An Improved Fault Diagnosis Method for Hierarchical Graph with Radar Systems

Huijie Zhang[1], Shulin Tian[1], Zhen Liu[1], Bin Long[1], Quan Zhou[1]

1 *School of Automation Engineering*
University of Electronic Science and Technology of china
Chengdu 610054, China

Abstract — **Fault diagnosis based on graph theory has been widely studied. Matrix graph representation is commonly used as the basic data structure, but this technique is time consuming because it requires numerous matrix multiplications. As such, applying traditional graph algorithms into large-scale complex electronic systems is difficult because of the huge time requirement and space complexity of matrix operations. In this study, a new hierarchical graph algorithm was established to improve the efficiency of fault diagnosis. The algorithm was then tested in Radar systems. First, a new indegree representation was proposed to reduce data storage space. Second, a new hierarchical graph searching strategy of indegree diffusion was used to avoid complex operation. Finally, the entropy-based degree of node was applied to improve the speed of graph search. The proposed method was validated through experiments using a radar system. Results demonstrated the efficiency and simplicity of the proposed algorithm in hierarchical graph of fault propagation; hence, the algorithm is important for fault diagnosis of electronic systems.**

Keywords - Fault diagnosis; Hierarchical graph; Entropy

I. INTRODUCTION

Fault diagnosis is used to ensure the stability and safety of modern systems 0, such as power and electronic systems. Given the increasing complexity of these systems, scholars have focused on developing efficient fault diagnosis methods 0. The traditional hierarchical graph strategy is widely used in reachability graph-based fault diagnosis. In the process, matrix representation is generated from adjacency matrix, which requires several matrix computations. However, this method is unsuitable in cases where the size of the electronic system becomes large because of rapid market demand. Applying traditional graph hierarchal algorithms to large-scale electronic systems is difficul; hence, efficient methods must be developed to solve this limitation.

Hu 0 used equivalent-input-disturbance approach to propose a hierarchical model for fault diagnosis of power systems. Zhou [1] proposed an innovative ontology-based fault path to analyze transient fault propagation effects on networked control systems. Lu [2] applied fuzzy theory to a large radar system to satisfy the requirements of real-time accuracy and multi-target diagnosis. Fault diagnosis can be performed through various techniques, such as graph theory, computational intelligence, multi-agent technology, and optimization techniques. Hierarchical fault diagnosis is one the most popular methods based on graph model. Most existing studies are based on typical graph theory, which is suitable and efficient for large-scale complex electronic systems. Nevertheless, the performance and applicability of hierarchical fault diagnosis for large complex electronic systems should be improved.

II. STATE-OF-THE-ART

Fault diagnosis based on graph theory has been widely studied. Scholars have improved the precision and applicability of large-scale complex electronic systems. Yang [3] used a hierarchical model based on signed directed graph (SDG) to improve the efficiency of graph search. Barua [4] proposed and applied a hierarchical fault diagnosis framework and methodology based on fuzzy rule in satellite formation flight. Yuan [7] developed a simulation model based on directed fault propagation graph. Zhang [8] proposed a directed cyclic graph and joint probability distribution. Moreover, numerous works have introduced entropy into fault propagation diagnosis. Jiang [9] used normalized mutual information to identify correlated metrics without detecting errors in the system. Hierarchical fault diagnosis is one of the popular methods adopted by researchers. Bauer [10] employed transfer entropy and proposed a method to identify the direction of fault propagation. Guo [11] proposed a fault isolation method based on estimation error by entropy to solve multiple faults in nonlinear, non-Gaussian systems. Kodali [6] used diagnostic matrix to isolate faults after dividing the fault universe of the cyber-physical system in a hierarchical manner. Hierarchical and matrix methods are also used in single-fault and multi-fault diagnosis [13]. However, most of state-of-the-art works are based on traditional graph theory [16] and focused on determining new methods and developing new functions or factors to satisfy different diagnosis requirements of different systems. The traditional hierarchical graph strategy can be generated from adjacency matrix based on reachability matrix but requires huge matrix computation time. This process is also unsuitable for large and complex electronic systems. Moreover, applying traditional graph hierarchal algorithms to large electronic systems is difficult. After establishing the hierarchical model of the electronic system, traditional graph search presents the advantage of the hierarchical structure of the graph, rather than the advantages of the full graph. Hence, hierarchical strategy is important to improve the electronic system performance.

In this paper, an indegree representation hierarchical algorithm is proposed to improve the efficiency of fault propagation and diagnosis of large-scale complex electronic systems. The remaining parts of the paper are organized as follows. Section 3 describes the methodology of the indegree representation of the hierarchical algorithm for system-level fault diagnosis. Section 4 presents the experiments performed to evaluate the performance of the model as well as the results, analysis, and discussion. Conclusion is drawn in Section 5.

III. METHODOLOGY

3.1. Hierarchical graph strategy

The relationship of nodes in an electronic system, whether as an SDG or a directed graph, would be expressed by adjacency matrix X. If a directed edge exists from node i to j, then $X(i,j)$ is assigned with a value of "1;" otherwise, it is assigned with "0." In a directed graph, $X(i,j)$ is assigned with the value of the edge weight. If nodes i and j are not related, then $X(i,j)$ =0. The hierarchical diagraph can be generated from the adjacency matrix in the two main stages, the first of which is generating reachability matrix R from adjacency matrix X:

$$X = \left(X + X^2 + \cdots + X^N\right)^{\#} \tag{1}$$

Where N is the maximum reachable path length in the graph. When $X^N = 0$, the computation is terminated immediately. The Boolean equivalent operation # is defined as follows:

$$A^{\#}(i,j) = \begin{cases} 0, ifA(i,j) = 0 \\ 1, fA(i,j) \neq 0 \end{cases} \tag{2}$$

After generating reachability matrix R, the algorithm for generating the hierarchical graph is shown as follows:

Step 1: For each node in the graph, find the reachability set Ri, which consists of all nodes reachable from vi. Then, find antecedent set Ai, which consists of all nodes where vi is reachable. Thus, we can obtain $R_i = \{v_j \in V \mid r_{ij} = 1\}$ and $A_i = \{v_j \in V \mid r_{ji} = 1\}$.

Step 2: Compute the basic level (level 1) nodes as:

(i) Find the intersection of reachability and antecedent sets for all nodes.

(ii) If the intersection set for node vi is the same as the antecedent set, then vi is a level 1 node (i.e., L1). Thus,

$$L_1 = \{v_i \mid R_i \cap A_i = A_i\} \tag{3}$$

Step 3: Compute level k nodes ($k \neq 1$) by removing nodes assigned to high levels. Remove the edges connected to these nodes to obtain a new component digraph. Then,

(i) Find the new reachability and antecedent sets, and repeat Step 1.

(ii) Repeat the steps until all nodes are traversed. New graphs will be obtained, namely, $L_k = \{v_i \mid v_i \in V - L_1 - L_2 - L_{k-1}\}$ and $(R_i \cap A_i = A_i)$, for i = 1,2,3…n.

The graphs cannot be easily applied to large electronic systems because of the huge time requirement and complexity of matrix multiplication at $O(n^3)$, where n is the number of nodes in the electronic system. When ignoring the time consumption of matrix addition and Boolean equivalent operation, a total of N+M times matrix multiplication exists, where M is the number of layers of the hierarchical diagraph. An enormous amount of time is consumed in this stage if millions of nodes are present in the electronic system. Thus, the traditional hierarchical graph strategy is inapplicable for large-scale complex electronic systems.

3.2. Graph search strategy

Inverse direction search is the key operation in directed graph and SDG fault propagation diagnosis. Algorithms in most studies merely traverse every possible inverse direction path without priority. However, each node in the graph exhibits unique and important effect on path search, which can be used to improve the performance of fault propagation diagnosis.

3.3. Indegree graph representation

We represent a graph using a matrix similar to that in Definition 1, where the rows and columns represent vertex numbers, and the value is the weight. If the value is infinite, the two vertexes are not connected. However, this matrix representation method exhibits several shortcomings. First, this representation is not space efficient. If N vertexes and E edges exist, the size of the matrix will be N^2. However, the number of edges in most cases is significantly smaller than N^2. This finding indicates that most elements of the matrix will be infinite in case of matrix representation. Obviously, these records would consume unnecessary space. Second, this representation presents increased complexity in computation. For example, if we want to search for a particular vertex in adjacent vertexes, we must examine the entire row and column in the matrix for this particular vertex, which has a total of 2N searches. However, in reality, we only consider values that are not infinite. All operations on infinite values are therefore unnecessary. Finding adjacent vertexes is a common operation in path-finding algorithms. Additional complexity derived from this method will reduce the efficiency of all path-finding algorithms. In this regard, scholars proposed the use of a link to improve the search method. A linked list exhibits the advantages of a dynamic structure, namely, fast addition and deletion; however, the access speed is not as fast as that of matrix representation.

Definition 1: Degree representation: For a graph with N number of vertexes, the largest indegree is m. The relationship matrix is $V(N(m+2))$, and the weight matrix $W(N,M)$ is the indegree representation of the graph:

$$V = \begin{vmatrix} V_s & degree & V_{in1} & \dots & V_{inm} \\ i & m & j_1 & \dots & j_m \end{vmatrix}$$

$$W = \begin{vmatrix} weight(in1) & weight(in2) & \dots & wight(inm) \\ x_{ij1} & x_{ij2} & \dots & x_{ijm} \end{vmatrix}.$$

Indegree is relative to outdegree, and the outdegree representation is similar to the indegree representation. Thus, only indegree will be used in the following discussion. This definition is based on Theorem 1. A given graph always has a unique degree representation

Proof: We can use the following matrix to represent any given graph:

$$X = \begin{pmatrix} 0 & x_{12} & \cdots & x_{1m} \\ x_{21} & 0 & \cdots & x_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix},$$

Where x_{ij} is the weights from vertex i to vertex j, and 0 represents the lack of connection from i to j. For matrix X, we ignore all 0 elements before they are transferred to the corresponding sparse matrix:

$$X' = \begin{vmatrix} V_s & V_e & weight \\ 1 & 2 & x_{ij} \\ \cdots & \cdots & \cdots \\ j & i & x_{ji} \end{vmatrix} \quad X_{ji} \neq \infty .$$

Transfer this sparse matrix into a new form. Group all ending vertexes to form a new matrix, which is the same as the indegree representation. Moreover, the transfer is an equivalent process, in which the property of the indegree representation method is the same as the original matrix representation.

Indegree or outdegree representation exhibits improved performance than matrix representation and linked list. If the average degree is $\overline{m}$, then space complexity is the sum of V and W table: $N\left(2\overline{m}+2\right)$. Accessing adjacent vertexes is a basic step in any path search algorithm and significantly influences the time performance. For indegree representation, every row in V table corresponds to the indegree adjacent vertex. Time complexity is denoted as $\overline{m}$. Compared with other representation methods, the space complexity of indegree representation is less if $\overline{m} < \frac{N}{2} - 1$. A detailed comparison is shown in Table I. In real large-scale graphs, such as maps or electronic system graph, $\overline{m} << N$ usually exists. Hence, the space complexity of this method is significantly reduced.

TABLE I TIME COMPLEXITY COMPARED WITH OTHER GRAPH REPRESENTATION

Method	Memory	Cost of indicating neighbors
Matrix	N*N	N
Linked List	$4*N*\overline{m}$	$2*\overline{m}$
IN	$N*2*\overline{m}+2$	$\overline{m}$

The initiation of indegree (outdegree) representation is simple. Tables V and W can be easily initialized by traversing every edge in the graph based on Definition 1.

3.4. Indegree representation of hierarchical algorithm

Basing on the indegree graph representation, we can obtain the hierarchical graph without reachability matrix R. The new hierarchical graph algorithm is shown as follows:

Initiation: Initiate traverse variable *layerTop*=1, *LayerBottom* = N and copy the indegree and outdegree information of the indegree representation matrix Vi to Vi' and Vo to Vo'.

Step 1. Traverse every node in Vi' and find nodes with indegree m=0; place these nodes in *layerTop and* L_i. Then, increase *layerTop* by 1.

Step 2. Remove all nodes in L_i from Vi' and Vo', and then traverse every node in L_i. Afterward, determine outdegree adjacent nodes and reduce their indegree *m* by 1 in Vi'.

Step 3. Repeat Steps 1 and 2 until $L_i = \phi$.

Step 4. Repeat Step 1 to 3 based on outdegree, and reduce *LayerBottom* by 1 instead of increasing *layerTop*.

Step 5. (Ring detection) Select one node j from the remaining nodes in Vi' and access one of its indegree adjacent nodes, which are also located in the remaining Vi'. Access the indegree adjacent nodes literately until j is accessed again.

Step 6. Place all these nodes in the below layer until one of their adjacent nodes has been established.

Step 7. A cyclic node set is found until j is reached again. Place this node set into layer L_i and remove these nodes from Vi'. If a node exists with 0 indegree or outdegree, then record such node to the high level.

Step 8. Repeat Steps 4 to 6 until $G' = \varphi$.

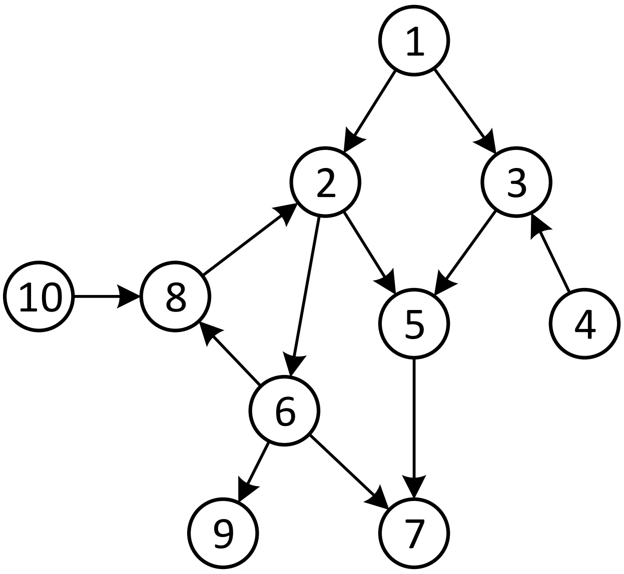


Figure 1. Example in literature [2].

The example in literature [2] is used to explain the algorithm. First, the indegree representation matrix V is shown in Table II.

TABLE II DEGREE REPRESENTATION FOR FIGURE 2.

1	0		
2	2	1	8
3	2	1	4
4	0		
5	2	2	3
6	1	2	
7	2	5	6
8	2	6	10
9	1	6	
10	0		

1	2	2	3	
2	2	5	6	
3	1	5		
4	1	3		
5	1	7		
6	3	7	8	9
7	0			
8	1	2		
9	0			
10	1	8		

In the first step, the indegree value of node1, node 4, and node 10 is 0 because these nodes are in the first layer. After performing Step 2, nodes 1, 4, and 10 are removed. The indegree value of nodes 2 and 8 are reduced by 1, and that of node 3 is reduced by 2. Thus, V is transferred to Table III.

TABLE III REMAINING VI'AND VO'AFTER STEP 2.

2	1
3	0
5	2
6	1
7	2
8	1
9	1

2	2
3	1
5	1
6	3
7	0
8	2
9	0

The indegree of node 3 is 0. Thus, node 2 is in the second layer. Repeating the steps will result in nodes 1, 4, and 10 in the first layer and 3 in the second layer, as shown in Table IV.

TABLE IV REMAINING VI'AND VO'AFTER STEP 3

2	1
5	2
6	1
7	2
8	1
9	1

2	2
5	1
6	3
7	0
8	2
9	0

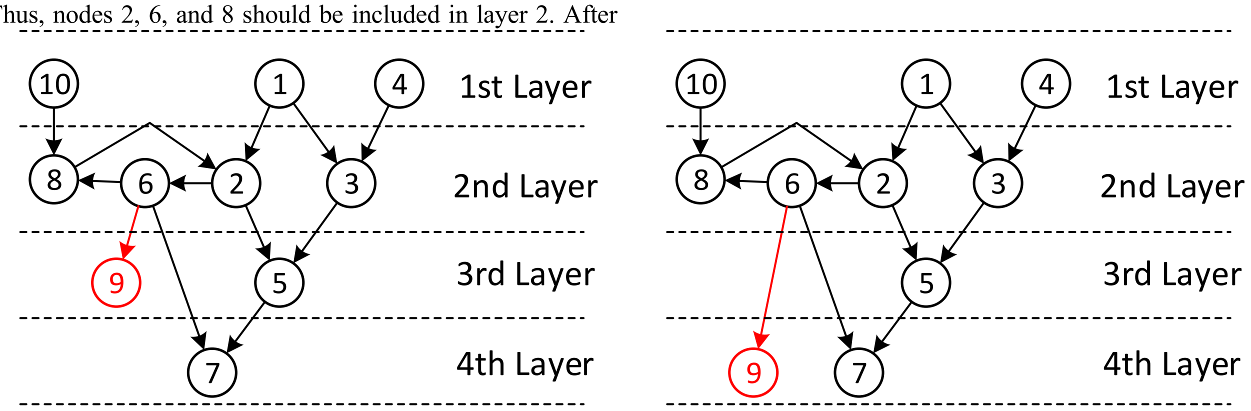
The bottom layer based on outdegree in Step 4 is performed. The results are nodes 7 and 9 in layer 10 as well as node 5 in layer 9. The result is shown in Table V.

TABLE V REMAINING VI'AND VO'AFTER STEP 4.

2	1
6	1
8	1

2	1
6	1
8	1

In Steps 5 to 8, the indegree adjacent nodes of nodes 2, 6, and 8 is nodes 1 and 10, both of which are in the first layer. Thus, nodes 2, 6, and 8 should be included in layer 2. After reorganizing the layers, the final hierarchical graph presenting the comparison of the result in Literature [2] is shown in Figure 1(b). In Step 1, we need to access N_i node to check the top-layer nodes, where N_i is the nodes in layer i. In Step 2, we need to remove N_i nodes and access N_i nodes to reduce the indegree number. Time complexity in the following steps is related to the result of the earlier steps, thereby increasing the difficulty of calculating total time complexity. In worst cases, only one node will be checked in each iteration, and the total time complexity could be $O(n+(n-1)+(n-2)+...)=O(n^2)$. Even in worst cases, this new method will present lower time complexity than the traditional method.



(a) Hierarchical graph in literature (b) Hierarchical graph using the proposed method

Figure 2. Hierarchical graph result comparison.

3.5. Entropy-based graph search

The original definition of entropy [6] of a discrete random variable X is shown as:

$H(X)=-\sum_i P(x_i)\log_b P(x_i)$

Where $P(x_i)$ is the probability mass function for x_i, and b is the base of the logarithm used. The common values are 2, 10, or Euler's number e; hence, the entropy values show the amount of information acquired from the observation. However, H(X) will be fixed after generating the value of X. This complex calculation formula will not be used in the present paper because it will increase the time complexity of the algorithm. Instead, we consider entropy as the representation of the importance of information. Therefore, we initially categorize information in any graph into two types:

Edges: Edge connects vertexes in the graph, which contains necessary information, such as vertexes, direct, and weight. We can simply store the edges to represent any graph.

Vertexes: Information of a vertex includes the unique numbers of the vertex, indegree, and outdegree.

In traditional SP algorithms, the information of the indegree or outdegree adjacent to a vertex is only used to traverse all possible paths. For example, the basic approaches for SP problems are breadth-first search and depth-first search, whereas traditional method searches the adjacent vertex without priority. For example, Dijkstra used greedy strategy to determine the vertex that should be visited first. New intelligent algorithms based on iterative strategies are used to obtain the optimal solution. In fact, we can obtain additional information from the indegree or outdegree adjacent to vertexes and improve the selection strategy

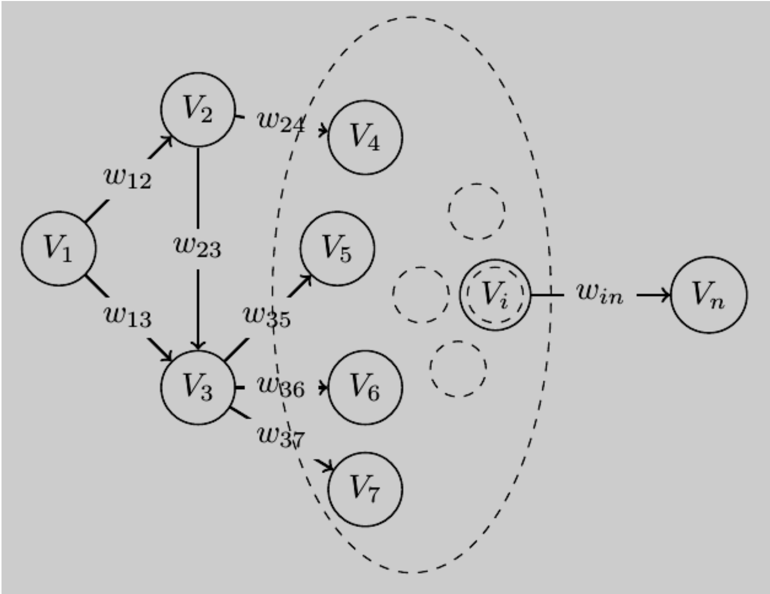


Figure 3. Entropy information in a graph.

For example, we assume that we need to obtain the shortest path from V_1 to V_n and select either V_2 or V_3 as the next step in our search. In traditional search algorithms, the decision between V_2 and V_3 is usually arbitrary. The diagram shows that V_3 has four outdegree adjacent vertexes, whereas V_2 has only 1. Therefore, of the five paths from V_1 to V_n, four could pass through V_3. When the number of outdegree adjacent nodes lacks information, the probability of V_3 and V_2 may be equal. However, with this given information, V_3 is obviously a good candidate. In conclusion, the adjacent vertex with the maximum outdegree should be searched first.

However, in most cases, vertexes in graphs do not only have outdegree adjacent vertexes but also indegree vertexes. If a vertex is searched before its indegree vertexes, redundant operations will occur, which can be proven using the case in Figure 4. Given the observation of V_1 and V_2, V_3 should be visited first because it has more outdegree vertexes. However, if $w_{12} + w_{23} < w_{13}$, all the visited vertexes affected by V_3 should be recalculated after V_2 is checked. To avoid these redundant operations, we need to visit the vertexes with less unchecked indegree vertexes.

Weight information is required to solve the contradiction problem between indegree and outdegree information. For example, if $w_{13} \le w_{12}$, V_3 is obviously a food candidate for first search because of lack of chance that $W_{12} + W_{23} < W_{13}$ in a positive weight graph. If $w_{12} \le w_{13}$, V_2 should be visited first because of the high probability that W_{13} is higher than $w_{12} + w_{23}$. Only when $w_{12} \approx w_{13}$ shall we need to adjust the selection strategy by indegree and outdegree information.

If the graph is sparse, the possibility that redundant operations will occur is low. Thus, we can reduce the factor of indegree information. Otherwise, we will need to reduce the factor of outdegree information. In conclusion, all the information based on the graph properties shall be combined to obtain the selection strategy.

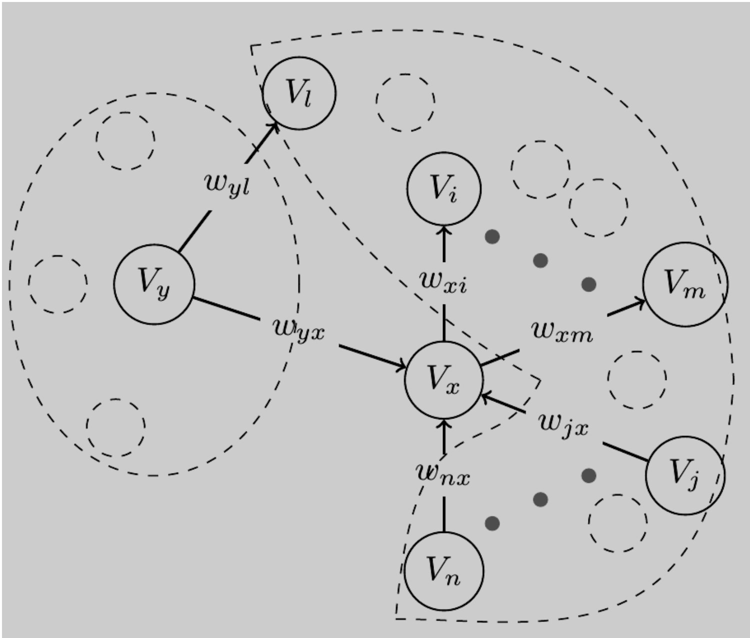


Figure 4. Entropy calculation.

In conclusion, the information observed from the adjacent vertexes can be used for improving the strategy to determine the vertex that should be visited first. Based on entropy theory, three types of information are combined to form the "entropy."

$$E(x) = H_0(x) + k_i H_i(x) + k_w H_w(x) \quad (4)$$

$$H(x) = \sum_i \frac{1}{w_1} \quad (5)$$

$$H_w(x) = w_x \quad (6)$$

Where w represents the weights that have been normalized. Thus, $0 < w \leq 1$,$0<W<1$; where k is the weight factor related to the property of graphs. Assuming that the average outdegree in the graph is $\overline{m_{out}}$, then the values of each part are determined as follows:

$H_0(x)$ is the "entropy" based on outdegree information. The smaller the weight of the outdegree vertex, the higher the possibility of the vertex will be in the shortest path. Therefore, we add all the outdegree inverse weight. For example, the outdegree entropy

$H_0(x)$ of vertex x in Figure 4 is

$$H_0(x) = \frac{1}{w_{xi}} + \cdots + \frac{1}{w_{xm}} . \quad (7)$$

$H_i(x)$ is the "entropy" based on indegree information. Indegree would cause redundant operations, especially when the weight is small. Hence, we add all weight information:

$$H_i(x) = \frac{1}{w_{jx}} + \cdots + \frac{1}{w_{nx}} . \quad (8)$$

$H_w(x)$ is the "entropy" based on the weight information of the vertex itself. Only one edge connects one vertex to one of its adjacent vertexes. Hence,

$$H_w(x) = \frac{1}{w_{jyx}} , \quad (9)$$

Where k_i is the factor of indegree "entropy." The higher the $H_i(x)$, the higher the chance that redundant operations will occur. This finding proves that k_i is a negative value. If we want to avoid redundant operations, then absolute value should not be smaller than the maximum outdegree m_{out} . However, if we can confirm the presence of few loops in the graph, k_i could thus be increased. Moreover, outdegree "entropy" can expedite the algorithm to determine a possible path after increasing k_i . Thus, outdegree "entropy" plays an important role in the formula. In the present paper, we used random graphs for the experiments. Thus, we cannot predict the results of the graph. The average value was used for this factor:

$$k_i = m_{out} \quad (10)$$

Where k_w is the factor of weight "entropy." The smaller the weight is, the higher the chance that the vertex will be in the shortest path. If the total weight of the requested shortest path is lower than the average weight of paths, we should ensure that $k_w > (1 + k_1)$. However, if $k_w >> (1 + k_1)$, the total entropy would be simplified as $E(x) = H_w(x)$. In this condition, the algorithm would be the same as the Dijkstra algorithm because both of them are based on greedy strategy. However, we cannot ensure that the total weight of the requested shortest path is small. Thus, we use $k_w = (1 + k_1)$ to ensure that every part in Formula (4) will play its role in the shortest path search.

In conclusion, the role of the first part $H_o(x)$ in (4) is to determine any possible path as fast as possible. The role of the second part $H_i(x)$ is to avoid redundant operations as much as possible. The role of the last part $H_w(x)$ is to limit the search range to the greatest extent. The factors of k_i and k_w are required to obtain the adjusted factors with different graphs. The first part $H_0(x)$ plays an important role in large graphs, whereas the last part is vital when the graph is small. If the structural complexity of graphs is known, second part $H_i(x)$ will be adjusted to speed up the search algorithm by reducing redundant operations.

IV. RESULT ANALYSIS AND DISCUSSION

Figure 5 shows the radar system after preprocessing. We initially obtain the indegree representation of the radar systems.

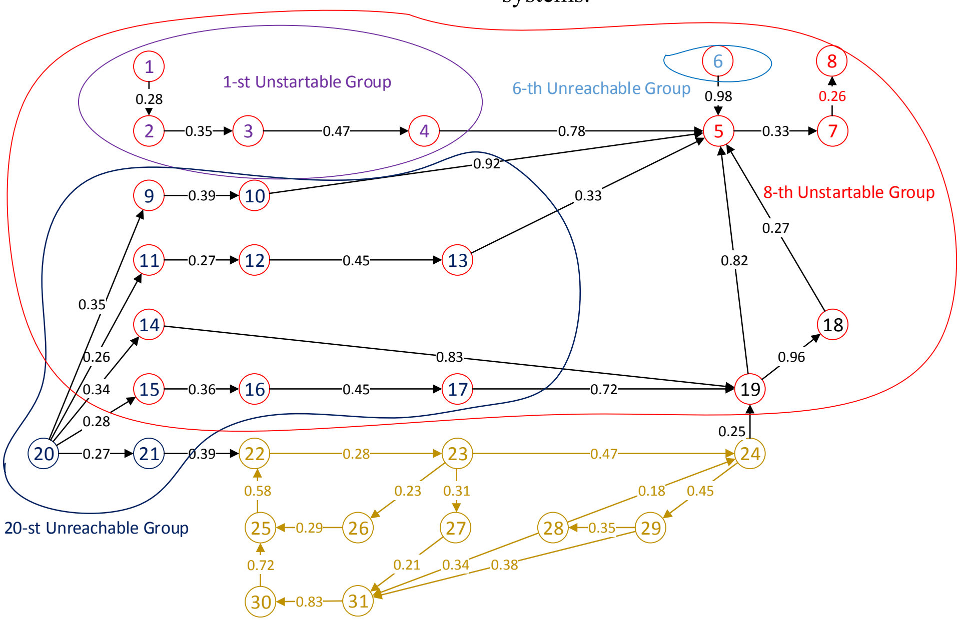


Figure 5. Radar system after preprocessing.

TABLE VI INDEGREE REPRESENTATION OF THE RADAR SYSTEM

Node	1	2	3	4	5	6	7	8	9	19	22	23	24	25	26	27	28	29	30	31
Indegree	0	1	1	1	6	0	1	1	1	3	2	1	2	2	1	1	1	1	1	3
Outdegree	1	1	1	1	1	1	1	0	1	1	1	3	2	1	1	1	2	2	1	1

Table VI shows the application of the indegree representation into the radar systems. The hierarchal method is introduced in Section 3.4. The representative result is shown in the following table. The first layer removes each node with a value 0, as shown in Table VII. Table VIII is the second layer that ticks out nodes with a value of 0. Table IX is the layer that extract nodes with 0 indegree. Table X shows the layer that removes the first ring.

TABLE VII FIRST LAYER OF THE GRAPH

Node	2	3	4	5	7	8	9	19	22	23	24	25	26	27	28	29	30	31
Indegree	0	1	1	5	1	1	0	3	2	1	2	2	1	1	1	1	1	3
Outdegree	1	1	1	1	1	0	1	1	1	3	2	1	1	1	2	2	1	1

TABLE VIII SECOND LAYER OF THE GRAPH

Node	5	7	8	19	22	23	24	25	26	27	28	29	30	31
Indegree	2	1	1	1	1	1	2	2	1	1	1	1	1	3
outdegree	1	1	0	2	1	3	2	1	1	1	2	2	1	1

TABLE IX LAYER THAT REMOVES EVERY NODE WITH 0 INDEGREE

Node	22	23	24	25	26	27	28	29	30	31
Indegree	1	1	2	2	1	1	1	1	1	3
Outdegree	1	3	1	1	1	1	2	2	1	1

TABLE X LAYER THAT REMOVES THE FIRST RING

Node	24	27	28	29	30	31
Indegree	1	0	1	1	1	3
Outdegree	1	1	2	2	0	1

Figure 6 shows the hierarchal result.

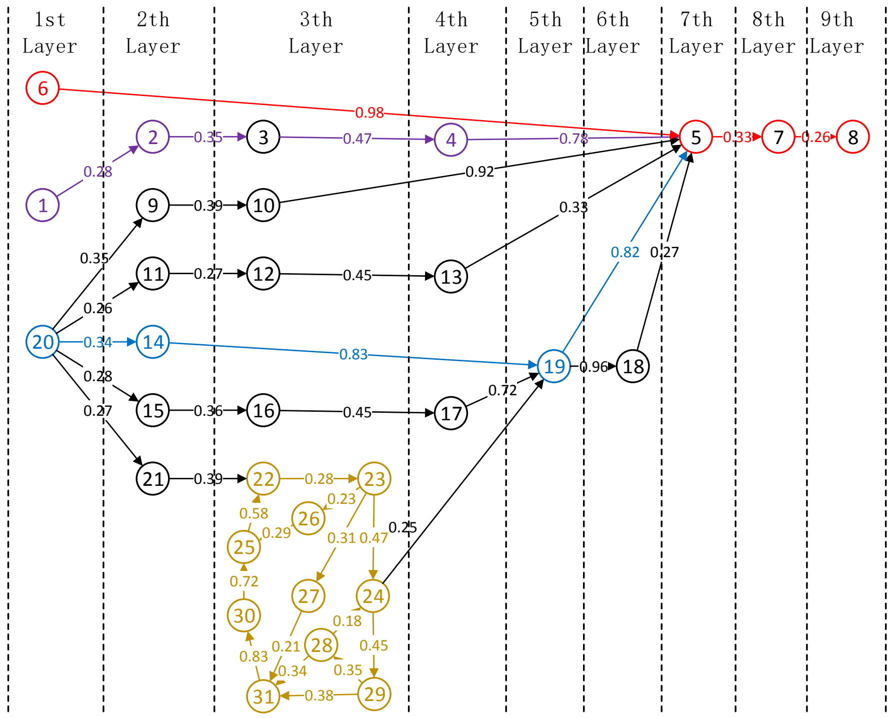


Figure 6. Hierarchical result of radar systems.

In this paper, we compared the traditional graph method [3] used in fault propagation analysis with our new graph representation and the entropy-based graph search method. The methods are run in Matlab 2014a on a desktop with Intel Core i7 3.2G CPU and 16G RAM. The same result can be achieved, but the proposed method is 10 times faster than the traditional one. Moreover, memory consumption is less than 7/13. Table XI shows the results.

TABLE XI COMPARISON WITH TARDITIONAL METHOD

Method	Traditional method[3]	Proposed method
Time (s)	0.62	0.06

V. CONCLUSION

Based on the hierarchical graph model, a new indegree hierarchical algorithm was proposed to analyze the fault propagation of complex electronic systems. A simulation analysis was conducted to evaluate the speed of the algorithm.

The main conclusions of this paper are presented as follows.

(1) Indegree representation is used to replace the traditional matrix representation for the first time. Indegree representation can reduce space storage and complex operation of graph preprocessing, which is suitable for large electronic systems.

(2) The proposed hierarchical algorithm could be used to identify the rings in the graph. Thus, redundant operations of fault diagnosis could be avoided.

(3) A new hierarchical algorithm was proposed to improve fault path search speed. As shown in the experiment, the proposed method could be applied to large-scale electronic systems. The speed of this method is faster than the traditional one.

Future work should focus on combining this entropy-based model with SDG to obtain improved results. This research direction would be meaningful for future studies.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China under Grant No. 61271035.

REFERENCES

[1] Hu B., She J., Yokoyama R., Hierarchical Fault Diagnosis for Power Systems Based on Equivalent-Input-Disturbance Approach. IEEE Transactions on

Industrial Electronics, vol.60, No.7, pp. 3529-3538, 2013.

[2] Zhou C., Huang X., Naixue X., A Class of General Transient Faults Propagation Analysis for Networked Control Systems. Systems, Man and Cybernetics: Systems, vol.45, No.4, pp. 647-661, 2015.

[3] Bai L., Du C., Guo Y., A fuzzy fault diagnosis method for large radar based on directed graph model, Journal of Shanghai Jiaotong University, vol.2015, No.20, pp. 363-369, 2015.

[4] Yang F., Xiao D., S. L. Shah D., Signed directed graph-based hierarchical modeling and fault propagation analysis for large-scale systems, Control Theory & Applications, vol.7, No.4, pp. 537-550, 2013.

[5] Barua A., Khorasani K., Hierarchical Fault Diagnosis and Fuzzy Rule-Based Reasoning for Satellites Formation Flight, IEEE Transactions on Aerospace and Electronic System, vol.47, No.4, pp. 2435-2456, 2011.

[6] Barua A., Khorasani K., Verification and Validation of Hierarchical Fault Diagnosis in Satellites Formation Flight, IEEE Transactions on Systems, Man, & Cybernetics, vol. 42, No.6, pp. 1384-1399, 2012.

[7] Kodali A., Zhang Y., Sankavaram C., Pattipati K. R., Salman M., Fault diagnosis in the automotive electric power generation and storage system, IEEE ASME Trans. Mechatron, vol. 18, No.6, pp. 1809–1818, 2013.

[8] Yuan H., Development of simulation model based on directed fault propagation graph, Computer Application and System Modeling, 2010, Vol.3, pp.686-690

[9] Zhang Q., Dynamic Uncertain Causality Graph for Knowledge Representation and Probabilistic Reasoning: Directed Cyclic Graph and Joint Probability Distribution, Neural Networks and Learning Systems, vol.26, No.7, pp. 1503-1517, 2015.

[10] Jiang M., Munawar M. A., Reidemeister T., Efficient fault detection and diagnosis in complex software systems with information-theoretic monitoring, Dependable and Secure Computing, vol.8, No.4, pp. 510-522, 2011.

[11] Bauer M., Cox J. W., Caveness M. H., Finding the direction of disturbance propagation in a chemical process using transfer entropy, Control Systems Technology, vol.15, No.1, pp. 12-21, 2007.

[12] Guo L., Yin L., Wang H., Entropy optimization filtering for fault isolation of nonlinear non-Gaussian stochastic systems, Automatic Control, vol. 54, No.4, pp. 804-810, 2009.

[13] Wang Y., Shi H., Jia L., Multi paths fault propagation model for network modeled system, Intelligent Control and Automation, 2014 11th World Congress on IEEE, pp. 5915-5920, 2014

[14] Rao N.S., On parallel algorithms for single-fault diagnosis in fault propagation graph systems, Parallel and Distributed Systems, vol.7, No.12, pp. 1217-1223, 1996.

[15] Jiang H., Patwardhan R., Shah S. L., Root cause diagnosis of plant-wide oscillations using the adjacency matrix, Proceedings of 17th IFAC World Congress 2008, pp.13893-13900, 2008.

[16] Shakeri M., Raghavan V., Pattipati K. R., Sequential testing algorithms for multiple fault diagnosis, Systems, Man and Cybernetics, Part A: Systems and Humans, vol. 30, No.1, pp. 1-14, 2000.

[17] Sehgal R., Gandhi O. P., Angra S., Fault location of tribo-mechanical systems a graph theory and matrix approach, Reliability Engineering & System Safety, vol. 70, No.1, pp. 1-14, 2000.