node the complex line currents and their complex load current have to be calculated according to Kirchhoff’s current law (KCL). At Node C, for example, the following equation holds:

\[ I_1 = I_0 + I_8 + I_C, \]  

whereas \( I_C \) is the complex load current supplying the load of Node C and the complex line currents \( I_0 \) and \( I_8 \) have been calculated by using the KCL at Node B respectively Node F. This simple consideration reveals how the algorithm has to travel through the radial tree structure: It has to start at the source node and then to pass in the direction of the power flow until it finds an end node. There, the algorithm has to turn and to pass back towards the source node. On its way back, it checks for each node, if there is a branch not yet being explored. If this is the case, the algorithm has to pass on this branch following the direction of the power flow until it again finds an end node. In Fig. 3, this described traversing method is visualized by means of red, green and blue arrows. Steps towards an end node are visualized with red arrows. Since in radial grid structures, such steps are always in the direction of the power flow (as long as there is no distributed generation), they are denoted as downstream steps. Steps back towards the source node and hence in the opposite direction of the power flow are denoted as upstream steps. They are visualized with green arrows. The blue arrows at the end nodes indicate that the algorithm changes its traversing direction from downstream to upstream. At the head of each arrow, the associated step number is annotated so that the traversing path of the algorithm can easily be traced when observing Fig. 3. This observation of the traversing path reveals that the proposed method has the same principle as the depth-first search (DFS) method, which is one of the most used traversing algorithms in mathematics and computer science (see for instance [9]). As the algorithm uses the condensed incidence matrix as data base, the DFS path is essentially also determined by the identifiers of the infeeds and the lines. To make this comprehensible, all downstream steps of the traversing method are visualized with red arrows together with the associated step number in the condensed transparency matrix, which is also presented in Fig. 3.

The gained knowledge that the traversing method uses the same principle as the DFS algorithm however is not most important: The determination of which operations have to be done at each node or along each branch is more important. Actually, the DFS strategy only provides the sequence in which the relevant calculations have to be performed. As this sequence mainly depends
on how many neighbors each node has, it is worthwhile to classify the nodes into different types:

- **Source node**
- **Normal node**
- **Junction node**
- **End node**

In Fig. 3, these four types of nodes are indicated with different colors. The red colored source node and the green colored end nodes are already discussed above. In contrast to these exterior nodes, the *normal nodes* and the *junction nodes* are appearing inside of the grid structure. Tree topologies assumed, the gray colored normal nodes are adjacent to exactly two branches, whereas the blue colored junction nodes are adjacent to at least three branches\(^2\). Using this classification of the nodes, the working principle of the DFS strategy is briefly described depending on the type of node:

**Source node:**

- If the source node is reached via the infeed branch, mark this infeed branch as *Explored downstream*. Then, chose the adjacent branch that appears in the condensed transparency matrix first from left and is not yet explored. Follow this branch downstream to the next node. Finally, mark both the source node and the chosen branch as *Explored downstream*.

- If the source node is reached via a branch on the way upstream, chose the adjacent branch that appears in the condensed transparency matrix first from left and is not yet explored. Then, follow this branch downstream to the next node. Finally, mark the chosen branch as *Explored downstream* and change the direction marker to *Downstream*.

- If the source node is reached via a branch on the way upstream and if there is no branch that is not yet explored, then follow the branch that leads further upstream to the next node. Finally, mark both the junction node and this branch as *Explored upstream*.

**End node:**

- If an end node is reached via a branch on the way downstream, return back on this branch to its previously considered node being located upstream. Finally, mark both the end node and this branch as *Explored upstream* and change the direction marker to *Upstream*.

With these instructions listed for each type of node, the complete DFS algorithm for a radial grid structure is described in a formal way. By examining Fig. 3, the listed instructions can be easily verified. It has to be mentioned that actually a concrete implementation of such a DFS strategy combines the instructions for all node types such that the type of each node is detected when the latter is explored on the way upstream.

Once the principle of the DFS strategy is explained for radial grid structures, it can be enhanced for grid topologies containing rings. Thus, the exemplary radial grid structure is expanded to a meshed grid structure with two rings: As shown in Fig. 4, two meshed rings are formed by inserting Line 10 and Line 11.

Using the DFS strategy for the grid topology pictured in Fig. 4, the algorithm on its alleged way downstream will meet again nodes that it has already passed by a downstream step. Consequently, such a node is a special junction node with at least three adjacent branches. Two of these adjacent branches form with other branches together the ring structure, which is the reason that the same node is reached again on the “way downstream”. The junction node itself is like an entrance or an exit node to this ring structure. Hence, such a node is denominated as *ring junction node*. Due to the two meshed rings in the exemplary grid structure, this is the case for Node C and Node A. Thus, Node A is both a source node and a ring junction node. In Fig. 4, the ring junction nodes are highlighted with magenta color. Arrived again at such a ring junction node, the algorithm has to turn and to pass back towards the source node, as it is also the case for end nodes. Accordingly, the basic working principle of the DFS algorithm for a ring junction node type combines the instructions for the end node type and the instructions for the junction node type:

\(^2\)Junction nodes and normal nodes can change their type in ring topologies, as written below.
Ring junction node:
- If a ring junction node is reached via a branch on the way downstream and is not yet marked as *Explored downstream*, chose the adjacent branch that appears in the condensed transparency matrix first from left and is not yet explored. Then, follow this branch downstream to the next node. Finally, mark both the considered branch and the ring junction node as *Explored downstream*.
- If a ring junction node is reached via a branch on the way downstream and is already marked as *Explored downstream*, return back on this branch to its previously considered node being located upstream. If the ring detection flag is not yet set to *True*, this has to be conducted. Finally, mark the considered branch as *Explored upstream* and change the direction marker to *Upstream*.
- If a ring junction node is reached via a branch on the way upstream, chose the adjacent branch that appears in the condensed transparency matrix first from left and is not yet explored. Then, follow this branch downstream to the next node. Finally, mark the considered branch as *Explored upstream* and change the direction marker to *Downstream*.
- If a ring junction node is reached via a branch on the way upstream and if there is no branch that is not yet explored, then follow the branch that leads further upstream to the next node. Finally, mark both this branch and the ring junction node as *Explored upstream*.

Although the instructions for the ring junction node type are described in a formal way, they are rather a broad outline, as no details are given what exactly has to be done when a ring junction node is detected. For the further development of the algorithm, it thus has to be decided if rings have only to be detected and marked or if they should be treated in a more sophisticated way.

As distribution grids are mostly operated as radial tree structures, the aim could be that the algorithm determines the cut-off points such that the power losses in the feeder lines of the resulting radial grid topology are minimized. As this determination of such *power-optimal* cut-off points would need already a special algorithm of its own, a less sophisticated criterion is chosen. The aim is that in the resulting radial tree structure each node has the smallest possible *node distance* to the source node. Thereby, the number of branches that have to be passed to get from one node to another node is referred to as their node distance. This criterion of how existing rings can be broken is denoted as *distance-optimal*.

Actually for the power flow calculation method described in Subsection 3.1, it would be possible just to determine the ring junction nodes and to break open the rings exactly at these detected ring junction nodes. But the cut-off points determined in this way would in general not be distance-optimal ones, as in the resulting tree structure not all nodes have the smallest possible node distance. Thus, with the aim to use the condensed grid topology detection algorithm also for supporting the distribution grid operator when existing rings in the actual grid topology really have to be broken open, the distance-optimal cut-off points have to be preferred.

Fig. 5 presents the radial grid structure that results when the rings of the meshed grid structure of Fig. 4 are broken at the detected ring junction nodes during the DFS cycle. In comparison, Fig. 6 presents the resulting radial grid structure that would be the outcome of a method breaking the rings distance-optimally. In Fig. 5 and Fig. 6, the determined node distance to the source node, hereinafter referred to simply as *source node distance*, is annotated for each node. As can be seen by comparing the two figures, the method of breaking open the rings distance-optimally is clearly the better one concerning the resulting node distances of the end nodes: The method of breaking open the rings distance-optimally produces as result a radial grid topology where three end nodes have a source node distance of 3 and one end node has a source node distance of 2. In contrast, in the radial grid topology of Fig. 5, the two end nodes have a source node distance of 6.

Consequently, if the loads have more or less the same power values and if all lines have approximately the same impedance values, the radial grid topology with the distance-optimal cut-off points will be in terms of power consumption almost or even completely optimal.
1) Ring Detection and Determination of the Distance-Optimal Cut-Off Points

To figure out if the determination of the distance-optimal cut-off points needs several DFS cycles or if another search method is required, various test examples have been studied and careful considerations have been made. The findings revealed that the DFS strategy is not an appropriate search method to determine for each node of the power grid its smallest possible source node distance, hereinafter denoted as *minimal source node distance*. The reason is that the DFS strategy, whenever possible, moves downstream over an adjacent grid branch to the next node instead of checking first all neighboring nodes of the considered node. But for the determination of the minimal source node distances exactly this search method would be needed: The search algorithm has to check first all neighboring nodes of the source node, which consequently all have 1 as the minimal source node distance, before moving downstream to one of these neighboring nodes. Then, one after another, these neighboring nodes are checked for their own neighboring nodes not yet explored, which accordingly have a minimal source node distance of 2. In this way, the search algorithm proceeds till it has explored all grid nodes. Hence, this search principle of spreading out radially from the source node to explore first the nearer nodes in the breadth of the power grid is the *breadth-first search* (BFS) method. Like the DFS method, the BFS method is very often used in computer science and for instance described in [9]. The condensed grid topology detection algorithm therefore needs both the DFS principle as well as the BFS principle.

In Fig. 7, the traversing principle of the BFS method is visualized by means of arrows with different colors: Steps towards grid nodes with a minimal source node distance of 1 are visualized with blue arrows. Steps from these nodes to those with a minimal node distance of 2 are drawn with cyan colored arrows. Similarly, steps to nodes with a minimal node distance of 3 are visualized with green arrows and steps to nodes with a minimal node distance of 4 have arrows with dark green color. Since this search principle of spreading out radially explores first the nearer nodes with the smaller minimal node distances, the grid nodes are processed like in levels. For each level, its level number is given by the minimal source node distance of the grid nodes belonging to it. The levels 0 to 3 are visualized with different colored areas in Fig. 7. In addition, at the head of each arrow, the associated step number is annotated so that the spreading out of the algorithm can easily be observed. For instance, Node B is reached by step 4 of the BFS algorithm coming from Node C. As Node C has a source node distance of 1, the minimal source node distance of Node B is consequently set to 2.

Due to the two existing rings, the spreading out of the BFS algorithm reaches Node G and Node I over more than one grid branch, as can be seen in Fig. 7. For this kind of grid nodes, their minimal source node
The minimal source node distance has to be determined when they are reached by the spreading out of the algorithm for the first time. The reason is that the associated path over which such a node is reached is the shortest possible one (or one of the shortest possible paths). The second time such a grid node is reached, the associated path has at least the same length or is even longer than the one over which the considered node was reached the first time. This can be verified for instance for Node G, which is reached the first time by step 6 of the algorithm. Since this step comes from Node D with the minimal source node distance of 1, the minimal source node distance of Node G is determined to 2. The second time Node G is reached from Node F over Line 10 by step 8 of the algorithm. As Node F already has a minimal source node distance of 2, the associated path over which Node G is reached the second time is longer than the one over which it is reached the first time. To visualize that the second time Node G respectively Node I is reached, its minimal source node distance has not anymore to be determined, step 8 and step 11 respectively are drawn with dashed arrows in Fig. 7. The comparison of the minimal source node distances of Node G and Node I with their respective neighboring nodes shows that these two special nodes have the largest minimal source node distance (or one of the largest source node distances) of all nodes of the concerning ring. Thus, given that the minimal source node distance is considered like a negative altitude difference, the two special nodes are lying like in the “valley bottom” of the concerning ring. Thus in the following, this kind of grid nodes are denominated as valley nodes.

In Fig. 7, the valley nodes are highlighted with dark green color. As a valley node per definition has the largest minimal source node distance (or one of the largest source node distances) of all nodes of the concerning ring, it constitutes the distance-optimal cut-off point at which this ring has to be broken. The appropriate grid branch that has to be “virtually” switched off is the one over which the valley node is reached the second time. Hence, in the exemplary grid structure, Line 10 and Line 11 have to be switched off. The comparison of the thus resulting radial grid structure with the distance-optimal radial grid structure pictured in Fig. 6 proves that the BFS method works correctly as desired.

As the condensed grid topology detection algorithm uses the condensed incidence matrix as basis, the sequence in which the nodes of the same level are explored is essentially also determined by the identifiers of the data objects. To make this comprehensible, all steps of the spreading out of the BFS strategy are visualized with arrows together with the associated step numbers in the condensed transparency matrix, which is also presented in Fig. 7. Each of these arrows has the same color as the corresponding step arrow pictured together with the exemplary meshed grid structure.

With the basic working principle of the BFS method explained, the algorithm can be described in more detail. In particular, the setting of the minimal source node distances, the detection of the valley nodes and the identification of the grid branches to “virtually” switch off are explicitly described.

To make the results of this first subprocedure of the condensed grid topology detection algorithm available for the second subprocedure, they have to be stored in an appropriate form. Thus, the condensed transparency matrix has to be enhanced with two attribute rows:

- Minimal source node distance row
- Ring number stack row

As its name suggests, the first of these attribute rows stores the determined minimal source node distances of all nodes of the condensed grid topology. Consequently, the elements of this attribute row are only determined for data objects of grid nodes. The second of these attribute rows contains for each data object representing a grid node or a grid branch a special stack. Such a stack stores the ring number of each ring that is attached to the concerning data object.

To store the minimal source node distances already determined during the BFS algorithm is working, a column vector is needed, which is denominated minimal source node distance vector. This column vector has for each node of the actual grid topology one element as shown in Fig. 7. To store the actual source node distance, an actual source node distance variable is needed. For counting the detected rings, a variable denominated as number of detected rings is defined and is initially set to 0. A grid discovered variable is required, which stores if the whole grid structure is already explored. This variable is initialized with True. The actual source node distance has to be determined when they are reached by the spreading out of the algorithm for the first time. The reason is that the associated path over which such a node is reached is the shortest possible one (or one of the shortest possible paths). The second time such a grid node is reached, the associated path has at least the same length or is even longer than the one over which the considered node was reached the first time. This can be verified for instance for Node G, which is reached the first time by step 6 of the algorithm. Since this step comes from Node D with the minimal source node distance of 1, the minimal source node distance of Node G is determined to 2. The second time Node G is reached from Node F over Line 10 by step 8 of the algorithm. As Node F already has a minimal source node distance of 2, the associated path over which Node G is reached the second time is longer than the one over which it is reached the first time. To visualize that the second time Node G respectively Node I is reached, its minimal source node distance has not anymore to be determined, step 8 and step 11 respectively are drawn with dashed arrows in Fig. 7. The comparison of the minimal source node distances of Node G and Node I with their respective neighboring nodes shows that these two special nodes have the largest minimal source node distance (or one of the largest source node distances) of all nodes of the concerning ring. Thus, given that the minimal source node distance is considered like a negative altitude difference, the two special nodes are lying like in the “valley bottom” of the concerning ring. Thus in the following, this kind of grid nodes are denominated as valley nodes.

In Fig. 7, the valley nodes are highlighted with dark green color. As a valley node per definition has the largest minimal source node distance (or one of the largest source node distances) of all nodes of the concerning ring, it constitutes the distance-optimal cut-off point at which this ring has to be broken. The appropriate grid branch that has to be “virtually” switched off is the one over which the valley node is reached the second time. Hence, in the exemplary grid structure, Line 10 and Line 11 have to be switched off. The comparison of the thus resulting radial grid structure with the distance-optimal radial grid structure pictured in Fig. 6 proves that the BFS method works correctly as desired.

As the condensed grid topology detection algorithm uses the condensed incidence matrix as basis, the sequence in which the nodes of the same level are explored is essentially also determined by the identifiers of the data objects. To make this comprehensible, all steps of the spreading out of the BFS strategy are visualized with arrows together with the associated step numbers in the condensed transparency matrix, which is also presented in Fig. 7. Each of these arrows has the same color as the corresponding step arrow pictured together with the exemplary meshed grid structure.

With the basic working principle of the BFS method explained, the algorithm can be described in more detail. In particular, the setting of the minimal source node distances, the detection of the valley nodes and the identification of the grid branches to “virtually” switch off are explicitly described.

To make the results of this first subprocedure of the condensed grid topology detection algorithm available for the second subprocedure, they have to be stored in an appropriate form. Thus, the condensed transparency matrix has to be enhanced with two attribute rows:

- Minimal source node distance row
- Ring number stack row

As its name suggests, the first of these attribute rows stores the determined minimal source node distances of all nodes of the condensed grid topology. Consequently, the elements of this attribute row are only determined for data objects of grid nodes. The second of these attribute rows contains for each data object representing a grid node or a grid branch a special stack. Such a stack stores the ring number of each ring that is attached to the concerning data object.

To store the minimal source node distances already determined during the BFS algorithm is working, a column vector is needed, which is denominated minimal source node distance vector. This column vector has for each node of the actual grid topology one element as shown in Fig. 7. To store the actual source node distance, an actual source node distance variable is needed. For counting the detected rings, a variable denominated as number of detected rings is defined and is initially set to 0. A grid discovered variable is required, which stores if the whole grid structure is already explored. This variable is initialized with True. The actual source node
distance variable and all elements of the minimal source node distance vector are initialized with -1. With the data preparation done, the BFS algorithm can be started:

1. Search in the condensed transparency matrix for the infeed branch. Then, search the adjacent node of this infeed branch, which is the source node. Set the value in the corresponding element of the minimal source node distance vector to 0. Finally, mark the infeed branch as Explored and set the actual source node distance variable to 0.
2. Search in the minimal source node distance vector for elements that have the same value as the actual source node distance variable. For each such vector element that is found, examine the corresponding row of the condensed transparency matrix by doing the following:
   [A] Chose the adjacent branch that appears in the condensed transparency matrix first from left and is not yet explored. Then, search for this branch its other adjacent node. For this other node, check if the value in the corresponding element of the minimal source node distance vector is set to -1:
   - If this is the case, set the value in this element of the minimal source node distance vector to the value of the actual source node variable increased by 1. Set the value in the corresponding element of the minimal source node distance row of the condensed transparency matrix to the same value as well. If the grid discovered boolean is not yet set to False, this has to be done.
   - If this is not the case, then this other node has already been reached and consequently is detected as a valley node of a ring. Thus, the variable storing the number of detected rings has to be increased by 1. As the ring itself is to be identified by the actual number of detected rings, this number has to be stored in the ring number stack of the considered grid branch. In addition, the actual number of detected rings has also to be stored in the ring number stack of the valley node and in the ring number stack of the other adjacent node. Then, the considered branch has to be “virtually” switched off. For that, check all incidence elements of the corresponding column of the condensed transparency matrix:
     - If the condensed incidence matrix element is 0, nothing has to be changed.
     - If the condensed incidence matrix element contains a negative connectivity integer (see [3]), replace it with the connectivity integer 5 expressing a “virtually” opened branch that is leaving the concerning node.
     - If the condensed incidence matrix element has a positive connectivity integer, replace it with the connectivity integer 5 expressing a “virtually” opened branch that is leaving the concerning node.
   Finally, mark the considered branch as Explored.
3. Repeat step arrows and the depicted condensed transparency matrix in Fig. 7.

Within these three steps of the algorithm, all instructions needed to determine the minimal source node distance of every node, to detect every valley node and to identify every grid branch that has to be “virtually” switched off are included. Thereby, 2 and 3 are repeated until the grid structure is completely discovered. The instructions can easily be verified by examining the exemplary grid structure together with the visualized step arrows and the depicted condensed transparency matrix in Fig. 7.

In Fig. 8, the result of this described algorithm is presented for the exemplary grid structure: Line 10 and Line 11 are correctly identified as branches that have to be “virtually” switched off. In the corresponding columns of the condensed transparency matrix, it can be seen that their connectivity integers are set to -5 and 5 as it should be for “virtually” switched off branches. For each of these “virtually” switched off branches, the identifier of the attached ring is stored in the corresponding cell of the ring number stack row, which is added to the condensed transparency matrix. Thus, it can be checked that Line 10 is contained in the ring with the number 1 and Line 11 is contained in the ring with the number 2. The ring number stacks of these branches and the ring number stacks of their adjacent nodes are also annotated in the picture of the meshed grid structure. For each node, the determined minimal source node distance is annotated too.

In the following, the identification of the grid branches that are within a ring structure will show that each of the “virtually” switched off branches is belonging to just one ring. Hence, these special branches are denominated as
**ring branches.** As the DFS algorithm will not pass over these ring branches, but turn at their adjacent nodes and pass back towards the source node, these adjacent nodes, namely the valley nodes and the nodes at the other end of the ring branches, get a new meaning: With regard to the ring branches, these nodes act like special end nodes. Because of that, they are renamed to **ring end nodes** and are highlighted with cyan color.

2) Identification of the Grid Branches within the “Virtually” Broken Rings

In case of existing rings in the actual topology, the first subprocedure identifies all ring end nodes. This makes it possible to identify all branches that are within a “virtually” broken ring by just one second DFS cycle through the condensed grid topology. Thereby, besides the basic instructions described in Subsection 4.1, additional tasks have to be executed for each type of node. For this purpose, the ring number stacks are used again: For each branch, its allocated ring number stack stores in which rings it is contained. For each node, its allocated ring number stack stores which rings lead through it.

In the following, the additional instructions to perform in this DFS cycle are described in detail. As all additional instructions have to be performed only on the way **upstreams**, the basic instructions are not repeated again. Furthermore, the end node type is not listed at all because no additional instructions have to be done for it. The additional instructions for the source node type, the junction node type and the ring end node type are absolutely identical. Thus, these three node types are described together:

**Source node, junction node and ring end node:**
- If the node is reached via a branch on the way upstream, in addition to the basic instructions, copy all ring numbers of the ring number stack of the previously checked node to the ring number stack of the considered branch. Then, compare the ring number stack of the considered node with the ring number stack of the previously checked node:
  - If a ring number in the ring number stack of the previously checked node does not appear in the ring number stack of the considered node, then copy this ring number to the ring number stack of the considered node.
  - If the same ring number is found in both stacks, that implies that the considered node has already been reached via a branch contained in the ring with this ring number. Hence, the considered branch “closes” the concerning ring. For that reason, delete this ring number from the ring number stack of this node.

**Normal node:**
- If a normal node is reached via a branch on the way upstream, in addition to the basic instructions, copy all ring numbers of the ring number stack of the previously checked node to the ring number stack of the considered branch. Copy all ring numbers
algorithm work closely together to provide a fully transparent data architecture for a DMS that has base system functions and also advanced application functions.

5.2 Outlook

A possible enhancement of the condensed grid topology detection algorithm is to determine instead of the distance-optimal cut-off points the power-optimal ones. As defined in Subsection 4.1, power-optimal cut-off points break the existing rings such that the power losses in the feeder lines of the resulting radial grid topology are minimized.

This enhancement would allow to use the condensed grid topology detection algorithm as a kind of an optimization and planning tool: For a given load situation, the distribution system operator can start this tool to determine the optimal radial grid topology. A fully transparent DMS providing such an optimization and planning tool would have two main advantages. First, there is no need for an additional (mostly expensive) grid planning software and second, since the planning and optimization work is done in the DMS and thus no external data exchange is needed, there is no risk for data inconsistency or data incorrectness.

ACKNOWLEDGMENT

The authors want to thank Christoph Bücheler, Matthias Hartmann, Gerhard Nigg and Bernhard Müller of Rittmeyer AG for valuable and stimulating discussions. The financial support of the project by Rittmeyer AG is gratefully acknowledged.

REFERENCES