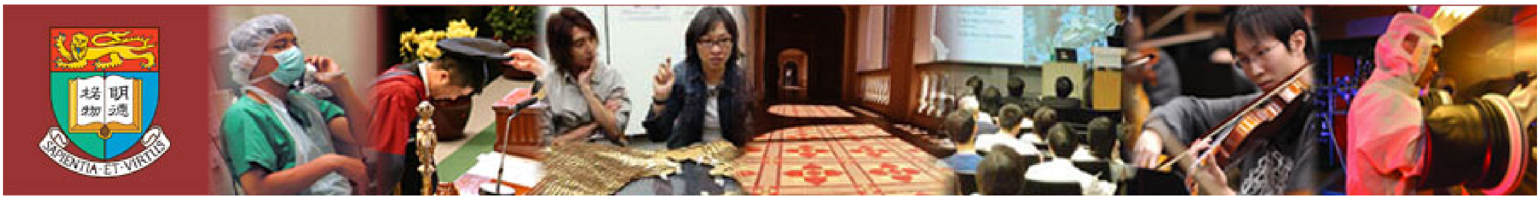


Title	An efficient graph partition method for fault section estimation inlarge-scale power network
Author(s)	Tianshu, B; Ni, Y; Shen, CM; Wu, FF
Citation	IEEE Power Engineering Society Winter Meeting, Ohio, USA, 28 January - 1 December 2001, v. 3, p. 1335-1340
Issued Date	2001
URL	http://hdl.handle.net/10722/46335
Rights	Creative Commons: Attribution 3.0 Hong Kong License

An Efficient Graph Partition Method for Fault Section Estimation in Large-Scale Power Network

Tianshu Bi, St. M. IEEE Yixin Ni, S.M. IEEE C.M. Shen Felix F. Wu, Fellow IEEE

Dept. of Electrical and Electronic Engineering
The University of Hong Kong
Hong Kong SAR

Abstract In order to make fault section estimation (FSE) in large-scale power networks use distributed artificial intelligence approach, we have to develop an efficient way to partition the large-scale power network into desired number of connected sub-networks such that each sub-network should have balanced working burden in performing FSE. In this paper, a new efficient multiple-way graph partition method is suggested for the partition task. The method consists of three basic steps. The first step is to form the weighted depth-first-search tree of the power network. The second step is to further partition the network into connected, balanced sub-networks. And the last step is an iterative process, which tries to minimize the number of the frontier nodes of the sub-networks in order to reduce the required interaction of the adjacent sub-networks. The proposed graph partition approach has been implemented with applications of sparse storage technique. It is further tested in the IEEE 14-bus, 30-bus and 118-bus systems respectively. Computer simulation results show that the proposed multiple-way graph partition approach is suitable for FSE in large-scale power networks and is compared favorably with other graph partition methods suggested in references.

Keywords graph partition, fault section estimation, large-scale power network

I. INTRODUCTION

Fault section estimation (FSE) aims at identifying the faulty elements in power network based on the information of the current status of the protective relays and circuit breakers, which is available from SCADA systems. As the first step to system restoration, FSE is of great importance in enhancing service reliability and reducing power supply interruption. FSE should be implemented quickly and accurately in order to isolate the faulty elements from the rest of the system and to take proper countermeasures to recover normal power supply. It is clear that on-line automatic FSE is significant and crucial to the restorative operations.

Many artificial intelligence techniques have found their use in solving the problem such as expert-system-based [1], fuzzy-set-based [2], artificial-neural-network-based [3], stochastical-optimization-based [4] and logic-based [5] approaches. However FSE of large-scale power networks still remains unsolved because of the large amount of information to be dealt with and the FSE speed and accuracy required. The FSE is even more difficult in cases with failure operations of relays and circuit breakers, or multiple faults at the same time.

Based on the idea of "divide and conquer", we suggest to use distributed artificial intelligence systems (DAIS) for FSE in large-scale power networks. The local nature of FSE makes this DAIS approach very attractive. The faulty element could only be identified according to the status of its main protection, neighboring back-up protection and the corresponding circuit breakers, which are all local signals. This makes DAIS very attractive in FSE. In addition, once the power network is expanded and some new elements are added, for distributed FSE only related sub-systems need to be changed to adapt this network expansion, while centralized FSE have to be totally reconstructed and retrained, which is extremely time-consuming.

Two issues have to be solved in implementing FSE using DAIS. The first issue is to partition the large-scale power networks into several sub-networks. The second one is to construct the corresponding FSE sub-systems of obtained sub-networks by independent distributed artificial intelligent subsystems. In this paper, the first issue is our concern. The second issue has been introduced in other two papers [6-7].

It is obvious that a good power network partition method is essential and crucial to the success of distributed FSE systems. According to the characteristics of FSE, a good power network partition means that any obtained sub-network should be a connected network; the calculation burdens of FSE of sub-networks should be balanced in order to improve the parallel calculation efficiency in a multi-processor system; and the number of elements on the frontier should be minimized to reduce the overlapping of different neighboring sub-networks.

If we denote the buses and the transmission lines in the power network by the vertices and the edges in a graph respectively, this power network partition problem can be easily modeled by the following graph partition problem: to partition the graph vertices into connected and balanced subsets according to the weights of the vertex under the constraint that the number of frontier nodes crossing the different subsets is minimized.

Since this graph partition problem appears in many practical applications and is a *NP*-hard problem [8], a number of efficient and effective heuristic algorithms have been developed for its solution. Most of these attempts are based on one of the following three basic approaches: Kernighan-Lin heuristics [9], Joseph W.H. Liu heuristics [10] and stochastic optimization based heuristics [11]. The common demerit of these methods is that none of them consider the special requirement of the FSE that any obtained sub-network must be connected. It is not easy to modify these methods to adapt this requirement.

In this paper, a new efficient multiple-way graph partition method is proposed for FSE in large-scale power networks. It consists of three basic steps. The first step is to form the weighted depth-first-search tree of the power network. The second step is to further partition the network into connected, balanced sub-networks. And the last step is an iterative process, which tries to minimize the number of the frontier nodes of the sub-networks in order to reduce the required interaction of the adjacent sub-networks. The proposed graph partition approach has been implemented with applications of sparse storage technique. It is further tested in the IEEE 14-bus, 30-bus and 118-bus systems respectively. For comparison purpose we realized and tested Joseph W. H. Liu graph partition heuristics too. Computer simulation results show that the proposed multiple-way graph partition approach is suitable for FSE in large-scale power networks and is superior over the previous method of graph partition heuristics.

II. PROPOSED GRAPH PARTITION APPROACH

The proposed multiple-way graph partition method consists of three basic steps, which will be described one by one in detail in this section.

A. Weighted Depth-First-Search Tree

Depth-First-Search (DFS) algorithm [13] is a standard technique of systematically exploring nodes in a graph. It serves as a fundamental tool in devising many efficient graph algorithms. For completeness, we give a quick review of the overall algorithm in Table 1.

Table 1. Depth-first-search algorithm

Let G be a graph with n vertices $\{x_1, x_2, \ldots, x_n\}$,

Step 1. Choose the vertex with the largest subscript, that is x_n, label it 1, and proceed to Step 2 with this vertex and label.

Step 2: Given a vertex x_i labeled k, if there exists the vertices adjacent to x_i which have not been labeled, then:

(a) Select the vertex with the largest subscript among these unlabeled adjacent vertices, assign to it the smallest unused label from the set $\{1, 2, \ldots, n\}$ and repeat Step 2 for this new vertex and its label;

else means that all vertices adjacent to this vertex x_i have been labeled, then:

(b) • If the label k of vertex x_i satisfies $k>1$, backtrack to the vertex from which you arrived at x_i at the time k was labeled and repeat Step 2 with this vertex and its label.

• If the label k of vertex x_i satisfies $k=1$, stop and the algorithm end.

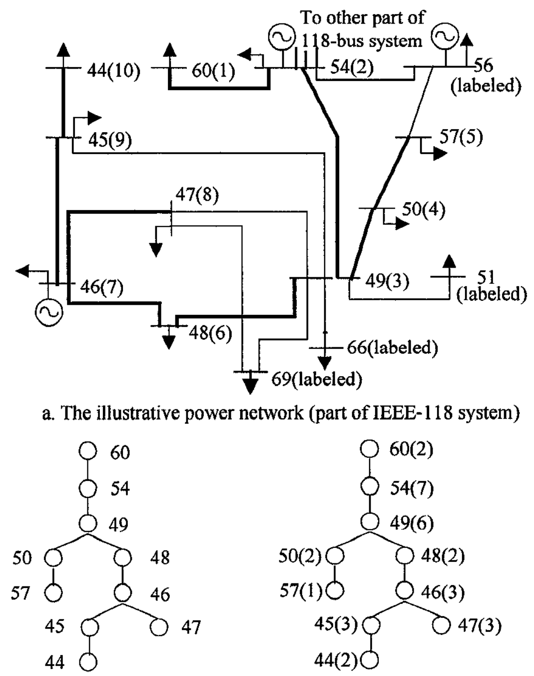
A simple power network is cut from IEEE 118-bus system as the illustrative example (Fig. 1a), in which the vertices are denoted by their subscripts and corresponding labels directly. When the algorithm terminates, those edges, which were used in the DFS (depicted by darker line in Fig. 1a), form a spanning tree [13] for the connected graph (Fig. 1b). This property is desired for our partition method because all the vertices would be explored systematically and any obtained sub-graph would be connected when the partition is carried out along the spanning tree. In addition, it should be pointed out that our partition method just requires a spanning tree as the basis of the partition and so the DFS algorithm can be replaced by other algorithms which could produce a spanning tree.

In order to make the calculation burdens of sub-graphs more balanced, we assign each vertex an integer, called weight, to form the weighted spanning tree. The investigation shows that the calculation burden of a sub-graph is mainly determined by the number of the involved fault estimation objects, hence the weight of a vertex is the number of the related fault estimation objects with the considered vertex.

Suppose Y_n is the admittance matrix of the given power network, we use the number of the nonzero elements of the upper diagonal matrix of Y_n as the weight of the corresponding vertex, which is denoted as $nodewt(x_i)$ of node x_i. Let $T[x_i]$ denote the set of nodes in the subtree rooted from node x_i. Then the total weights associated with this subtree will be:

$$wt(T[x_i]) = \sum_{j\in T[x_i]} nodewt(x_j) \tag{1}$$

The weighted DFS tree of the illustrative example is shown in Fig. 1c.



a. The illustrative power network (part of IEEE-118 system)

b. Depth-first-search tree

c. Weighted depth-first-search tree

Fig. 1 The illustrative example of weighted depth-first-search tree

B. Multiple-way graph partition algorithm

Suppose G is a graph with n vertices. Once the DFS spanning tree is formed, we could get the following basic terminology and concepts. If the root node of the spanning tree is x_n, the spanning tree could be denoted as $T[x_n]$. Let $x_i=parent(x_j)$ represent that node x_i is the *parent* of the node x_j, which means x_i is the neighboring node of x_j in the path from x_j back to the root node; on the contrast, x_j is called the *child* of the node x_i. A node, which has no children, is called leaf node. The length of node x_i is defined as the path length from node x_i back to the root node. Hence, the above weighted DFS tree in Fig. 1c could be denoted as *T[60]*, in which nodes *44, 47* and *57* are leaf nodes and their parents are *45, 46* and *50* respectively. For leaf node *44*, its path to root node *60* consists of nodes *45*, *46*, *48*, *49*, *54* and *60*, so the length of node *44* is 6.

For graph G with n vertices, let n_g be the number of the desired sub-graphs, then the objective of the proposed multiple-way partition algorithm is to divide the node set of G into n_g connected subsets under the constraint that the weight of each subset should be as close to $wt(T[x_n])/n_g$ as much as possible. Along the paths from leaf nodes to root node of the weighted DFS spanning tree, the proposed algorithm searches the partition points which satisfy the constraint. We described the proposed algorithm in detail in Table 2.

Table 2 Proposed multiple-way graph partition algorithm

Step 1: Denote the node set of G by S, the node sets of the sub-graphs by C_i, where i= 1, 2, …, n_g, and the temporary working set by *cw* Initially, set $S=\{x_1, x_2, \ldots, x_n\}$, $C_i=\varnothing$ and $cw=\varnothing$;

Step 2: Choose the leaf nodes, which have the maximum length, as the starting search points of the partition algorithm. Move these leaf nodes to the working set *cw* and delete them from S at the same time. In addition, suppose the maximum length of the leaf nodes is denoted by k and the pointer of the sub-graphs i=1;

Step 3: k=k-1;

Step 4: Find the different parents of the nodes in *cw*. For each parent node x_p, sort its child nodes $\{x_j \mid x_p = parent\,(x_j)\}$ in descending order according to their subtree weight $wt(T[x_j])$. Then test the child node x_j one by one by this sequence, if $T[x_j]$ satisfies:

$$|wt(T[x_p])-wt(T[x_n])/n_g|>|wt(T[x_j])-wt(T[x_n])/n_g| \quad (2)$$

then assign the subtree $T[x_j]$ as a sub-graph $C_i=T[x_j]$ and i=i+1. At the same time delete x_j and $T[x_j]$ from the working set *cw* and S respectively. If all the child nodes of x_p couldn't satisfy (2), delete all the child nodes of x_p from the *cw* and add the node x_p to *cw*. Repeat *step 4* until all the different parent nodes have been tested;

Step 5: Find the leaf nodes whose length equal k and add these leaf nodes to the working set *cw*;

Step 6: If k equals 0, then $S=\varnothing$ means all the nodes of G have been explored systematically, and so the algorithm terminates, else repeat *Step 3* to *Step 6*.

We still use the above simple power network (Fig. 1) cut from the IEEE 118-bus system as an illustrative example. Suppose the total weights *wt(T[118])* of IEEE 118-bus system is 297 and n_g=12, then the expected weights of each sub-network should be $wt(T[118])/n_g$=24.75. We use this as the measurement to perform partitioning task. For the weighted DFS tree in Fig. 1c, node *44* is the only node which has the maximum length k=6, so it's the starting search point and move it to *cw*. At the same time, perform *step 3* k=k-1=5. Then the parent x_p of *cw* is node *45*. This x_p only has one child, that is, node *44*. For node *44*, $|wt(T[44])-24.75|=22.75 > |wt(T[45])-24.75|=21.75$, eq. (2) (in Table 2) is not satisfied, which means that the parent is more approaching the expected weights of the sub-network. Hence, delete child node *44* from *cw* and add the parent node *45* to *cw*. In addition, find the leaf node *47* whose length is k=5 and move it to *cw* too. Now *cw={45 47}*, *S={46 48 57 50 49 60}* and k=5. Because of $k \neq 0$, continue the similar search process from *step 3* to *step 6* along the weighted DFS tree. When the search process encounters node *49*, that is *cw={49}* x_p=*{54}* *S={54 60}* and k=2, $|wt(T[49])-24.75|=2.75<|wt(T[54])-24.75|=4.25$, eq. (2) is satisfied, which means *T[49]* is the proper node set for one sub-network. Therefore, *T[49]={44 45 47 46 48 57 50 49}* makes up of one sub-network and *cw={54}* and *S={60}* is the new starting search point for the next sub-network. This multiple-way graph partition algorithm will terminate until all the nodes of IEEE 118-bus system have been explored. It can be seen that the proposed algorithm guarantees the obtained sub-networks are connected because the partition is carried out along the spanning tree and guarantees the calculation burdens are distributed evenly because all the partitions satisfy eq. (2).

C. Frontier reduction method

After the partition, an initial state of the frontier nodes crossing different sub-graphs is determined as well. We use the notation $Adj_G(x)$ to refer to the set of nodes adjacent to the node x in the graph G. We also extend this operator to include the adjacent set of a node subset, that is, for a subset W of nodes.

$$Adj_G(W)=\{u \mid u \notin W,\ u \in Adj_G(x),\ x \in W\} \quad (3)$$

When the graph G is clear from the context, we use Adj instead of Adj_G. In the same way, we use $Adj_G(x,U)$ or $Adj(x,U)$ to denote the adjacent nodes of x in U, that is, $Adj(x,U)= Adj(x)\cap U$. We also extend the notation to subsets, that is, for a subset W, $Adj(W,U)=Adj(W)\cap U$. For any obtained sub-graph C_i (i=1, 2, …, n_g), the initial value of its corresponding frontier node set is represented by F_i (i=1, 2, …, n_g) and defined as:

$$F_i=\{u \mid u \in Adj(C_i,U),\ U=\bigcup_{j>i} C_j\ ,\ i,j=1, 2, \ldots, n_g\} \quad (4)$$

We apply this definition to the illustrative example in Fig. 1 on the basis of the partition. Suppose C_1=*{44 45 47 46 48 57 50 49}* and then the initial value of its corresponding frontier node set is F_1=*{54 56 61 66 69}*. This is the original frontier node set to be reduced.

Consider a subset Y of any given frontier node set F_i, where $Adj(Y,C_i)\neq C_i$. The following simple but crucial results are stated.

Proposition 1: Suppose F_i is the frontier node set crossing C_i and U, then the set $\overline{F_i}=(F_i-Y) \cup Adj(Y, C_i)$ is the frontier node set of the two sets $\overline{C_i}=C_i-Adj(Y, C_i)$ and $\overline{U}=U\cup Y$.

Proposition 2: If $|Adj(Y,C_i)|<|Y|$, then $|\overline{F_i}|<|F_i|$.

The following heuristic method makes use of the above two observations and attempts to reduce the size of the frontier node sets one by one, in which the central issue is how to determine a subset Y of a given frontier node set F_i so that the size of $Adj(Y, C_i)$ is less than that of Y.

C.1 Background on bipartite graph and matching

The problem of finding a subset Y of the corresponding frontier node set with the desirable property is associated with a well-known combinatorial problem called *bipartite graph matching* [12]. We outline the necessary concepts and related terminology first.

A bipartite graph H is an undirected graph whose node set can be divided into two disjoint subsets I and J such that every edge has one endpoint in I and the other in J. it is customary to write H as (I, J, E). A edge subset M of E is called a matching of H if no two of the edges in M are adjacent, in other words, if there is no intersection node for any two edges in M. Furthermore if the vertex v of the graph H is the end vertex of some edge in the matching M then v is called M-saturated or we say that M saturates v. Otherwise v is M-unsaturated. The number of edges in M is called the size of the matching. If M is a matching in H such that every vertex of H is M-saturated then M is called a perfect matching. A matching M in H is called maximum if H has no matching M' with a greater number of edges than M has. Let M be a matching in H and let $E=E(H)$ be the edge set of H. An M-alternating path in H is a path whose edges are alternately in M and $E-M$, i.e., alternately in M and not in M. An M-alternating path whose origin and terminus are both M-unsaturated is called an M-augmenting path.

For a given large-scale power network, we could establish the bipartite graph H_i between any obtained sub-graph $\{C_i\}_{i=1}^{n_g}$ and its corresponding frontier node set $\{F_i\}_{i=1}^{n_g}$ by $(F_i, Adj_G(F_i, C_i), E)$. For example, we abstract the corresponding bipartite graph from Fig. 1a and redraw it in Fig. 2 to explain the above-mentioned concepts about bipartite graph. Two node sets, $I=F_I=\{54\ 56\ 61\ 66\ 69\}$, $J=Adj_G(F_I,C_I)=\{47\ 49\ 57\}$, and the edge set E crossing I and J construct a bipartite graph $H=(I, J, E)$. It's obvious that the set $\{\{69, 49\}, \{56, 57\}\}$ depicted by darker line in Fig. 2b is a matching of size 2 for H. With respect to this matching, the path $(47, 69, 49, 66)$ is an augmenting path, a special case of alternating path.

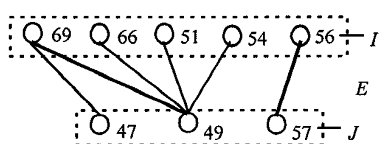


Fig. 2 The bipartite graph abstracted from Fig. 1a.

It should be pointed out that, if there is an augmenting path in H with respect to M, we could augment the matching M by this path as follows. For example, $(47, 69, 49, 66)$ is the augmenting path with respect to the matching $\{\{69, 49\}, \{56, 57\}\}$ shown in Fig. 2, remove the edges $\{69, 49\}$ from the matching M and replace them by $\{47, 69\}$, $\{49, 66\}$ in the augmenting path. This has the net effect of increasing the size of the matching by one. This operation is called a transfer along the augmenting path.

C. 2 Theorem and Hungarian method for frontier reduction

The following marriage theorem and its corollary relate bipartite graph and matching with the objective of this subsection, that is, find the smaller adjacent set of the subset Y of the given frontier node set F_i to make the frontier reduction.

<u>*Theorem 3*</u> (Hall's Marriage Theorem, 1935, [12]): Let H be a bipartite graph with bipartition $V= I\cup J$. Then H contains a matching that saturates every vertex in I if and only if

$$|Adj_H(Y)|\geq|Y| \text{ for every subset } Y \text{ of } I, \qquad (5)$$

<u>*Corollary 4*</u> ([12]): Let x be an unsaturated node in I with respect to the matching M. If there is no M-augmenting path with x in H, then there exists a subset Y containing the node x such that $|Adj_H(Y)|<|Y|$.

The Hungarian method [12] uses the concept of M-alternating tree to prove this corollary by a direct construction of the set Y. Let M be a matching in the bipartite H and let x_0 be an M-unsaturated vertex in I. Then the M-alternating tree rooted at x_o is defined as: $L_0, L_1, L_2, \ldots, L_{2j-1}, L_{2j}, \ldots$, where:

$L_0=\{x_0\}$; $j= 1, 2, 3, \ldots$;
For odd levels: $L_{2j-1}=Adj_H(L_0\cup\cdots\cup L_{2j-2})$;
For even levels: $L_{2j}=\{u \mid \{u,v\}\in M \text{ for some } u\in L_{2j-1}\}$ (6)

Since there is no augmenting path starting with the node x_0, there has no unsaturated node in the odd levels. This implies that every node in the odd level L_{2j-1} must have its mate in the even level L_{2j} and the alternating tree must end on an even level. Therefore, we have $|L_{2j}|=|L_{2j-1}|$. Define the set Y to be the union of all the even levels, including L_0. That is,

$$Y=L_0\cup L_2\cup\cdots\cup L_{2j} \qquad (7)$$

According to (6), $Adj_H(Y)$ is simply the union of the odd levels. Then we have $|Y|-|Adj_H(Y)|=1$. Therefore, the desired subset Y is obtained by (7) and the corollary is proved.

We could apply this Hungarian method to construct our frontier reduction algorithm. For the illustrative example in Fig. 2, suppose that $M=\{\{47, 69\}, \{49, 66\}, \{57, 56\}\}$ is the matching of the bipartite graph $H=(F_1, Adj_G(F_1, C_1), E)$. It can be seen that node 51 is a M-unsaturated node in I, so we could get the alternating tree rooted at node 51 by (6): $L_0=\{51\}$, $L_1=\{49\}$, $L_2=\{66\}$, and thus the expected subset of the frontier node set F_1 is $Y=\{51, 66\}$, which has a smaller adjacent set $Adj_H(Y)=\{49\}$. The same way, the frontier reduction algorithm is described systematically in Table 3.

Hence, for a given large-scale power network, the three steps described in these three subsections are performed in successive steps. At last, we could get the optimal solution, which partition the given network into connected and balanced sub-networks under the constraints that frontier node sets are minimized as much as possible. Once the large-scale power network is divided into solvable sub-networks, the artificial intelligence techniques could be used to solve FSE problem for each sub-network.

Table 3. Frontier reduction algorithm

Step 1: i=1;
Step 2: For the sub-graph C_i and its frontier node set F_i, construct the corresponding bipartite H_i. At the same time, initialize matching M=∅.
Step 3: For each node $x \in F_i$, generating the M-alternating tree rooted at the node x by (6). If an M-augmenting path starting from x is found, then augment the current matching M by this path; repeat perform *Step 3* until all nodes of F_i have been tested, then go to *Step 4*. else according to the alternating tree rooted at x, return the desired Y by (7). get the modified $\overline{C_i}$ and $\overline{F_i}$ based on the *Proposition 1*. break and restart this algorithm from *Step 1*.
Step 4: i=i+1, if i=n_g, then this frontier reduction algorithm terminates, else go to *Step 2*.

III. COMPUTER SIMULATION RESULTS

The proposed multiple-way graph partition method has been implemented with the application of sparse storage scheme. The sparse storage scheme only stores and operates nonzero elements and this could improve the calculation efficiency greatly.

IEEE 14-bus, 30-bus and 118-bus systems are used as the test systems respectively. Only IEEE 118-bus system is selected as an example to present here with n_g=12 the working process of the proposed graph partition method systematically. The weighted DFS tree of IEEE 118-bus system formed by the first step of the proposed method is shown in Fig.3, in which each node is denoted by x ($nodewt(x)$). Then the second step, i.e. the partition algorithm, is performed to get n_g=12 sub-networks, which are listed in Table 4. In addition, the initial frontier and the reduction results of the third step are displayed in Table 5.

Table 4. Obtained sub-networks of IEEE 118-bus system after partition

i=[1, n_g]	Obtained sub-networks C_i={x|$x \in C_i$}	wt (C_i)
1	2, 1, 3, 7, 6, 10, 9, 8, 4, 5	25
2	13, 11, 16, 117, 12, 14, 22, 21, 20, 18, 19, 15	29
3	33, 36, 35, 41, 42, 39, 40, 37, 34, 43	25
4	44, 45, 47, 46, 48, 57, 50, 49	22
5	55, 54, 53, 52, 51, 58, 56, 60, 59	26
6	25, 26, 63, 64, 61, 62, 67, 66, 65, 38, 30	25
7	113, 17, 31, 29, 28, 27, 115, 114, 32, 23	26
8	83, 84, 87, 86, 85, 88, 89, 90, 91	21
9	104, 103, 111, 112, 110, 109, 108, 105, 107,106	22
10	24, 72, 73, 71, 70, 74, 75, 69, 116, 68	26
11	93, 95, 94, 92, 102, 101, 98, 100, 99	25
12	78, 79, 81, 80, 97, 96, 82, 77, 76, 118	25

Table 5. The frontier node sets of IEEE 118-bus system

i=[1, n_g-1]	F_i before reduction	F_i after reduction
1	11, 12, 30	11, 12, 30
2	17, 23, 33, 34	17, 23, 33, 34
3	38, 44, 49	38, 44, 49
4	51, 54, 56, 66, 69	49, 56, 69
5	61, 62, 63	59
6	17, 23, 27, 68	17, 68, 25
7	24	24
8	82, 92	82, 92
9	100	100
10	77, 81, 118	77, 81, 118
11	80, 96	80, 96

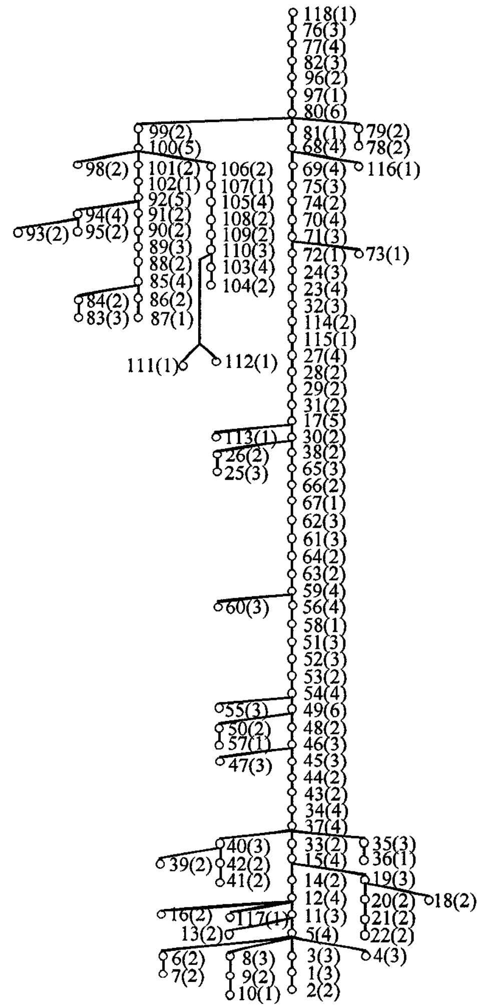


Fig. 3 The weighted DFS tree of IEEE 118-bus system

From this case, it can be concluded that the proposed multiple-way graph partition method works effectively and could satisfy all the requirements of FSE problem. IEEE 14-bus and 30-bus systems could get the similar results. Because of desired weights = $wt(T[x_n])/n_g$, we use max{ | $|C_i|$– desired weights|}/desired weights to denote the unbalance degree of the partition. Table 6 shows unbalance degree, the number of nodes on the frontier before reduction and after reduction for these IEEE test systems with different n_g's.

Table 6. The test results of IEEE 14-bus, 30-bus and 118-bus systems

	14-bus	30-bus		118-bus			
	(n_g=2)	n_g=2	n_g=3	n_g=2	n_g=3	n_g=8	n_g=10
Unbalance degree %	11.7	1.4	7.1	2.3	2	16.4	14.5
Before reduction (No.)	2	3	6	5	12	23	27
After reduction (No.)	2	3	5	5	10	21	24

The results in Table 6 prove that the proposed method could not only work effectively but also succeed in all the test power networks.

In addition, we realize Joseph W.H. Liu heuristics [10] by sparse storage technique too. Compared with the proposed graph partition method, Joseph W. H. Liu heuristics suffers three disadvantages. First, it only realizes 2-way partition if we require any obtained sub-graph must be connected. Second, it partition the network based on the elimination tree directly and not use weights to indicate the calculation burden. This is not helpful for getting the real balanced sub-networks for specific problem. In addition, the calculation of Joseph W. H. Liu heuristics is much more complicated than that of the proposed method, so its calculation efficiency is lower than that of the proposed graph partition method. Table 7 compares the calculation efficiency of 2-way partition between this method and our proposed graph partition method. The results show the proposed multiple-way graph partition method works more efficiently than Joseph W.H. Liu heuristics. Hence, our method could work effectively for FSE in large-scale power network.

Table 7. CPU time for 2-way partition by proposed method and Joseph W.H. Liu heuristics

Cpu time (s)	14-bus	30-bus	118-bus
Proposed method	0.05	0.11	0.28
Joseph W H Liu heuristics	0.05	0.16	2.15

IV. CONCLUSION

In this paper, a new efficient multiple-way graph partition method has been suggested and tested successfully for FSE in large-scale power network. It consists of three basic steps. The first step is to form the weighted depth-search-first tree of the power network. The second step is to further partition the network into connected, balanced sub-networks. And the last step is an iterative process, which tries to minimize the number of the frontier nodes of the sub-networks in order to reduce the required interaction of the adjacent sub-networks. The proposed graph partition approach has been implemented with applications of sparse storage technique. It is further tested in the IEEE 14-bus, 30-bus and 118-bus systems respectively. Computer simulation results show that the proposed multiple-way graph partition approach is suitable for FSE in large-scale power networks and is compared favorably with other graph partition methods suggested in previous work.

V. ACKNOWLEDGMENTS

The research is supported by National Key Basic Research Special Fund (No. G1998020305), P.R. China, to whom sincere acknowledgement are expressed.

VI. REFERENCES

[1] C. Fukui and J. Kawakami, "An expert system for fault section estimation using information from protective relays and circuit breakers", IEEE Trans. Power Delivery, Vol. 1, No. 4, October 1986, pp. 83-90.

[2] Hyun-Joon Cho and Jong Keun Park, "An expert system for fault section diagnosis of power systems using fuzzy relations", IEEE Trans. Power Systems, Vol.12, No.1, February 1997, pp.342-348.

[3] Z.E. Aygen, S. Seker, M. Bagriyanik, F.G. Bagriyanik and E. Ayaz, "Fault section estimation in electrical power systems using artificial neural network approach", 1999 IEEE Transmission and Distribution Conference, 1999, pp. 466-469.

[4] C.S. Chang, L. Tian and F.S. Wen, "A new approach to fault section estimation in power systems using ant system", Electric Power System Research, 1999(41), pp. 63-70

[5] T.S. Sidhu, O. Cruder and G.J. Huff, " An abductive inference technique for fault diagnosis in electrical power transmission networks", IEEE Trans. Power Delivery, Vol.12, No.1, January 1997, pp.515-522.

[6] Tianshu Bi, Yixin Ni, C M. Shen and F. F. Wu, " A novel ANN fault diagnosis system for power systems using dual GA-loops in ANN training", IEEE power engineering society 2000 summer meeting, 2000, pp.

[7] Tianshu Bi, Yixin Ni, C. M. Shen, F. F. Wu and Q. X. Yang, " A novel radial basis function neural network for fault section estimation in transmission network", APSCOM' 00, 2000, pp.

[8] M.R. Garey and D.S. Johnson, "Computer and interactability: a guide to the theory of NP-completeness", W.H. Freeman&Co., 1979.

[9] B.W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs", Bell Syst. Tech. J., Vol.49, February 1970, pp. 291-307.

[10] Joseph W.H. Liu, "A graph partitioning algorithm by node separators", ACM Trans. On Mathematical Software, Vol.15, No.3, September 1989, pp. 198-219.

[11] L. Tao and Y.C. Zhao, "Effective heuristic algorithms for VLSI circuit partition", IEE Proceedings-G: Circuits, Devices and Systems, Vol. 140, No. 2, April 1993, pp.127-134.

[12] John Clark and Derek Allan Holton, "A first look at graph theory", World Scientific, 1991, pp. 121-155.

[13] Edgar G. Goodaire and Michael M. Parmenter, "Discrete mathematics with graph theory", Prentice Hall, 1998, pp. 451-463.

VII. BIOGRAPHIES

Tianshu Bi received her B. Eng, and M. Eng degrees from North China Electric Power University, P. R. China, in 1994 and 1997 respectively. She is currently a Ph. D. candidature at the Dept. of E.E.E., the University of Hong Kong. Her research area includes power system operation, protective relay and the application of artificial intelligence in power systems.

Yixin Ni received her B. Eng., M. Eng., and Dr. Eng. from Electrical Engineering Department, Tsinghua University, P. R. China in 1968, 1981 and 1983 respectively. Her research interest is in power system modeling, simulation, stability and control, and power electronics applications in power systems. She was a professor of Tsinghua University and is now with the University of Hong Kong.

C.M. Shen received his B.Sc. (Eng.) and M.Sc. (Eng.) degrees in EE from the University of Hong Kong (HKU) in 1963 and 1965 respectively and his Ph.D. degree in EE from Queen Mary College, University of London in 1969. Since then he has joined the HKU. His research interest is in Load flow, fault analysis, state estimation and stability and control of power systems. He is an executive committee member of the Specialized Section in Power, IEE Hong Kong Centre.

Felix F. Wu joined the University of Hong Kong as Chair professor of Electrical Engineering in September 1995 and is now Pro-vice-chancellor of HKU. Prior to that, he was Professor and Vice Chairman, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, the same institute where he received his Ph.D. degree. His research interests are in power system planning and operation, including economics and reliability in system planning, real-time security assessment, and design of energy management systems and distribution automation. Recently he has been involved in the design of industry restructuring and electricity pricing.