World Applied Sciences Journal 32 (6): 1121-1129, 2014
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2014.32.06.657

Development of Algorithm and Method for Multi-Machine Troubleshooting Systems Based on Technical Diagnostics

Kyaw Zaw Ye, Htike Aung Kyaw, E.M. Portnov, L.G. Gagarina and A.M. Bain

Department of Informatics and computer software system,
National Research University of Electronic Technology (MIET), Zelenograd, Moscow, Russia

Abstract: Our research paper focuses on the improvement of technique of multi-machine troubleshooting systems, based on the “AND-OR” graphs with the algorithms of majority principle which allows on the basis of the input information about the correctness of the decision of problems to identify the faulty machines and based on the complete testing algorithm for elements of multi-computer complexes searching by criteria failed element.

Key words: Technical diagnostics・Fault・Identification・“AND-OR” graph・Efficiency・Control devices・Multi-machine・Troubleshooting・Majority principle・algorithm

INTRODUCTION

The control devices and multi-computing complexes which are nowadays exploited to improve the overall performance of industrial processes involve both sophisticated digital system design techniques and complex hardware (input-output sensors, actuators, components and processing units). In such a way, the probability of failure occurrence on such equipment may result significant and an automatic supervision control should be used to detect and isolate anomalous working conditions as early as possible. At present, a number of high-tech industries, research and educational processes use many kinds of instrumentations that greatly improve the efficiency of information processing. Despite the undoubtedly positive effect of the use of instruments, complex ongoing devices, we have to state their lack of effectiveness, due to a number of technical and economic circumstances. In particular, rather acute problem of improving the resiliency and reliability of elements of instrumentations, the life of which often exceeds the standard. In connection with the above, one of the most important requirements for instrument complexes is their high availability and the ability to effectively identify failures [1-5].

Many buildings fail to perform as well as expected because of problems that arise at various stages of the life cycle, from design planning to operation. These problems relate to different parts of the building, including the envelope, the automatic manufacturing system and the lighting system. The consequences include increased energy costs, occupant discomfort, poor productivity and health problems and increased maintenance costs. Manual methods for commissioning and trouble-shooting are available and are starting to be used more widely. Over the last few years, a number of researchers in the US and elsewhere have developed automated methods for diagnosing faults in buildings, primarily in automated control systems (Hyvarinen 1997) and are now starting to demonstrate their use in real buildings.

Since the early 1970's, the problem of fault detection and isolation in multi-machine processes has received great attention and a wide variety of model-based approaches has been proposed and developed. Theoretical and practical aspects of technical diagnostics, fault tolerance issues involved in instrument making such famous scientists as P.P Parkhomenko, Caribbean V.V, Sogomonyan E.S, Loaf M.F, A.V Lobanov, Schlichting R., Rennels D.A, Dolev D., Professor Brett Neilanc(Australia) and many others etc.

Corresponding Author: Kyaw Zaw Ye, Department of Informatics and computer software system,
National Research University of Electronic Technology (MIET), Zelenograd, Moscow, Russia.

On the other hand, for the diagnosis of faults, mathematical models of the process under investigation are required, either in state space or input-output. A state space description of the system provides general and mathematically rigorous tools for system modeling and residual generation, which may be used in fault diagnosis of industrial systems, for both the deterministic and the stochastic case.

The analysis showed that the modern industrial technologies used in various fields, require new approaches to ensure their reliability and effective methods of technical diagnostics. In this regard, there is a steady increase in the number of faults and failures, worsening the number of products increase the probability of accidents and crashes. Such a negative situation related to the unreliability of the devices can be neutralized by increased personnel skills, able to justify decisions taken in the event of a negative situation. One aim of this article is to develop for providing a conceptual framework for complete testing algorithm, searching fault in multi-computer and describing diagnostic methods based on AND-OR graphs.

Complete Testing Algorithm for Elements of Multi-computer Complexes Searching by Criteria Failed Element: Optimize the process of diagnosing for multi-computer complexes (MCC) using overlapping tests with complete coverage of elements. The theoretical aspects of the problem are the following.

Let the beginning of a M-*th* step of the verification process carried out by a sequence of tests $H^{(M-1)*}$ = {h1,..., h(M-1)} and the problem reduces to finding a subset of the failed component of $\Omega^{(M-1)}$. (Before the start of the system checked $\Omega^{(0)} = \Omega$. $\Omega^{(0)}$ Includes all elements of the system and does not include any test). The search algorithm only failed component is as follows [4].

- Defined values $\tilde{q}_j^{(0)}$ - conditional probability of failure is the j-th element, if the tested set exactly one element failed:

$$\tilde{q}_j^{(0)} = \hat{q}_j \left(\sum_{i \in \Omega(0)} \hat{q}_i \right)^{-1} \text{ where } \hat{q}_i = q_i p_i^{-1}$$

- For each significant test to calculate the probability of unsuccessful outcome of the tested subset: $Q_i^{(0)} = \sum_{j \in \Omega_j \cap \Omega^{(0)}} \tilde{q}_j^{(0)}$.
- For each material test h_i are associated costs $Z_i^{(0)}$ in view of the fact that a test sequence is performed $\sigma^{(0)}$. In general, the costs of conducting the test as hi can decrease or increase, subject to other tests. (For example, or can be connected by previous inspections necessary for the test devices, or vice versa, holding previous audits may hinder access to the right parts of the system).

For each test t_i determined values

$$g_i^{(0)} = Z_i^{(0)} / Q_i^{(0)}$$

- Selected this test h_k, for which a minimum:
- Applicable test h_k:
  - If the test succeeds h_k, the problem reduces to finding a subset of the failed component $\Omega^{(1)} = \Omega^{(0)} \setminus \Omega_k$;
  - If the test h_k fails, the problem is reduced to finding a subset of the failed component $\Omega^{(1)} = \Omega^{(0)} \cap \Omega_k$; if in these cases, the subset $\Omega^{(1)}$ consists of a single element, the search failed element ends here.

- New fixed sequence of tests applied $H^{(1)}$, which contains the previous sequence $H^{(0)}$ and the last applied test h_k: $H^{(1)} = \{H^{(0)}, h_k\}$.
- To Subset $\Omega^{(1)}$, starting with c ï.1, the procedure checks with the corresponding change in the superscript (0) in the index (1). The verification procedure continues as long as claimed in 6, at some step k is formed subset $\Omega^{(1)}$, which consists of a single element.

Procedure described in the application to multi-computer complexes will implement consistent with the development of the verification process. For current calculations and selection of another test used computer with the necessary software and advance the memorized array of source data (probability of failure, duration of inspections, test specifications).

The same procedure can be done in advance and make use of the instruction sequence of tests depending on the results of previous, for example: "if the test h_k is successful, then continue to test hi, if the same test h_k unsuccessful, then to test h_j » (Figure 1).

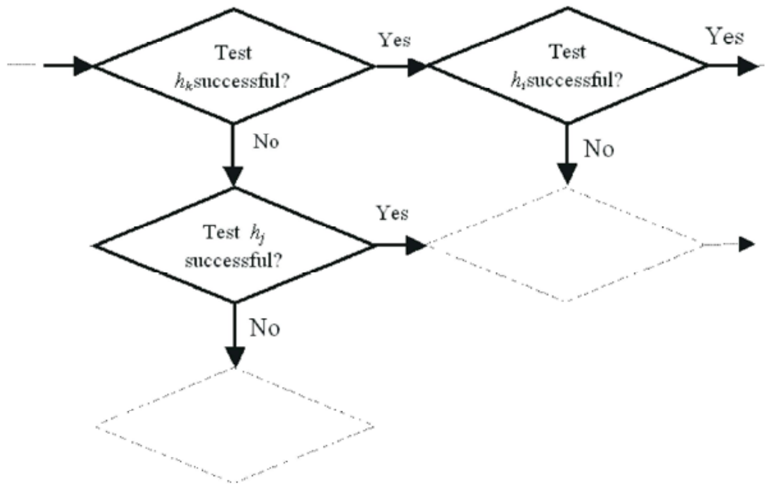


Fig. 1: Fragment algorithm of complete testing for multi-computer complexness

Table 1: Matrix of distributing elements for testing

Numbers of test	Numbers of elements 1	2	3	4	5	6	7	8
1	1	1			1	1		
2		1		1			1	
3			1		1	1		
4	1					1		1
5	1		1	1	1			
6					1		1	1

Concretizing the task. When testing multi-computer systems chooses to 8 devices (main modules). №1 — LA (line adapter), №2 —controller for internal line devices MCC; №3 –first PC; №4 —second PC; №5 —Linear controller for interfacing of PC; №6 — Analog Input Module; №7 —third PC; №8 —Output of the module of control commands and can be tested within six tests which matrix is shown in Table 1.

Known time costs (in relative terms) to conduct each test: Z_1 = 2.5; Z_2 = 2; Z_3 =1,0; Z_4 =1,2; Z_5 =1,5; Z_6 = 1,3,wherein the values Z_i do not depend on the order of tests.

Empirically established as a prior probability of failure of the above-listed items: q_1 = 0,04; q_2 = 0,03; q_3 = 0,01; q_4 = 0,01; q_5 = 0,03; q_6 = 0,02; q_7 = 0.01; q_8 = 0.02. Form the complete test complex instruction checks the minimum average cost for necessary in this test.

Find the conditional probability of failure for each element:

$$\tilde{q}_1 = 0.028;\ \tilde{q}_2 = 0.018;\ \tilde{q}_3 = 0.005;\ \tilde{q}_4 = 0.005;$$
$$\tilde{q}_5 = 0.018;\ \tilde{q}_6 = 0.011;\ \tilde{q}_7 = 0.005;\ \tilde{q}_8 = 0.011;$$

Calculate the magnitude of the probability of unsuccessful outcome $Q_i^{(0)}$ for each test:

$$Q_1^{(0)} = \tilde{q}_1 + \tilde{q}_2 + \tilde{q}_5 + \tilde{q}_6 = 0.075;\ Q_2^{(0)} = \tilde{q}_2 + \tilde{q}_4 + \tilde{q}_7 = 0.028;$$
$$Q_3^{(0)} = \tilde{q}_3 + \tilde{q}_5 + \tilde{q}_6 = 0.034;\quad Q_4^{(0)} = \tilde{q}_1 + \tilde{q}_6 + \tilde{q}_8 = 0.049;$$
$$Q_5^{(0)} = \tilde{q}_1 + \tilde{q}_3 + \tilde{q}_4 + \tilde{q}_5 = 0.055;\ Q_6^{(0)} = \tilde{q}_5 + \tilde{q}_7 + \tilde{q}_8 = 0.034;$$

Next, for each test we find:

$$g_1^{(0)} = \frac{Z_1}{Q_1^{(0)}} = \frac{2,5}{0,075} = 33,2;$$

$$g_2^{(0)} = \frac{Z_2}{Q_2^{(0)}} = 72,4;$$

$$g_3^{(0)} = \frac{Z_3}{Q_3^{(0)}} = 29,8;$$

$$g_4^{(0)} = \frac{Z_4}{Q_4^{(0)}} = 24;$$

$$g_5^{(0)} = \frac{Z_5}{Q_5^{(0)}} = 26,\ 8;$$

$$g_6^{(0)} = \frac{Z_6}{Q_6^{(0)}} = 38,8.$$

It is seen that the first test should be used h_4, т.к. value $g_4^{(0)}$ is the smallest. The test h_4 can be successful or unsuccessful. Consider first the outcome, т.е. the failed element is among those which were not covered by the testh_4.

You're a subset of the $\Omega^{(1)}$ = {2, 3, 4, 5, 7}. We calculate for each of the remaining test $Q_i^{(1)}$:

$$Q_1^{(1)} = \tilde{q}_2 + \tilde{q}_5 = 0.036; Q_1^{(1)} = \tilde{q}_2 + \tilde{q}_4 + \tilde{q}_7 = 0.028; Q_1^{(1)} = \tilde{q}_3 + \tilde{q}_5 = 0.023;$$
$$Q_1^{(1)} = \tilde{q}_3 + \tilde{q}_4 = 0.028; Q_1^{(1)} = \tilde{q}_5 + \tilde{q}_7 = 0.023,$$

And then $g_i^{(1)}$:

$$g_1^{(1)} = 68,8; g_2^{(1)} = 72,4; g_3^{(1)} = 43,7; g_5^{(1)} = 54,3; g_6^{(1)} = 56,8.$$

Thus, after a successful test h_4 should be carried out test h_3. By itself, this test may be, in turn, successful and non-successful. Consider the second possibility: the failed element is in the subset, which is verified test h_3, of the elements $\Omega^{(2)}$ = {2, 4 and 7}.

Then:

$$Q_1^{(1)} = \tilde{q}_2 = 0.036; Q_2^{(1)} = \tilde{q}_2 + \tilde{q}_4 + \tilde{q}_7 = 0.028;$$
$$Q_5^{(1)} = \tilde{q}_4 = 0.028; Q_6^{(1)} = \tilde{q}_7 = 0.023,$$
$$g_1^{(1)} = 68,8; \; g_2^{(1)} = 72,4;$$
$$g_5^{(1)} = 54,3; \; g_6^{(1)} = 56,8.$$

Value $g_5^{(1)}$ is lower, but successful test h_5^{oti} not give useful information, as does not share many elements into two subsets. We now consider the other branch, т.e. test h_3^{oti} is unsuccessful and for a subset of elements $\Omega^{(2)}$ = {3, 5} we need to compute $Q_i^{(1)}$. т.e.

Value $g_5^{(1)}$ is lower, but successful test h_5^{oti} not give useful information, as does not share many elements into two subsets.

The next step - the test h_4 is unsuccessful. Repeat the procedure for a subset of $\Omega^{(2)}$ = {1, 6 and 8}.

$$Q_1^{(1)} = \tilde{q}_1 + \tilde{q}_6 = 0.039;$$
$$Q_3^{(1)} = \tilde{q}_6 = 0.011;$$
$$Q_5^{(1)} = \tilde{q}_1 = 0.028;$$
$$Q_6^{(1)} = \tilde{q}_8 = 0.011,$$

And then $g_i^{(1)}$:

$$g_1^{(1)} = 64,3;$$
$$g_3^{(1)} = 94,3;$$
$$g_5^{(1)} = 53;$$
$$g_6^{(1)} = 122,5.$$

Hence we conclude: if h_5 is unsuccessful, then refused the first element. If successful, it will explore many $\Omega^{(2)}$ = {6, 8}.

$$Q_1^{(1)} = \tilde{q}_6 = 0.011;$$
$$Q_3^{(1)} = \tilde{q}_6 = 0.011;$$
$$Q_6^{(1)} = \tilde{q}_8 = 0.011,$$

And then $g_i^{(1)}$:

$$g_1^{(1)} = 235,7;$$
$$g_3^{(1)} = 94,3;$$
$$g_6^{(1)} = 122,5.$$

Obviously, that has minimal cost test h_3 and if it is successful, then the failed element - 8 if not succeed - 6. If you want to make a guide with a description of the sequence of inspections, you should fix the resulting sequence only (Figure 2) and return to the stage when the test h_4 was performed, but now assume that the test was unsuccessful, i.e. search for the failed element of the subset $\Omega^{(2)}$ = {1, 6, 8}.

The result is a second fragment of test instructions complex, shown in Fig. 3.

Such a procedure continues until all the pieces are constructed algorithm complete testing systems to localize the failure to the room a single element. Chart of a complete test for this case is shown in Figure 4.

It should be noted that in the case of testing one element after another we can get a simple rule for numbering test for finding procedures minimizing search costs failed element. Commutes trick is that from any arbitrary numbering pair permutation tests can only check (change verb) the neighboring finite number of steps to go to any predetermined sequence of them conducted, including optimal.

If you can find a useful criterion for comparing two different tests on the effect of their applications on the target functional result- the average search time failed component, under certain conditions it is possible to calculate the criterion for each test and then enumerate all the tests in accordance with a monotonic variation of this criterion [3, 5, 6, 9].

For an arbitrary numbering objective functional tests

$$C\,[\sigma(\Omega_i)] = z_{(i)}^k + Q_{(i)}^k \; C\,[\sigma(\Omega_{(i)}^k)] \; + \; (1 - Q_{(i)}^k) \; C\,[\sigma(\bar{\Omega}_{(i)}^k)] \tag{1}$$

has the form:

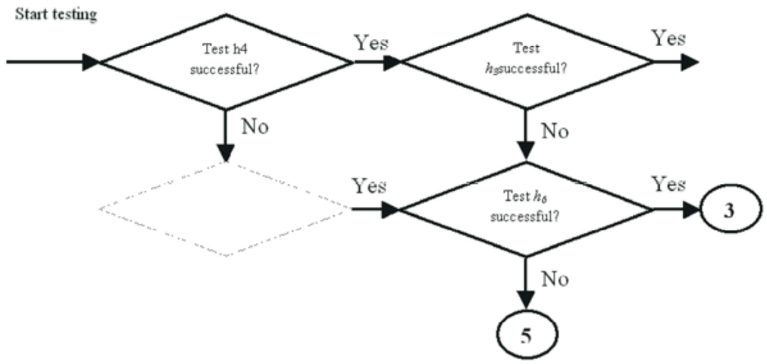


Fig. 2: The first fragment of the algorithm for complete testing of multi-computer complexness (in the circles - numbers of failed elements)

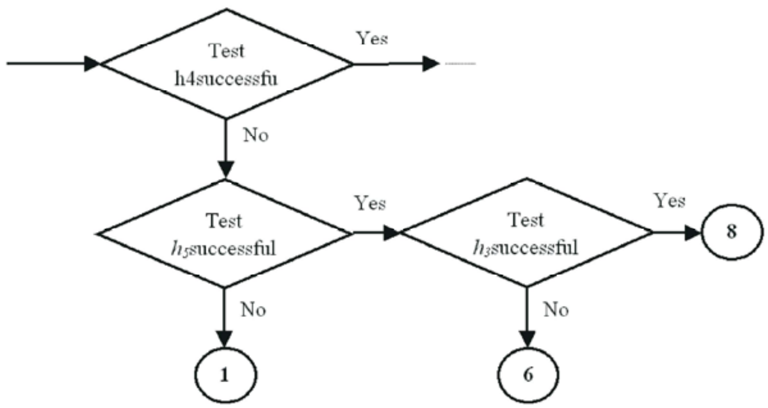


Fig. 3: The second fragment of the algorithm for complete testing the multi-computer complexness

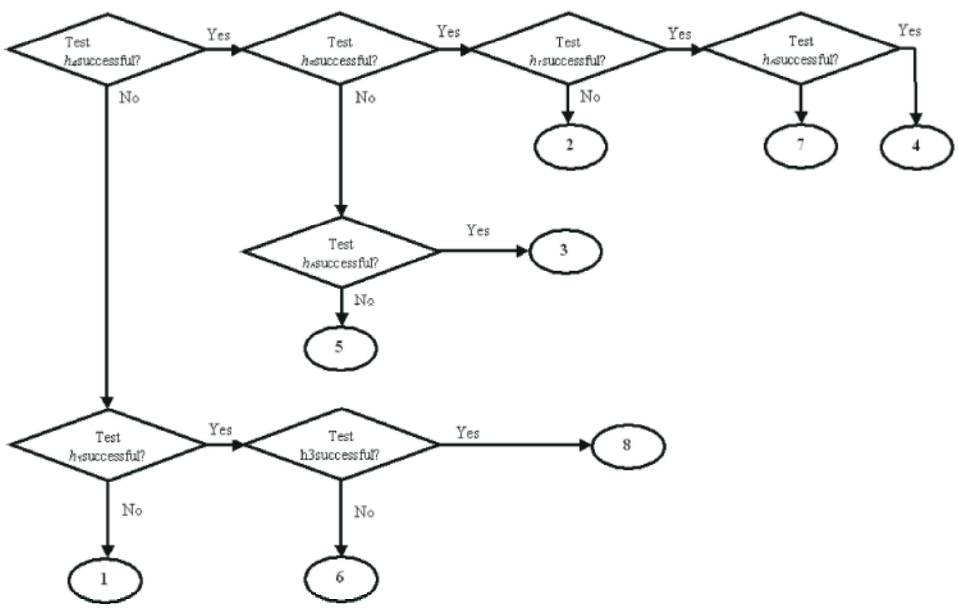


Fig. 4: Diagram complete testing MCC

$$C\,[\sigma(\Omega)] = z_1 + \tilde{q}_1 C\,[\sigma(e_1)] + \tilde{P}_1 C\,[\sigma(\Omega \setminus e_1)], \qquad (2)$$

where, as e1-singleton, which is then no longer need to check and

$$C\,[\sigma(\Omega \setminus e_1)] = c_2 + C\,[\sigma(\Omega \setminus e_1 \vee e_2))].$$

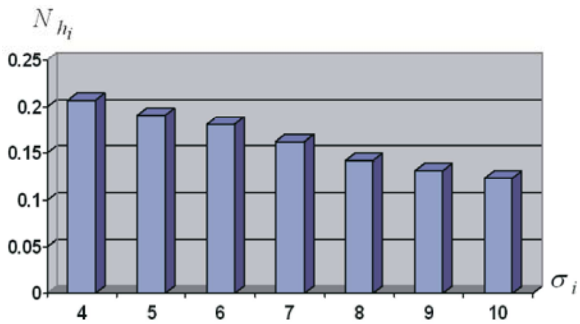


Fig. 5: Graphical relationship $\sigma_i\left(N_{h_i}\right)$

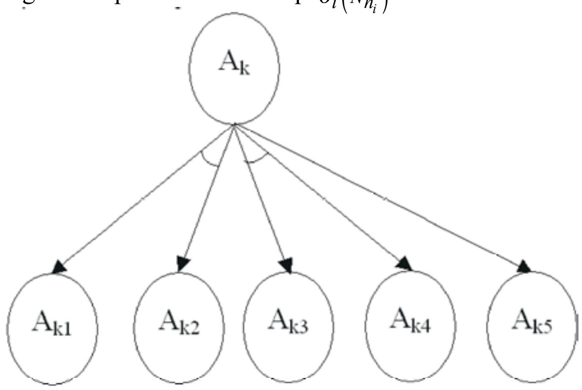


Fig. 6: General Scheme of the “AND-OR” graph

Table 2: Results of simulation by complete algorithm testing MCC

N_{h_i}	$Z^{mp}_{cym_i}$	$Z^{np}_{cym_i}$	σ_i
4	6,32	5,01	0,207
5	7,96	6,44	0,191
6	9,22	7,54	0,182
7	11,11	9,31	0,162
8	12,55	10,77	0,142
9	13,72	11,91	0,132
10	16,2	14,19	0,124

Finally

$$C = z_1 + \tilde{p}_1\ (z_2 + \tilde{p}_2\ (z_3 + \tilde{p}_3\ (z_4 + ...))). \qquad (3)$$

Writing a similar expression for the case when the item numbers k and k + 1 changed the procedure for checking and comparing the values of the total cost for both of these cases, we find that the optimal order if possible check piecemeal responsible numbering elements in accordance with the condition:

$$\frac{z_1}{q_1} \le \frac{z_2}{q_2} \le \ ... \le \frac{z_n}{q_n}.$$

To confirm the benefits of the proposed method was carried out in the software simulation for different numbers of test - N_{h_i}, providing complete testing complex. During the simulation of testing MCC to compare the relative time spent on the search for a single failed component to the traditional - $Z^{??}_{???_i}$ and the proposed methodology $Z^{??}_{???_i}$ and defines improve performance testing $\sigma_i = \frac{Z^{??}_{???_i} - Z^{??}_{???_i}}{Z^{??}_{???_i}}$. Simulation results are presented in Table 2 and Figure 6 shows a graphical representation of $\sigma_i\left(N_{h_i}\right)$.

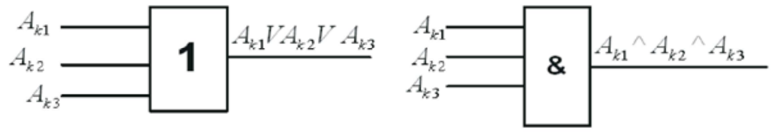
From the data presented in Table 2 and in Fig. 5, it is clear that depending on the number of tests provided speedup complete testing multi-computer complex from 12 to 20%. And the result of simulation confirmed the advantages of the proposed method, which helps to test the full MCC searching by criteria single failure.

Method of Searching Fault in Multi-computer Complexes Based on the “And-or" Graphs”: One of the main aspects of increasing multi-computer systems resiliency is the task of fault diagnosis of network computers, identifying their location and cause [6, 12].

It is well known that this problem is difficult to solve, especially for large, including cluster computing systems, in which the number of computers can reach several thousands. The reason for this is the difficulty of formalizing the fault information inMCC, as it requires ways to describe hard to formalize such characteristics that are as the intensity of problems, features of the machines operation in the process of solving the problems before and during the fault occurrence, etc.

One of the ways of solving the problem to determine the cause and fault location in these systems is to use the results of problems solving, or to perform other actions when there is information on ways of problem motion in the computer network. While solving this problem the following aspects should be taken into account:

- Each problem is solved on several nodes (computers) at the same time, using different methods. Result is not known beforehand. When the results coincide on all computers, all computers are considered serviceable. If the results do not coincide, one computer is faulty.



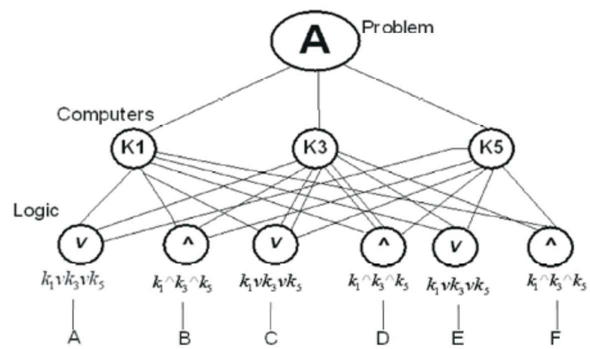


Fig. 7: Model "AND-OR" graph for the problem solving of fault searching in third computer.

Table 3: Matrix D

	Computers						
Problems	K_1	K_2	K_3	K_4	K_5	K_6	K_7
A	+			+		+	
B	+	+			+		
C	+	+	+			+	
D	+	+	+				+
E			+	+	+		+

- Each problem is solved in one of the *n* nodes (computers). The computer, which solves the problem is unknown, that is, it is randomly selected by the system itself. The result of the solutions is known in advance and the faulty computer is determined by the wrong answer.
- Each problem can be solved on several nodes (computers) of the randomly selected systems, using different methods. The faulty computer is determined by comparing the responses and taking into account the errors. The users determine the appropriateness of solutions made ??by the user task.

The structures that look like graphs are called "AND / OR" graphs and are used to partition the problem image into alternative sets of the resulting problems [1, 13]. Let there be given the task of A_k, which can be solved either by solving the problems A_{k1} and A_{k2}, or by solving the problems $_{\mathrm{Ak3}}$ and A_{k3}, either by solving the problem A_{k5}. This relationship is represented by the structure in Figure 6.

Under the " AND- OR" graph we often understand the graph for which the first property holds and for the output arcs functions "and" always holds.

The model of "AND-OR" graph for solving the problem A on three computers K_1, K_3 and K_5 is presented in Figure 7.

The conclusion whether the computer is faulty on not faulty is made based on the output results, presenting the combination of solving the problems K_1, K_3 and K_5 on computers [13-15]. In general, when a sufficiently large number is used the solutions whether the problem is solved correctly on not is taken using the majority principle, that is, the right result is considered to be the same result obtained on more than half of the computers, if their number is not less than 2m +1 in case of friendly fault and is not less 3m +1 in case of hostile malfunction. Later the faults considered are assumed to be friendly only.

Consider the example of a diagnosis problem MCC consisting of 7 computers that solve 5 similar problems. Initial problem allocation to computers is represented by the following matrix D.

Lines A, B, C, D of the matrix D correspond to current tasks and columns - to computers K_1, K_2, K_3, K_4, K_5, K_6, K_7, in which they are solved. After analyzing the results of solving the problem they are compared and processed using the "AND-OR" graphs. The result of solving the problem is not known beforehand. Each problem is solved on three computers. For example, the problem A is solved on computers K_1, K_4, K_6; the problem B - on computers K_1, K_2, K_5, etc. Thus, the computer K_1 solves problems A, B, C and D, the computer K_2 are solves problems K_2 B, C and D, etc.

Each wrong answer while solving the problem in the proposed model corresponds to 0 and the correct to 1. If the problem solutions coincide each with its own response, the output result k_i= 1 otherwise k_i = 0.

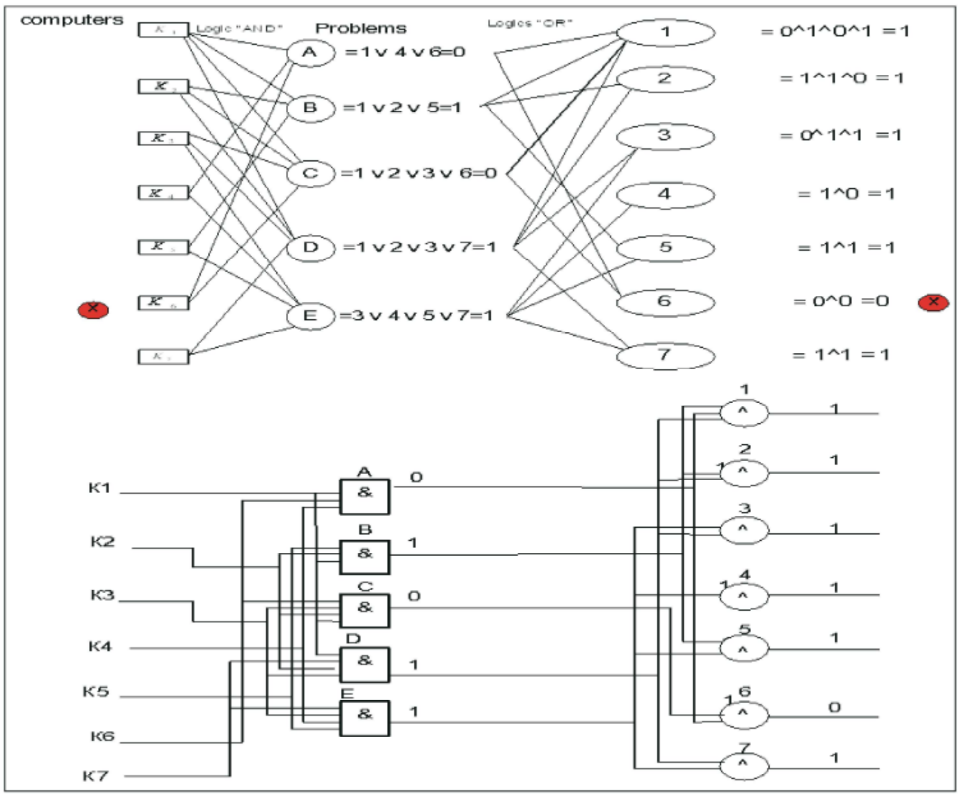


Fig. 8: Model diagram for the problem solving in the multi-computer computing system.

Figure 8 presents a model - diagram for solving these problems in a multi-computer computing system. The left side of figure 8 is the solution of problems based on the "AND" graph, which shows that the problem A and B are solved in correctly (output result is 0).

Thus, the preliminary conclusion is that there is a fault in one of the computers K_1, K_2, K_3, K_4 and K_6. The right side of the figure shows the solution of problems using the "OR" graph, this scheme is an inspection and is a mirror image of the circuit the "AND" graph. Using the output, results of "OR" the graph we can come to final conclusions about the computer malfunctions K6 (output value is 0).

Thus, the model presented in the form of “AND-OR” graph enables us to make an algorithm that enables us to issue a message about the potentially faulty computer using the input information on the correctness of the problem solution.

CONCLUSION

In this paper, based on the studies and research can be following conclusion.

- An algorithm for complete testing the elements MCC criterion at the minimum search time of the failed component, in the course of simulating shows that depending on the number of tests provided speedup complete testing multi-computer complex from 12 to 20%.
- The technique of multi-computer troubleshooting complexes based on the “AND-OR” graphs, allowing on the basis of the input information about the correctness of the decision to identify the faulty computer problems.

REFERENCES

1. Kyaw Zaw Ye, E.S. Kukushkin and O.I. Lisov, 2012. Fault finding in the multi-computer computer systems based on the "AND-OR" graph / / journal of "The defense complex - the scientific and technical progress of Russia." -M.: Federal State Unitary Enterprise "VIMI," 1: 71-75.
2. Kyaw Zaw Ye, 2012. Method for determining a single fault with the technical diagnosis of multiple computer systems / / Scientific and practical journal "Industry aspects of Engineering." - M.: Publishing INGN, 1: 18-19.

3. Kyaw Zaw Ye, 2013. Evaluation model of CPU performance of distributed fault-tolerant automation systems / / Modern information technology. International Scientific and Technical Conference: Collected state., Penza, pp: 124-127.
4. Kazak, D.S., 2009. The model for the evaluation of fault tolerance of automated control devices and diagnostics// Proceedings of the universities. Electronic.-M.: MIET, 1: 89-92.
5. Kazak, D.S., 2008. A mathematical model of CPU performance evaluation of distributed automation systems for process testing// Scientific and technical journal "Engineering and Technology", Moscow: Publishing House of the "Company Sputnik, 5: 37-38.
6. Kazak, D.S., 2008. The checks performance of automated systems for technological tests in instrument//Scientific and technical journal "Engineering and Technology", Moscow: Publishing House of the "Company Sputnik, 5: 39-40.
7. Kyaw Zaw Ye and Alexander M. Bain, 2013. Methods to improve the detection of failures and troubleshooting for technical diagnostics in instrument// Manufacturing Science and Technology 1(2): 31-35, 2013, DOI: 10.13189/mst.2013.010202, Horizon research publishing cooperation. USA, http://www.hrpub.org/download/20131107/MST2-15401707.pdf
8. Philippot, A., P. Marangé, F. Gellot and R. Riera, 2013. Decentralized Diagnosis and Diagnosability by Model Checking. Universal Journal of Control and Automation, 1, 28 - 33. doi: 10.13189/ujca.2013.010202. http://www.hrpub.org/journals/article_info.php?aid =379
9. Berezhnaya, E.V. and V.I. Berezhnoy, 2001. Mathematical modeling of economic systems : Training. Allowance. - Moscow: Finances and Statistics, pp: 368.
10. Kazak, D.S., 2008. Selection of hardware and software platform for distributed automated information systems / / Proceedings of the Second All-Russian Inter-University Scientific Conference "Actual problems of information. The development of information technology infrastructure and systems. " -M. : MIET, pp: 118.
11. Kyaw Zaw Ye and O.I. Lisov, 2011. Intelligent training complex for learning processes of technical diagnostics in multi-computer computer systems / / drive higher education institutions Chernozem, 1(23): 73-75.
12. Kyaw Zaw Ye, Htike Aung Kyaw, A.M. Bain and R.A. Kassimov, 2013. Technique for improve efficiency intermodal information exchanges in automated control systems in the energy complex / / drive higher education institutions Chernozem, 1(31): 49-53.
13. Kyaw Zaw Ye, A.M. Bain and R.A. Kassimov, 2013. Technique of reducing the intensity of information flows integrated information management systems in the energy / / Journal "Defense Complex - scientific and technical progress of Russia." -M. : FSUE " VIMI, 3: 33-37.
14. Kyaw Zaw Ye and Pyae Thein Naing, 2013. Automated control system of technological processes in gas streams / / Journal "Scientific Review " -M. : PH " Science Education, 7: 65-69.
15. Kyaw Zaw Ye, A.M. Bain, R.A. Kassimov and E.A. Hrinchenko, 2013. Methodology of cycles during information exchanges and control for distributed energy units / / Journal "Defense Complex - scientific and technical progress of Russia." -M. : FSUE " VIMI, 4: 18-23.