

# Determination of the Minimum Break Point Set of Directional Relay Networks based on k-trees of the Network Graphs

Bapeswara-Rao V. Vinnakota and Cristinel Ababei, *Member, IEEE*

**Abstract**—We propose a new graph theory based algorithm for the determination of the minimum break point set of meshed systems. We show that if there is a k-tree in the graph  $G(V, E)$  associated with the system, then the minimum break point set has a cardinality of  $|E| - |V| + k$ . The proposed algorithm implements an iterative procedure to identify such a k-tree and having minimum k. It is found that the cardinality of the minimum break point set is 9 for the IEEE 14 bus system and 16 for the IEEE 30 bus system. These minimum values are smaller than the previously best reported results of 10 and 18 for the same systems.

**Index Terms**—Directional relays. Break points. Minimum number of break points. k-trees.

## I. INTRODUCTION

THE problem of directional relay coordination in a meshed system requires the selection of a set of relays. Since opening of these relays opens all directed simple loops in the system, this set is called a break point set (BPS) [1]. The minimum cardinality BPS is known as the minimum break point set (MBPS). Determination of the MBPS in interconnected networks is the key step during calculation and setting of overcurrent relays protection values.

Even though the problem of locating the minimum set of break points of a meshed system has been studied extensively, most of the previous solutions obtain a minimal and not a minimum set. Generally, these solutions are based on the enumeration of all loops in the network graph associated with the system. The enumeration process may become prohibitively computationally expensive due to the exponential increase of the time complexity [2], [3]. Recently, the authors of [4] have proposed an integer linear programming (ILP) solution to find the optimal break point set. Their solution represents the best result (break point set of cardinality 10 for IEEE 14 bus system) in the published literature and improves over many other recent approaches cited and discussed in [4]. The ILP solution and the recent genetic algorithm based approach studied in [5] have found the break point set of cardinality 18 for IEEE 30 bus system. In this paper, we introduce a new algorithm for the determination of the minimum set of break points. Our algorithm identifies a minimum set of cardinality 9 and 16 for the IEEE 14 bus system and the IEEE 30 bus system, respectively.

B.-R.V. Vinnakota and C. Ababei are with the Department of Electrical and Computer Engineering, NDSU, Fargo ND, 58108, USA (e-mail: bapessa.rao@ndsu.edu; cristinel.ababei@ndsu.edu);

The remainder of the paper is organized as follows. Section II presents the statement of the problem addressed in this paper. Also, our main contribution is outlined in this section. Section III describes the proposed algorithm for the the determination of the minimum set of break points. In Section IV, we report experimental results for several systems, including the IEEE 14 and 30 bus systems. The paper is concluded in Section V.

## II. STATEMENT OF THE PROBLEM AND CONTRIBUTION

Let  $N$  be the system or network<sup>1</sup> on which directional relays are to be placed and  $G(V, E)$  be its associated graph, where  $V$  is the set of nodes and  $E$  is the set of edges, respectively. A bus of the network  $N$  corresponds to a node in  $V$  and a line of  $N$  corresponds to an edge in  $E$ . We assume that the network and therefore the associated graph  $G$  is connected and has no self-loops [6]. A *directional relay* placed on a line of the network  $N$  is represented by an arrow-head alongside the corresponding edge of the graph. The arrow-head is called a *dire* (short for directional relay). An example network  $N$  is shown in Fig.1.a and its corresponding graph  $G(V, E)$  is shown in Fig.1.b. The graph  $G$  has the set of nodes  $V = \{v_1, v_2, v_3, v_4\}$  and the set of edges  $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ . A directional relay on line  $L_1$  at bus  $B_1$  in network  $N$  is represented by a *dire* along edge  $e_1$  near the node  $v_1$  in the graph  $G$ , oriented away from the node  $v_1$ .

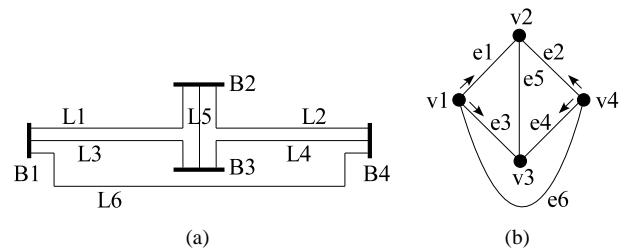


Fig. 1. (a) Example system or network  $N$  with four buses. (b) The associated graph  $G$  with four nodes and six edges.

The problem of computing the minimum break points set (MBPS) is stated or formulated as follows. *Given the connected associated graph  $G(V, E)$  of a network  $N$ , place as few dires as possible, such that when one travels along each*

<sup>1</sup>System and network are used interchangeably in this paper.

loop in  $G$  (in any direction), at least one dire oriented against the direction of the travel is encountered.

In this paper, we introduce a new graph theory based algorithm for the determination of the minimum set of break points. Our main contribution is as follows: 1) The proposed deterministic algorithm is based on finding a  $k$ -tree whose complementary set of edges in the system graph  $G$  corresponds to the minimum set of directional relays if (i) each edge in the complementary set has its vertices in different parts of the  $k$ -tree and (ii) no  $(k-1)$ -tree exists which satisfies the first condition, 2) To address the computational time for large systems we introduce novel techniques to simplify the network graph without compromising the quality of the final result, and 3) We report experimental results, which improve on the state of the art: the proposed algorithm can find a minimum break point set of cardinality 9 and 16 for the IEEE 14 and 30 bus systems respectively, which is smaller than the cardinality of 10 and 18 reported for the same systems in [4], [5].

### III. DETERMINATION OF THE MINIMUM BREAK POINT SET OF DIRES

#### A. Range of Possible Values for the Cardinality of the Minimum Break Point Set

Let  $\delta$  be the minimum number of dires required to satisfy the condition stated for a given graph in the MBPS problem formulation from the previous section. It is important to note that, for a given graph  $G(V, E)$ ,  $\delta$  can only be a number from the integer interval  $[\delta_{lower\_bound}, \delta_{upper\_bound}] = [|E| - |V| + 2, |E|]$ . The upper bound of the range of possible values of  $\delta$  for the given graph  $G$  is  $\delta_{upper\_bound} = |E|$  and represents the worst-case situation, when all lines of the network  $N$  must have a directional relay. An example of a system graph, which requires  $\delta = \delta_{upper\_bound}$  is shown in Fig.2.b. The lower bound of the range of possible values of  $\delta$  for the given graph  $G$  is  $\delta_{lower\_bound} = |E| - |V| + 2$  and represents the best-case situation, when only the minimum possible number of lines of the network  $N$  must have a directional relay. The system shown in Fig.2.a is an example when this situation occurs:  $\delta = 6 - 4 + 2 = 4$  dires are needed for the graph with four vertices ( $|V| = 4$ ) and six edges ( $|E| = 6$ ). In addition, it is also important to note that, the location of the dires is not unique. For example, a different placement of the same number  $\delta = 4$  of dires for the graph from Fig.1.b is shown in Fig.2.a.

Because for a given graph  $G(V, E)$ ,  $\delta$  is an integer from the interval  $[|E| - |V| + 2, |E|]$ , we can write it as  $\delta = |E| - |V| + k$ , where the integer  $k$  corresponds to a suitable  $k$ -tree (details

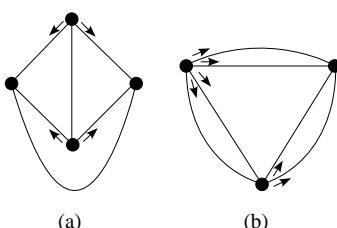


Fig. 2. Example graphs with different  $\delta$ 's (a) Graph with  $\delta = |E| - |V| + 2$ . (b) Graph with  $\delta = |E|$ .

will be provided in the next subsections) in the given graph  $G$  and is uniquely determined from the interval  $k \in [2, |V|]$ . In this paper we propose an algorithm to find the minimum  $k$  (thereby the minimum number of dires  $\delta$ ) and identify the location of dires. The proposed algorithm is based on an iterative procedure, which starts with the best case situation  $k = 2$  and may iterate until  $k = |V|$ . In the next subsection, we present the outline of proposed algorithm and then we provide details in the subsequent subsections.

#### B. Outline of the Proposed Algorithm

In this subsection we present our algorithm for finding  $\delta = |E| - |V| + k$  and the location of dires for a given graph. As mentioned in the previous subsection, the proposed procedure is an iterative search process to find the minimum integer  $k$  belonging to  $[2, |V|]$ , and thereby a  $\delta$ , which satisfies the condition of having at least one dire oriented against the direction of travel in any loop of the given graph  $G$  in any direction. The pseudocode of the proposed iterative algorithm is shown in Fig.3.

#### Algorithm 1: Determination of the MBPS

```

1: compute_MBPS()
2: In: graph  $G(V, E)$ 
3: Out:  $\delta$  and location of dires
4: Set  $k \leftarrow 1$ 
5: repeat
6:    $k \leftarrow k + 1$ 
7:    $\delta \leftarrow |E| - |V| + k$ 
8:    $k\_tree\_found \leftarrow check\_delta(\delta)$ 
9: until  $!k\_tree\_found$  OR  $k = |V|$ 
10: return  $\delta$  and location of dires

```

Fig. 3. Pseudocode of the general algorithm for the determination of the minimum break point set of dires. The procedure  $check\_delta(\delta)$  is described in Fig.4.

#### Algorithm 2: Determination of $k$ -tree

```

1: check_delta( $\delta$ )
2: In: graph  $G(V, E)$ ,  $\delta$  for a given  $k$ 
3: Out:  $k$ -tree if any and location of dires
4: repeat
5:   Enumerate new set  $E^i_{|V|-k}$  of  $|V| - k$  edges
6:   Build incidence sub-matrix corresponding to  $E^i_{|V|-k}$ 
7:   Compute rank  $r$  of the sub-matrix
8:   if  $r = |V| - k$  then
9:     Find parts  $S_a, S_b, \dots$ 
10:    if all edges in  $T_{compl}$  have ends in different parts then
11:      Record that  $k$ -tree is found
12:      Mark dires
13:      return true (k-tree found)
14:    end if
15:  end if
16: until all possible combinations of  $|V| - k$  edges out of  $E$  have been enumerated
17: return false (k-tree not found)

```

Fig. 4. Pseudocode of the procedure  $check\_delta(\delta)$  for searching a  $k$ -tree. Note that the enumeration process would stop in line 13 and the procedure call finishes once a  $k$ -tree is found. This procedure is utilized inside the proposed algorithm from Fig.3.

We start by checking if a break point set with  $\delta = |E| - |V| + k, k = 2$  dires is sufficient to satisfy the condition of having at least one dire oriented against the direction of travel in any loop in any direction. If no such set exists, then we continue this process and check whether  $\delta = |E| - |V| + k, k = 3$  dires are sufficient. If not, then we repeat the check for  $\delta = |E| - |V| + k, k = 4$  and so on. This iterative process can continue until the  $k = |V|$  is reached, when  $\delta = |E|$ , which requires that all edges in the system graph  $G$  must be marked with a dire.

The proposed algorithm relies on the procedure *check\_delta*( $\delta$ ) (called in line number 8 from Fig.3), which is detailed in Fig.4 and described in the following subsections.

### C. Networks for which $\delta = |E| - |V| + k, k = 2$

The procedure from Fig.4 is illustrated with respect to the graph  $G$  shown in Fig.5.a. Consider the 2-tree of  $G$  formed by the edges  $T_a \cup T_b = \{e1, e4, e6\}$ , where  $T_a = \{e1, e4\}$  and  $T_b = \{e6\}$  represent the two parts (i.e., sub-trees) that constitute the 2-tree (see Fig.5.b). Let  $S_a = \{v1, v4, v5\}$  be the set of vertices of  $T_a$  and  $S_b = \{v2, v3\}$  be the set of vertices of  $T_b$ . Note that in some cases, one of these sets may have only one element [6].

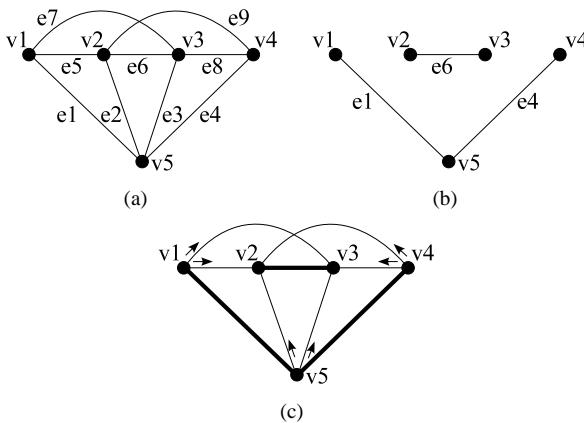


Fig. 5. (a) Example graph. (b) The relevant 2-tree. (c) Graph with dires.

The complement of the 2-tree is the set of edges  $T_{compl} = \{e2, e3, e5, e7, e8, e9\}$ . Each of these edges has one of its vertices (or ends) in  $S_a$  and the other in  $S_b$ . Then, the following statement is valid (its justification is presented in the next paragraph):

**Statement 1:** The minimum number of dires  $\delta$  for the graph  $G$  is precisely the cardinality of the set  $T_{compl}$  (i.e.,  $|E| - |V| + k = 9 - 5 + 2 = 6$ ).

Therefore, the minimum break point set of dires can be placed on these edges near vertices in  $S_a$  (oriented away from the vertex in  $S_a$  at which each edge is incident). For example, Fig.5.c shows the location of the minimum number of dires marked along the edges in  $T_{compl}$  (edges of the relevant 2-tree are drawn with thick lines). It is important to emphasize that dires are not located on any of the edges of the 2-tree.

The justification of Statement 1 is as follows. Because by definition the trees  $T_a$  and  $T_b$  contain no loops, any loop of

$G$  must include some edges from the complementary edge set  $T_{compl} = G - T_a - T_b$ . These edges form a cutset of  $G$  since the removal of these edges leaves  $G$  in two disconnected subgraphs [6]. In addition, each loop in  $G$  will contain an even number of edges from this set. Thus, when one travels along a given loop of  $G$ , the traveling path must traverse an edge from the set  $T_{compl}$  going towards a vertex in  $S_a$  and later will be leaving from a vertex in  $S_a$  (in the opposite direction). Since each edge of this set has a dire oriented away from the vertices of  $S_a$ , the traveling will encounter at least one dire oriented against the direction of travel.

Searching for a 2-tree inside the procedure from Fig.4 is based on an enumeration process. We enumerate all possible sets with  $|V| - 2$  edges of the  $|E|$  edges (a procedure for such enumeration is presented in [7]). For the graph in Fig.5 these sets are  $\{e1, e2, e3\}, \{e1, e2, e4\}, \{e1, e2, e5\}, \dots, \{e7, e8, e9\}$ . For each of the enumerated  $|V| - 2$  edge sets, we build the sub-matrix of the incidence matrix of  $G$  corresponding to the edge set. Then, the following statement is valid [6]:

**Statement 2:** If this sub-matrix has its rank equal to  $|V| - k = |V| - 2$ , then the enumerated edge set constitutes a  $k$ -tree of  $G$ .

If in addition, all edges from the complementary edge set  $T_{compl} = G - T_a - T_b$  have their ends in different parts of the 2-tree, then this set  $T_{compl}$  gives the edges which will have dires placed on them (away from the vertices of  $S_a$ ). At this point, the process of identification of the minimum number of break points is completed. It is also worth noting that, the rows and columns of the sub-matrix can be arranged to make it a block-diagonal matrix. The rows of one block correspond to the vertices of one part ( $S_a$ ) of the 2-tree while the rows of the second block correspond to the vertices of the other part ( $S_b$ ) of the 2-tree.

### D. Networks for which $\delta = |E| - |V| + k, k = 3$

This is the case of the second main iteration of the proposed algorithm from Fig.3. The complete graph with five vertices is an example for such a case (see Fig.6). We now search for a 3-tree formed by  $|V| - 3$  edges of the graph  $G$ . The 3-tree represents three separate parts [6]. Similar to the earlier case (described in the previous subsection for 2-trees), one or more of these parts may have only one vertex. Let  $T_a, T_b$ , and  $T_c$  be the three parts of the 3-tree and also let  $S_a, S_b$ , and  $S_c$  be their corresponding sets of vertices.

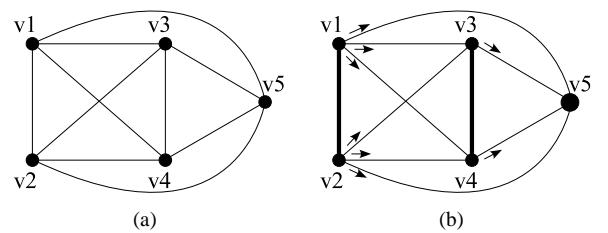


Fig. 6. (a) Complete graph with 5 vertices. (b) Relevant 3-tree.

**Statement 3:** Following a rationale similar to that of Statement 1, if each edge in the complementary set of edges  $T_{compl} =$

$G - T_a - T_b - T_c$  has its vertices in different parts of the 3-tree, then those edges correspond to the minimum set of dires.

We place a dire on every edge present in the complementary set which is incident to a vertex in  $S_a$ . It is placed near the vertex in  $S_a$  and is oriented away from it. Then, we place a dire on every edge present in the complementary set which is incident to a vertex in  $S_b$  if that edge is not already marked by a dire. The dire is placed near the vertex in  $S_b$  and is oriented away from it.

For the example in Fig.6.a the 3-tree has the parts  $S_a = \{v1, v2\}$ ,  $S_b = \{v3, v4\}$ , and  $S_c = \{v5\}$  and the minimum number of dires is  $\delta = |E| - |V| + k = 10 - 5 + 3 = 8$ . The edges of the 3-tree are shown by thick lines in Fig.6.b, which also shows the marking of the dires. Note that the third component of the 3-tree is an isolated vertex.

The search process for a 3-tree is similar to the case of searching for a 2-tree and described generally in Fig.4. The difference is that now we enumerate sets with  $|V| - 3$  edges of the  $|E|$  edges.

**Statement 4:** Following a rationale similar to that of Statement 2, if the sub-matrix of the incidence matrix of  $G$  that corresponds to one of these enumerated sets has its rank equal to  $|V| - k = |V| - 3$ , then the enumerated edge set constitutes a  $k$ -tree of  $G$ .

If in addition, all edges from the complementary edge set  $T_{compl} = G - T_a - T_b - T_c$  have their ends in different parts of the 3-tree, then this set  $T_{compl}$  gives the edges which will have dires placed on them (away from the vertices of  $S_a$  and  $S_b$ ). Once this situation is identified, the process of identification of the minimum number of break points is completed. Again, note that the rows and columns of the sub-matrix can be arranged to make it a three block-diagonal matrix. The vertices in each block correspond to vertices in each part of the 3-tree.

#### E. Generalization $\delta = |E| - |V| + k, k = 4, 5, \dots, |V|$

If a 3-tree is not found from the iteration for  $k = 3$  as described in the previous subsection, we then continue with the next main iteration in Fig.3 and look for a 4-tree. An example of such a case is the complete graph with seven nodes for which  $\delta = |E| - |V| + k = 21 - 7 + 4 = 18$  (see Fig.7). In general, this procedure can be continued for values of  $k = 4, 5, \dots, |V|$  (see Fig.3) until an appropriate  $k$ -tree is found. In the worst-case scenario, the procedure will continue until  $k = |V|$  is reached. In that case  $\delta = |E|$ , which means that all edges in the graph  $G$  must be marked with a dire.

To summarize the presentation of the proposed algorithm, the main result of this paper may be stated as follows: *The complementary set of edges of a  $k$ -tree corresponds to the minimum set of dires if (i) each edge of the complementary set has its vertices in different parts of the  $k$ -tree and (ii) no  $(k-1)$ -tree exists which satisfies the first condition.*

## IV. EXPERIMENTAL RESULTS

In this section we present our experimental results. Based on the proposed algorithm, a computer program is written in C++. Several networks are analyzed by running the program on a Linux machine with a 2.8 GHz Intel Quad processor and

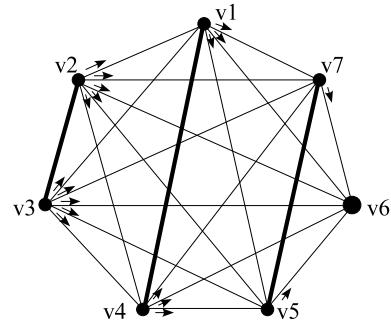


Fig. 7. The complete graph with 7 vertices is an example of graph for which  $\delta = |E| - |V| + k = 21 - 7 + 4 = 18$ .

2 GB memory. The program can be downloaded from [8]. The results are summarized in Table I and discussed below (entries denoted as “-” signify data are not available or not applicable).

### A. Small Systems

We first apply the proposed algorithm to the 8 bus system from [5] and to the 10 bus Petersen testcase from [4]. The network graphs of these testcases with the minimum number of dires found are shown in Fig.8.

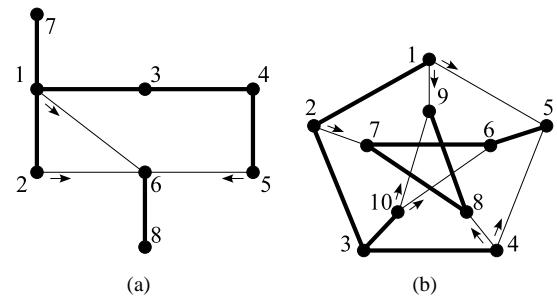


Fig. 8. (a) Network graph of the 8 bus system from [5]. (b) Network graph of the Petersen 10 bus system from [4].

### B. The IEEE 14 Bus System

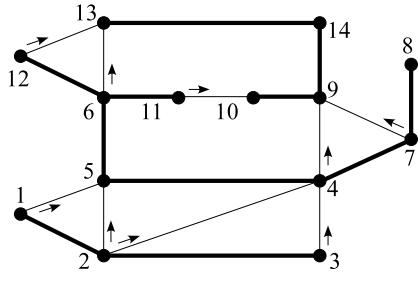
The graph of the IEEE 14 bus system is shown in Fig.9.a. It may be verified that a 2-tree does not exist for which each edge in the complementary set has vertices in different parts of the 2-tree. However, a 3-tree exists. The edges of the 3-tree are shown by thick lines in Fig.9.a. Note that such a 3-tree may not be unique. The sets  $S_a$ ,  $S_b$  and  $S_c$  are as follows:  $S_a = \{v1, v2, v3\}$ ,  $S_b = \{v4, v5, v6, v7, v8, v11, v12\}$ , and  $S_c = \{v9, v10, v13, v14\}$ . Thus the minimum number of dires is  $\delta = |E| - |V| + k = 20 - 14 + 3 = 9$ . The location and direction of the dires is shown in Fig.9.a. We note that the minimum number of 9 dires found by our algorithm is less than 10, which is the best previously reported result [4].

### C. Speeding-up Technique

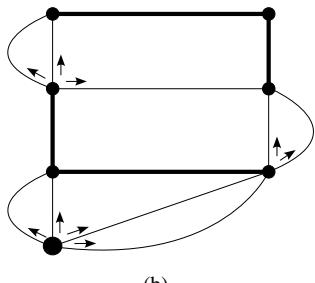
The complexity of the search process from Fig.4 can be reduced by first eliminating vertices of degree 1 and 2 from the graph. For example, using this simplifying technique the graph from Fig.9.a can be reduced to obtain the reduced

TABLE I  
COMPARISON OF THE PROPOSED ALGORITHM WITH THE ILP AND GENETIC ALGORITHMS STUDIED IN [4], [5].

Testcase			Proposed algorithm		ILP algorithm [4]		Genetic algorithm [5]	
	Num. vertices/edges of initial graph	Num. vertices/edges of reduced graph	Num. dires	CPU runtime [s]	Num. dires	CPU runtime [s]	Num. dires	CPU runtime [s]
8 bus	8/9	-	3	< 1	-	-	3	10
Petersen 10 bus	10/15	-	7	< 1	7	-	-	-
IEEE 14 bus	14/20	7/13	9	< 1	10	-	-	-
IEEE 30 bus	30/41	11/22	16	2.6	18	-	18	80



(a)



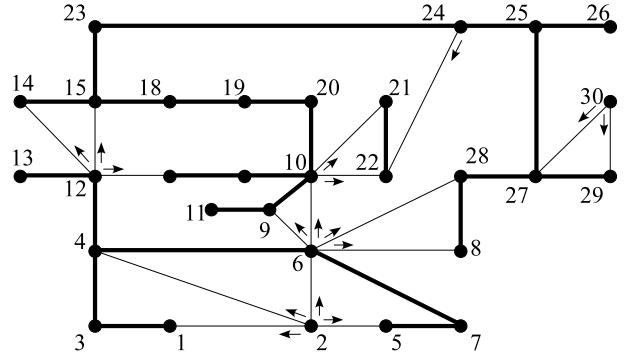
(b)

Fig. 9. (a) IEEE 14 bus network graph. (b) Reduced graph.

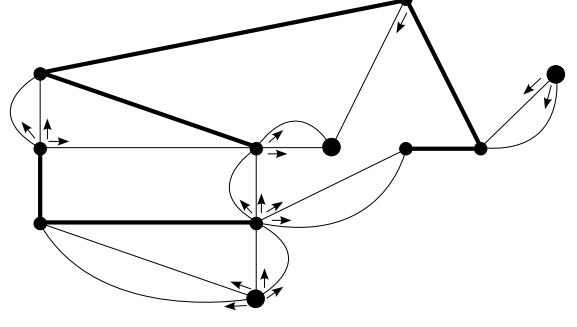
graph shown in Fig.9.b. Second, if an edge is in parallel with another edge (forming a loop), then we know we must place dires on both these edges. In other words, those two edges cannot be elements of a k-tree. Using these reduction techniques, the search process to find a k-tree is narrowed to fewer edges, which in turn will significantly reduce the computational runtime. The reduced graph from Fig.9.b has 7 vertices and 13 edges. Among these 13 edges, there are 8 edges that form loops of two edges in parallel and thus we know that these 8 edges must have dires placed on them. In other words, 8 out of the total minimum of 9 dires placed in the IEEE 14 bus system are located on edges that form loops of two edges as shown in Fig.9.b.

#### D. The IEEE 30 Bus System

The graph of the IEEE 30 bus system is shown in Fig.10.a. Using the reduction techniques discussed in the previous subsection the graph is reduced to 11 vertices and 22 edges (see Fig.10.b). Then, the proposed algorithm identifies the minimum break point set with 16 edges, which are also marked with dires in Fig.10.b. We note that the minimum number of 16 dires found by our method is less than 18 as the best previously reported result [4], [5].



(a)



(b)

Fig. 10. (a) IEEE 30 bus network graph. (b) Reduced graph.

### *E. Computational Runtime Discussion*

As seen in Table I the proposed algorithm has the shortest computational runtime. CPU runtimes were not reported in [4] but we suspect them to be longer than the CPU runtimes of the genetic algorithm studied in [5]. The proposed speeding-up techniques (discussed earlier) significantly reduce the computational complexity for the IEEE 30 bus system. However, because the proposed algorithm is an enumeration-based approach, for larger systems the computational complexity increases exponentially. To handle large systems, we propose a divide and conquer approach similar to the decomposition-based solution discussed in [4]. This is based on applying the proposed algorithm to individual partitions of the system (e.g., arising from natural, administrative, etc. reasons) and then adding additional necessary dires to form the (sub-optimal) break point set for the whole network graph. Because such an approach is not necessarily new, we do not propose to discuss it in detail in this paper.

## V. CONCLUSIONS

We proposed a new graph theory based algorithm for the determination of the minimum break point set for a meshed system. It is based on finding a k-tree whose complementary set of edges in the system graph corresponds to the minimum set of dires if (i) each edge in the complementary set has its vertices in different parts of the k-tree and (ii) no (k-1)-tree exists which satisfies the first condition. The proposed algorithm identified the minimum break point sets of 9 dires for the IEEE 14 bus system and of 16 dires for the IEEE 30 bus system. These results are better than the previously best reported values of 10 and 18 dires for the same systems.

## ACKNOWLEDGMENT

The authors thank the anonymous reviewers whose suggestions have strengthened the final presentation.

## REFERENCES

- [1] A.H. Knabel, "A standardized approach to relay coordination," *IEEE Winter Power Meeting Conference*, 1969.
- [2] M.J. Damborg, R. Ramaswami, S.S. Venkata, and J.M. Postforoosh, "Computer aided transmission protection system design, Part I: algorithms," *IEEE Trans. on PAS*, vol. PAS-103, no. 1, pp. 51-59, Jan. 1984.
- [3] V.V.B. Rao and K. Sankara Rao, "Computer aided coordination of directional relays: determination of break points," *IEEE Trans. on Power Delivery*, vol. 3, no. 2, pp. 545-548, April 1988.
- [4] R.K. Gajbhiye, A. De, and S.A. Soman, "Computation of optimal break point set of relays - an integer linear programming approach," *IEEE Trans. on Power Delivery*, vol. 22, no. 4, pp. 2087-2098, Oct. 2007.
- [5] H. Sharifian, H. Askarian Abyaneh, S.K. Salman, R. Mohammadi, and F. Razavi, "Determination of the minimum break point set using expert system and genetic algorithm," *IEEE Trans. on Power Delivery*, vol. 25, no. 3, pp. 1284-1295, July 2010.
- [6] M.N.S. Swamy and K. Thulasiraman, *Graphs, Networks and Algorithms*, John Wiley & Sons, New York, 1981.
- [7] V.V.B. Rao and V.G.K. Murti, "Enumeration of all trees of a graph," *Electronics Letters*, 1969.
- [8] PRO-DIRES tool, 2010, [Online]. Available: <http://venus.ece.ndsu.nodak.edu/~cris/software.html>



**Bapewara-Rao V. Vinnakota** received the Ph.D. degree in electrical engineering from the Indian Institute of Technology, Madras, India in 1970. He is a Professor in the Electrical and Computer Engineering Department, North Dakota State University. His research interests include circuit theory, graph theory, and logic design.



**Cristinel Ababei** (M'04) received the Ph.D. degree in electrical engineering from the University of Minnesota, Minneapolis, in 2004 and the M.Sc. and B.S. degrees in electrical engineering from the Technical University "Gheorghe Asachi" of Iasi, Romania. He is an Assistant Professor in the Electrical and Computer Engineering Department, North Dakota State University. Between 2004-2008, he worked for Magma Design Automation, San Jose, CA. His research interests include design automation of VLSI and FPGA circuits, analysis and optimization of power systems, and reconfigurable and parallel computing.