

SELF-DIAGNOSTICS OF COMPLEX SYSTEMS WITH A SOFTWARE-CONFIGURABLE STRUCTURE

I.V. Korobiichuk¹, R.V. Hryshchuk², V.O. Horoshko³, Yu.E. Hohlachova³

¹Warsaw University of Technology,

1, pl. Politechniki, Warszawa, 00-661, Poland; e-mail: igor@mchtr.pw.edu.pl

²Sergey Korolyov Zhytomyr Military Institute,

22, Mira, Ave., building ZhVI, Zhytomyr, 10004, Ukraine; e-mail: dr.hry@i.ua

³National Aviation University,

1, Cosmonaut Komarova, Ave., Kiev, 03058, Ukraine; e-mail: professor_va@ukr.net, hohlachova@gmail.com

Using the method of simulation and statistical modeling, the operability was established and estimates of the effectiveness of the proposed algorithm for self-diagnostics of complex computer systems with a software-configurable structure were obtained. The structure of complex computational systems is the Cayley diagram of some finite abelian arc group in the Cayley diagram that have the markers of the generators of the group. The described graphs are called CAIS graphs (graphs for providing finite abelian isotropic and symmetric structures). Due to the limited number of intermodule connections of complex computing systems, it is necessary to use working links between modules to determine its state. Consequently, the diagnostic graph for complex computing systems is CAIS graph. The symmetry property of the CAIS graph means that adjacent modules of complex computer systems are mutually testable. It is possible to correctly determine the state of complex computing systems if it is performed by deliberately working tools that make up the core of complex computer systems. Define the diagnostic core as a collection of tools that have a higher reliability than the system checked with it. The reliability of the kernel is provided by means that do not belong to the system it checks. One method of automating the detection of the state of modern complex computing systems is the method of promotion. At each step of the promotion, the serviceability of some nodes is determined by the integrity of the others, which is set in the previous steps. The process of spinning has always started from the core of the system. The functional characteristics of the modules of complex computing systems are such that any of them can be used as a core. If, then the task of determining the state of the system can be clarified: by the results of mutual verification, identify at least one known serviceable module. Identified modules are used as the core of complex computer systems. The method of determining the state of serviceability of modular survivable complex computing systems is the modification of the known method of promotion. For complex computational systems with a software-configurable structure, the concept of a diagnostic kernel is concretized, a heuristic decentralized kernel allocation algorithm is described.

Keywords: complex computing systems, diagnostic model, diagnostic graph, algorithm, costs of diagnostics

Introduction

Complex computing systems (CCS) with software-configurable structure [1,2] are durable modular systems, the functioning of which is based on the use of decentralized control algorithms. Current state of microelectronics makes it possible to implement CCS on computational modules, each of which is a complete computer. Modules, in terms of diagnosis, are units of indivisible system. Their functional characteristics allow to perform a full check of neighboring modules. In this case, the state of serviceability or operability (hereinafter – “the state”) of CCS in conditions of multiple faults is determined by comparing the results during cross-checking of modules [3-11]. Multiplicity of faults is a maximum permitted number of faulty modules, under which still exists the ability to determine the state

of the system. Dependence of the multiplicity malfunctions on the number of modules and the structure of CCS is investigated in paper [3,4].

The CCS structure is a Cayley diagram of some finite abelian G groups. Arches in the Cayley diagram have markers of group generators. The described graphs are called FAIS graphs (graphs used for providing finite abelian isotropic and symmetric structures) [4, 5].

Considered CCS are t -diagnosed with repair, i.e. there is a sequence of execution of the set of all possible tests and repair of identified malfunctions in modules, that allows to identify all the problems initially present in the CCS, provided that their number does not exceed t . Paper [3,4] provides a method to determine the state of CCS, t -diagnosed with repair, which are presented by Cayley graphs. This paper describes a heuristic algorithm that solves this method, and evaluates its effectiveness.

Materials and methods

The diagnostic model of CCS is the digraph $G=(V,E)$, in which set $V=\{1,2,\dots,N\}$ represent modules of the system, and set E represents performing arc tests. For $i,j \in V$ (i,j) an arc (i,j) (from i to j) exists, if and only if the module i can test the module j . The vertices of the graph have marks x , the values $x(i)=1$ of which determine the state and $x(i)=0$ if the module i is faulty. The test results for the CCS modules correspond to the weights of the graph arcs. For $(i,j) \in E$ the weight $a(i,j)=0$, if the module confirms the serviceability of the module j ; otherwise $a(i,j)=1$. The set of vertices, containing singular values of x marks, makes up an image of the system's malfunctions, and the totality of the values of the weights a of the graph forms its syndrome S . For any image of a malfunction F , $F \subseteq V$, the conditions for the formation (generation) of a syndrome S are as follows:

Condition 1. Performing $\forall (i,j) \in E \{ (i \in \bar{F} = V - F) \wedge (j \in F) \rightarrow a(i,j) = 1 \}$.

Condition 2. Performing $\forall (i,j) \in E \{ (i \in \bar{F}) \wedge (j \in \bar{F}) \rightarrow a(i,j) = 0 \}$.

Condition 3. Performing $\forall (i,j) \in E \{ (i \in F) \wedge (j \in \bar{F}) \text{ possible } a(i,j) = 1 \}$.

Condition 4. Performing $\forall (i,j) \in E \{ (i \in F) \wedge (j \in F) \text{ possible } a(i,j) = 0 \}$.

In the conditions the arrow is indicating an implication operator. The result of the test performed by a valid module determines the status of the module being tested (conditions 1 and 2), and when executed by the faulty module it does not depend on the state of the module being checked (conditions 3 and 4).

In the described model, the state of CCS is defined by identification of the graph vertices that form the set F , according to a given S syndrome.

Due to the limited number of inter-modular CCS links, it is necessary to use working links between the modules to determine its state. Therefore, the diagnostic graph $G=(V,E)$ for CCS is an FAIS graph. The symmetry property of the Cayley graph means that adjacent CCS modules are mutually testable.

It is possible to correctly determine the state of CCS, if it is carried out by knowingly effective means that make up the kernel of CCS. Define the diagnostic core as a collection of tools that have a higher reliability than the system checked with it. The reliability of the kernel is provided by means that do not belong to the system it checks.

One of the methods for automating the determination of the state of modern CCS is the method of promotion [3]. At each step of the promotion, the serviceability of some nodes is determined by the integrity of the others, which is in turn determined in the previous steps. The process of spinning has always started from the kernel of the system.

The functional characteristics of CCS modules are such that any of them can be used as a kernel. If $t < N$, then the task of determining the state of the system can be clarified: to reveal at least one known serviceable module by the results of a mutual check. Identified modules are used as the kernel of CCS.

It was shown in [3] that for a diagnostic graph $G = (V, E)$ with the symmetry property, the transformation

$$\forall i \in V \left\{ \left[(i, j) \in E \right] \wedge \left[a(i, j) = 1 \right] \rightarrow E = E - \left\{ (i, j), (j, i) \right\} \right\}, \quad (1)$$

is being performed by a graph G_0 , in which the marks of x vertices of each connected component coincide. We choose as t such t^* that for any $F, |F| \leq t^*$, and any generated F syndrome, there exists at least one connected component of order at least $(t^* + 1)$, where t^* is the number of faults. Given the assumption of the multiplicity of faults, it includes vertices that represent serviceable modules, which can therefore be used as the diagnostic core of CCS.

It is necessary for CCS modules to have an ability of mutual transfer of the test results and their comparison in order to allocate the kernel. Any module can be part of the kernel, so the means for performing the specified functions must be in each module. In order to allocate actually serviceable modules as a kernel, the considered means should be serviceable irrespective of the serviceability of the modules being checked. These means constitute a cellular automaton with the CCS structure and form its distributed kernel. The homogeneity of the structure of the control automaton makes it possible to automate the verification of its own state. In relation to the means used, the control device acts as a verifiable system. Thus, the means for automating the determination of the state of CCS have a hierarchical structure. Reliability of means related to adjacent levels of the hierarchy increases from bottom to top. Finiteness of the number of hierarchy levels is assured by the assumption of absolute reliability of its upper level. This property is assigned to the operational staff of CCS.

Theory of algorithm for isolating the diagnostic core

Suppose we are given an Cayley graph $G = (V, E)$ that represents a G group with g_0, g_1, \dots, g_{n-1} generators. By virtue of the symmetry property, each vertex of the G graph has $2n$ number of arcs with the generators and their inverses. Suppose that G is given a set of admissible images of faults and a set of syndromes generated by these images. In this case, the condition $0 \leq |F| \leq t^*$ holds for any $F \in F$. It is possible to find an algorithm by which for any $F \in F$ and any generated syndrome it is possible to determine the vertex markup $X = \{x(i)\}$ for some $W \subseteq V$ such that $W \cap \bar{F} \neq \emptyset$, singling out $W \cap \bar{F} \neq \emptyset$ at least one vertex i such that $i \in (W \cap \bar{F})$.

Let us describe some of the thesis's that form the basis of the kernel extraction algorithm. Any FAIS graph has a fundamental cycle, since the parity of the degrees of its vertices implies the existence of an Euler cycle for it. Consider the following algorithm for traversing the graph. (In the algorithms considered below, the \leftarrow sign is the operator of the left assignment).

Algorithm 1.

Stage 1. Suppose $P(i), T(i), H(i)$ are the marks of the i -th vertex, $i \in V$, such that $P(i) \in \{0, 1, 2, \dots, n-1\}, T(i) \in \{0, 1\}, H(i) \in \{0, 1\}$. The $P(i)$ and $T(i)$ values determine the bypass sequence G , and $H(i) = 1$ selects an arbitrarily chosen initial vertex. The initial value of the indicated marks is determined by the conditions $\forall i \in \{P(i) = 0, T(i) = 1\}; \exists k \in V \{H(k) = 1\}; \forall i \in (V - k) * \{H(i) = 0\}$. $k \in V$ will be the vertex of the graph G considered in the current pass of the algorithm.

Stage 2. If $T(k) = 1$, then $T(k) \leftarrow 0$, otherwise $P(k) \leftarrow P(k) + 1 \pmod{n}$.

Stage 3. Go to the vertex j , which is incident to the arc emanating from k , and has a mark of the generator g_l such that $l = P(k); k \leftarrow j$.

Stage 4. If $[\exists k \in V \{T(k) = 1\} \wedge [H(k) = 0]]$, then go to stage 2, otherwise end.

The condition of the operator 4 is fulfilled when the sequence generated by Algorithm 1 contains all the vertices of G and is closed. Only if this condition is fulfilled, Algorithm 1 is completed.

Assertion 1. Algorithm 1 provides the formation of the main cycle for the FAIS graph G .

Corollary 1. The length of the main cycle of the CAIS graph G formed by algorithm 1 is less than the length of the Euler cycle of this graph.

The proofs of Assertion 1 and Corollary 1 are not given in the paper.

Assertion 2. $G = (V, E)$ - is a random diagnostic graph possessing the symmetry property:

$\forall i, j \in V \{(i, j) \in E \rightarrow (j, i) \in E\}, aS = \{a(i, j) | (i, j) \in E\}$ is a syndrome, caused by an image of malfunctions F . Then the condition $\forall a(i, j), a(i, j) \in S \{a(i, j) = 0 \wedge a(j, i) = 1 \rightarrow i \in F\}$ is fulfilled.

Suppose the contrary: $i \in \bar{F}$. Then it follows from conditions 1 and 2 that $j \in \bar{F}$. But since $a(i, j) \neq a(j, i)$, then $(i \cup j) \cap F \neq \emptyset$. Consequently, $i \in F$ does not depend on whether it is $j \in F$ satisfied or $j \in \bar{F}$, as required to prove.

Corollary 2. Suppose $G^{\wedge} = (V^{\wedge}, E^{\wedge})$ is the connectivity component of a graph G_0 , which is given a value S for $G = (V, E)$, and suppose $i, j \in V^{\wedge}$ such that $i \in V^{\wedge}, (i, j) \in E^{\wedge}$. If the condition of Assertion 2 is satisfied for i and j in G , then $G^{\wedge} \subseteq F$.

The kernel is isolated as a result of a sequential analysis of the elements of the syndrome S , performed when the graph G is traversed. The rules for selecting the next vertex are the same for all vertices of the graph and are given by the following algorithm.

Algorithm 2.

Stage 1. $G^{\wedge} = (V^{\wedge}, E^{\wedge})$, $G^{\wedge} \subset G$ - the generated connectivity component and $k \in V^{\wedge}$. The set of vertices V_k of the graph G adjacent to k is subdivided into disjoint sets of direct V_{kn} and reverse V_{ko} testing directions (analysis of the values of the marks of arcs $a(i, j)$ emanating from k). If $j \in V_k$ does not enter into any of the previously formed connectivity components and the value $a(k, j)$ has not yet been checked, then $j \in V_{kn}$. If $j \in V_k$ and the transition to k is performed from j , then $j \in V_{ko}$. Introduce a mark k for the selection of vertices entering the

connectivity component: if the vertex belongs to some connectivity component, then $K(j)=0$, otherwise $K(j)=1$. Also introduce marks for the separation of arcs of the connectivity component G^{\setminus} : if $(i, j) \in E^{\setminus}$, then $M(i, j)=1$, otherwise $M(i, j)=0$. Define on the set V_{kn} an order relation that corresponds to an increase in the indices of marks g_i of arcs emanating from k and incident to the vertices of V_{kn} . Introduce the service Boolean variables A, B, C , the values of which will be used in Algorithm 3: $A \leftarrow 0; B \leftarrow 0; C \leftarrow 0$; form for $k \in V_{kn}$ and V_{ko} .

Stage 2. If $V_{kn} \neq \emptyset$, then go to stage 3, otherwise - go to stage 6.

Stage 3. Suppose j is another element V_{kn} . If $a(k, j)=0$, then go to stage 4, otherwise to stage 5.

Stage 4. Exclude j from $V_{kn}; K(j) \leftarrow 0; M(k, j) \leftarrow 1$; go to the vertex $j: k \leftarrow j$, form V_{ko} : if $[(V_{ko} = \emptyset \wedge j \in V_k \wedge M(j, k)=1)]$, then $j \in V_{ko}$; form V_{kn} : $\forall i \in V_k - V_{ko} \{K(i)=1 \rightarrow i \in V_{kn}\}$; $A \leftarrow 1$; go to stage 9.

Stage 5. Here $a(k, j)=1$. Delete j from V_{kn} ; go to stage 2.

Stage 6. Here $V_{kn} = \emptyset$. If $V_{ko} \neq \emptyset$, go to stage 7; otherwise, go to stage 8.

Stage 7. Testing is referred by a singleton set V_{ko} (vertex j); delete j from $V_{ko}; B \leftarrow 1$, go to stage 9.

Stage 8. Here $V_{kn} = V_{ko} = \emptyset$. Formation of G^{\setminus} is completed; $C \leftarrow 1$.

Stage 9. The end of the algorithm.

Assertion 3. When using algorithm 2, the generated connectivity component $G^{\setminus} = (V^{\setminus}, E^{\setminus})$ is a tree covering the set of vertices V^{\setminus} . The root is the vertex with which the formation of G^{\setminus} is started. Suppose $k \in V^{\setminus}$ is the vertex from which the analysis of marks $a(k, j); j \in V_k$ begins. Then the following assertions hold:

1. for any G^{\setminus} , a singular oriented simple goal C exists, starting from the root vertex and entering into k , such as $\forall j \in V^{\setminus} (V_{jn} = \emptyset \rightarrow j \in C)$; the reverse statement is not true;

2. for every $j \in C$: if there exists $(j, m) \in E^{\setminus}$ such that $(j, m) \notin C$, then there exists $(m, j) \in E^{\setminus}$;

3. suppose that $j \in V^{\setminus}$ satisfies the conditions of §3.2: exclude from G^{\setminus} the $(j, m)(m, j)$ arcs, denote the $G(m, j) = [V^{\setminus}(m, j), E^{\setminus}(m, j)]$ subgraph of the graph G^{\setminus} , for which $m \in V^{\setminus}(m, j)$. Then $\forall k \in V^{\setminus}(m, j) * \{(k, s) \in E^{\setminus}(m, j) \rightarrow (s, k) \in E^{\setminus}(m, j)\}$.

Corollary 3. Suppose $G^{\setminus}(m, j)$ is a subgraph of a graph G^{\setminus} that satisfies the conditions of §3 of Assertion 3. Then $\forall k \in V^{\setminus}(m, j) \{x(k) = x(j)\}$.

Assertion 4. Suppose C is an arbitrary simple chain defined on the structure of the connectivity component G^{\setminus} . Suppose i and j are the vertices entering into C , while j is accessible from i , i.e. $i < j$. Then, $\forall i, j \in C \{i < j \rightarrow x(i) \geq x(j)\}$, i.e. in C the vertices with marks $x(i)=0$ can not precede the vertices with marks $x(j)=1$.

The proof follows directly from conditions 1 and 2.

Corollary 4. Suppose $L(C)$ is the number of vertices in chain C . If $L(C) = (t^* + 1) + k, k \geq 0$, then C contains no more than t^* vertices with marks t^* preceding the vertices with marks $x(i) = 0$.

The proof follows from the condition of the multiplicity of faults. Suppose $x \in \{0, 1, 2, 3\}$. The values 0 and 1 of the x mark are defined as finite; the $x = 3$ value is the initial value when executing the kernel allocation algorithms; the vertex gets the value $x(i) = 2$, if, based on the results of the module check, the final value of the mark can not be determined.

The connectivity component G^{\setminus} is completed if one of the following conditions is true:

Condition 5. Fulfilled $\forall i \in V^{\setminus} \{(V_{in} = \emptyset) \wedge (V_{io} = \emptyset)\}$. This condition corresponds $C=1$ in the algorithm and 2. If $|V^{\setminus}| < t^* + 1$, then the value x for the vertices V^{\setminus} can not be determined until the CCS kernel is allocated. For such vertices $x(j) = 2$.

Condition 6. Fulfilled $\exists i, j \in V^{\setminus} \{a(i, j) = 0 \wedge a(j, i) = 1\}$. In this case, in accordance with Corollary 2, G^{\setminus} contains the vertices of two connectivity components of the graph $G_0: G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Here $\forall i \in V_1 \{x(i) = 1\}; \forall i \in V_2 \{x(i) = 2\}$, as for any component satisfying the condition $|V^{\setminus}| < (t^* + 1)$.

Condition 7. On passing to the vertex j in the course of direct viewing $|V^{\setminus}| = t^* + 1$ is received. In this case, we have $|V^{\setminus}| = t^* + 1$ for the last vertex included in V^{\setminus} . In accordance with Corollary 2 and Assertion 4, it is possible that $\exists i, j \in V^{\setminus} \{a(j, i) = 1\}$. In this case, G^{\setminus} decomposes into two connectivity components: G_1^{\setminus} , for vertices i of which $x(i) = 1$, and G_0^{\setminus} , for vertices of which $x(i) = 0$. All vertices G_0^{\setminus} form the diagnostic core of the graph.

Next, suppose that for some connectivity component G^{\setminus} , formed during the allocation of the kernel, $\forall i \in G^{\setminus} \{x(i) = 2\}$. Taking into account the property of the graph G_0 , the value of the marks x of all vertices G^{\setminus} is determined by the singular value $a(i, j)$ for such adjacent vertices i and j , that $i \in G^{\setminus}, j \notin G^{\setminus}, x(j) = 0$.

We now describe the kernel allocation algorithm using algorithms 1 and 2. In addition to the entered marks, a variable R will be used in it, the value of which determines the order of the connectivity component being formed: $R \in \{0, 1, 2, \dots, t^* + 1\}$.

Algorithm 3.

Stage 1. The initial values of the marks G and variables are determined by conditions

$$\forall i \in V \{P(i) = 0, T(i) = 1, V_{in} = \emptyset, K(i) = 1, x(i) = 3\};$$

$$\forall (i, j) \in E \{a(i, j) \in S, M(i, j) = 0\}; \exists s \in \{H(s) = 1\}; \forall i \in (v - S) \{H(i) = 0\}.$$

The initial vertex s is chosen arbitrarily. Suppose k is the vertex of the graph considered in the course of the algorithm. Then $k \leftarrow s; R \leftarrow 0$.

Stage 2. Run Algorithm 2 for k . If, $A = 1$ then go to stage 3, if $B = 1$, then go to stage 4; if $C = 1$, then go to stage 6.

Stage 3. $R \leftarrow R + 1$, if $R = t^* + 1$, then go to stage 9, otherwise go to stage 2. Here the order of G^{\setminus} increases, because $j \in V_{kn}$ is included in G^{\setminus} . The transition to stage 9 takes place when the first element of the kernel is defined.

Stage 4. If $a(k, j)$, then go to stage 8, otherwise go to stage 5. Fulfillment of the condition means the completion of the formation of $G^{\dot{}}$ by the condition of approval 3: $G = G_1 \cup G_2$ and $j \in G_1^{\dot{}}, k \in G_2^{\dot{}}$. The component $G^{\dot{}}$ is marked with notations M .

Stage 5. Move to the vertex $j: k \leftarrow j$; go to stage 2.

Stage 6. Here $G^{\dot{}} = G_2^{\dot{}}; H(k) = 1$. For all $i \in V_2^{\dot{}}$ perform $x(i) \leftarrow 2$.

Stage 7. Go to the vertex i of the component $G^{\dot{}}$, for which $H(i) = 1; k \leftarrow i; H(k) \leftarrow 0$; execute Algorithm 1 (stages 2, 3) until a vertex k is found, for which $K(k) = 1; K(k) \leftarrow 0; H(k) \leftarrow 1; R \leftarrow 0$; go to stage 2. As a result of the described sequence of actions, the mark H is transferred to a vertex that does not belong to one of the previously formed connectivity components.

Stage 8. For all $i \in V_1^{\dot{}}$ execute $x(i) \leftarrow 1$; for all $i \in V_2^{\dot{}}$ perform $x(i) \leftarrow 2$; go to stage 7.

Stage 9. $x(k) \leftarrow 0; K(k) \leftarrow 0$. Diagnostics begins with determining the final values of the marks x for the vertices of $G^{\dot{}}$. They are determined as a result of an analysis of the values of the marks a of arcs opposite to the arcs of a simple oriented chain C , which is starting from vertices s with a mark $H(s) = 1$ and entering the vertex k (see Corollary 4). We regard the component $G^{\dot{}}$ as a collection of subtrees $L(C)$, copies of which are vertices of the chain C . Each of these trees satisfies the conditions of section 3 of Assertion 3, and Corollary 3 holds for each of them.

Stage 10. For all the vertices i from $G^{\dot{}}$, included in the sub-tree with root in k , execute $x(i) \leftarrow 0$.

Stage 11. If $V_{ko} = \emptyset$, go to stage 15. The condition is fulfilled when the root vertex $G^{\dot{}}$ is reached.

Stage 12. One-element set $V_{ko} = \{j\}$ is to be tested. If $a(k, j) = 1$, then go to stage 14, otherwise go to stage 13. Fulfillment of the condition means that in accordance with Corollary 2 $j \in V_1^{\dot{}}$.

Stage 13. $k \leftarrow j; x(k) \leftarrow 0$ go to stage 10.

Stage 14. $k \leftarrow j; x(k) \leftarrow 1$ for all vertices $i \in V_1^{\dot{}}$ execute $x(i) \leftarrow 1$.

Stage 15. The analysis of $G^{\dot{}}$ is complete. All $i \in V_0^{\dot{}}$ are involved in the diagnosis. Diagnosis consists in determining the belonging to F of the elements, not included in $G^{\dot{}}$. The exceptions are the vertices for which Assertion 2 is satisfied. For diagnostics, it is necessary to perform an analysis k of the elements S , where $K \leq N - |V^{\dot{}}|$. The inequality is valid if there are vertices for which Assertion 2 is satisfied, and also in connection with the fact that the determination of the state of all elements from $G_2^{\dot{}}$, which are formed when the kernel is allocated, is performed by the singular value $a(i, j)$ of the arc weight (i, j) , if $i \notin G_2^{\dot{}}, j \in G_2^{\dot{}}, i \in F$ (Corollary 2).

Result and discussion

It is easy to see that Algorithm 3 allocates the kernel of CCS in a finite number of stages. It follows from the above description of Algorithms 1-3 that they allow a decentralized implementation: changing the value of the vertex and arc marks, as well as the rules for traversing the graph are determined only by the values of the marks of adjacent elements.

To obtain estimates of the effectiveness of the heuristic Algorithm 3, the method of simulation-statistical modeling is used. The structure of the diagnostic graph $G = (V, E)$ of the system and the power r of the fault sample, having values from 1 to t^* , are set. Using r as a parameter is allowing to obtain the potential characteristics of Algorithm 3. For each $r \in \{1, 2, \dots, t^*\}$ the number of experiments B is performed, in each of which an image of faults $F_r, |F_r| = r$, the syndrome generated by it, and the vertex with a mark C are arbitrarily chosen.

Events that consist that $i \in F_r$ or \bar{i} have a mark H take place with probability $1/N$ and do not depend on the value r and composition F_r . The values $a(i, j)$ for the graph G are determined for the chosen F_r , with the condition that when $i \in F_r$, then $a(i, j)$ takes the value 0 and 1 with probability $p(0)$ and $p(1)$, respectively.

Algorithm 3 is divided into three stages:

1. The allocation of the kernel - from the initialization of the algorithm to the selection of the first vertex with a mark $x = 0$;
2. Determining the value of the vertex marks for the connectivity component of the order $t^* + 1$;
3. Determining the marks of the remaining vertices of the graph.

During the simulation, the data are summarized to determine the integral potential characteristics of the algorithm's efficiency, as well as to assess the practicability of implementing its individual elements (practical completeness of the algorithm). The latter is due to the fact that under the conditions of practical application, some elements of Algorithm 3 can be performed rather rarely because of the low probability of the occurrence of events causing their execution.

To do this, integral characteristics of the algorithm are introduced.

1. Vector coefficient $K_k = \{K_{kr}\}, r \in \{1, 2, \dots, t^*\}$ of the relative overhead expenses for the allocation of the CCS kernel, the coordinates of which are the values $K_{kr} = [M_r(T_k) - t^*] / t^*$. Here the multiplicity of faults t^* coincides with the minimum number of checks of values $a(i, j)$ (tests) for the allocation of the kernel; $M_r(T_k)$ - the average number of tests required to perform the first mark $x = 0$, when $|F_r| = r$.

2. Vector coefficient $K_e = \{K_{er}\}, r \in \{1, 2, \dots, t^*\}$ of the relative expenses of determining the state of CCS, the coordinates of which are the values $K_{er} = M_r(T_c) / |E|$, where $M_r(T_c)$ is the average number of tests required to determine the state of CCS, when $|F_r| = r$.

For FAIS graphs $|E| = 2vN$ (here $v = 2$). Obviously, the smaller the values K_{er} are, the more efficient the algorithm.

To evaluate the conditions for the practical completeness of the algorithm, vector coefficients are used whose coordinates are the values of the characteristics of the following events, processed by Algorithm 3: completion of the formation of the connectivity component in stage 1 by the formation G_2^{\wedge} or formation G_1^{\wedge} and G_2^{\wedge} ; in stage 2, by formation G_0^{\wedge} and G_1^{\wedge} ; using Algorithm 1. The completion of the formation of connectivity components is characterized by the following indicators.

1. Vector coefficient $K_m = \{K_{mr}\}, r \in \{1, 2, \dots, t^*\}$ of the relative coverage of the image of malfunctions with the power r . The coordinates K_n are the values $K_{mr} = M_r(A_1) / r$, where $M_r(A_1)$ is the average number of vertices i , such that $i \in F_r$ received marks $x(i) = 1$ or $x(i) = 2$ in either stage 1 or 2 of Algorithm 3, $M_r(A_1) = \sum_{i=1}^3 M_{r_i}(A_1); M_{r_1}(A_1), M_{r_2}(A_1)$ and

$M_{r3}(A_1)$ the average number of vertices from F_r , which are included in the components of the species G_1 and G_2 in stages 1,2,1 respectively.

2. Vector coefficient $K_{m2} = \{K_{m2r}\}, r \in \{1, 2, \dots, t^*\}$ of covering the image of malfunctions in stage 2. The coordinates K_{m2} are the values $M_{r2}(A_1)$.

3. Vector coefficient $K_{cm} = \{K_{c2}\}, r \in \{1, 2, \dots, t^*\}$ of the possibility of completing stage 2 by forming G_0 and G_1 . The components of the vector coefficient are the probabilities of occurrence of the specified event.

So, FAIS graphs representing CCS of N modules are investigated, $13 \leq N \leq 90$. On fig. 1 and fig. 2 the characteristics of a set of graphs are given, the order of which corresponds to the values $N = 13, 20, 25, 30, 40, 50, 60, 70, 80, 90$; multiplicity of faults for them $t_{13}^* = t_{20}^* = 4$; $t_{25}^* = t_{30}^* = t_{40}^* = t_{50}^* = 12$; $t_{60}^* = t_{70}^* = t_{80}^* = 16$; $t_{90}^* = 20$. Defining relations (except those which are setting commutativity) for the group description of these graphs have the form $(g_0^{13} = I, g_1 = g_0^5)$; $(g_0^{20} = I, g_1 = g_0^8)$; $(g_0^{25} = I, g_1 = g_0^7)$; $(g_0^{30} = I, g_1 = g_0^6)$; $(g_0^{40} = I, g_1 = g_0^5)$; $(g_0^{50} = I, g_1 = g_0^6)$; $(g_0^{60} = I, g_1 = g_0^{10})$; $(g_0^{70} = I, g_1 = g_0^{22})$; $(g_0^{80} = I, g_1 = g_0^{12})$. Here I is the singular element of the group.

In fig. 1 and fig. 2, the coordinates of the vector coefficients are shown as functions of r for different values N . The investigated functions are monotonically increasing from r and monotonically decreasing from N .

From fig. 1 and fig. 2, where the functions K_k and K_e are respectively shown, it is clear that the efficiency of the algorithm increases with the number of modules in CCS. The number of tests required for the diagnosis of CCS does not exceed the value $0,4|E|$.

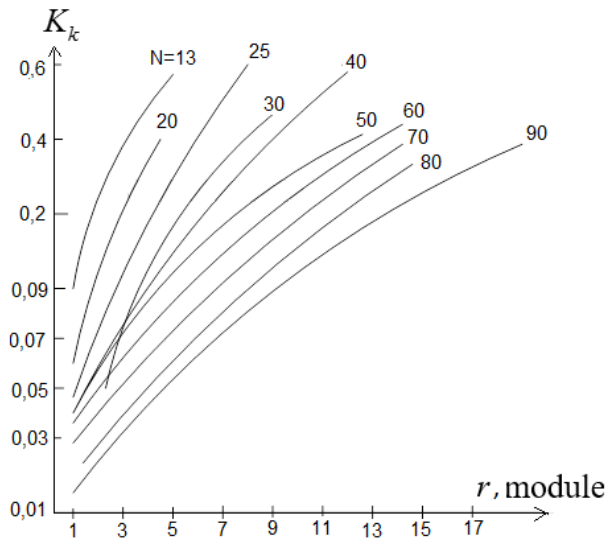


Fig. 1. Characteristics of a set of graphs from the function K_k

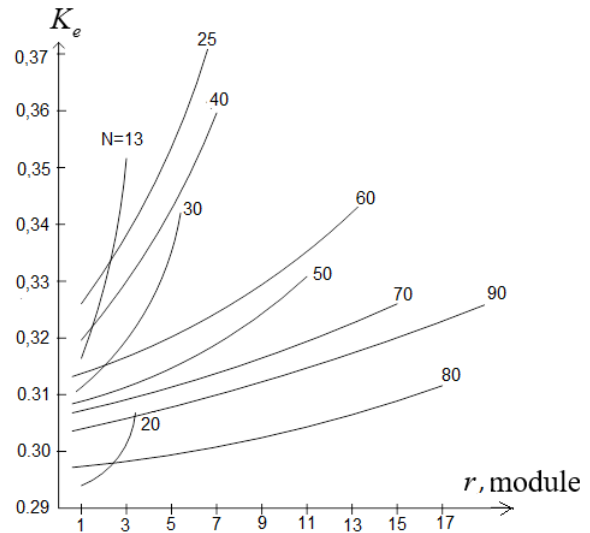


Fig. 2. Characteristics of a set of graphs from the function K_e

Fig. 2 shows the essential dependence of the costs of diagnostics on the relationship t^*/N .

The simulation results show that no less than $0,9r$ faulty CCS modules are determined at the stage 3 of the algorithm being investigated. The contribution to the value of the coordinates of the vector coefficient K_m is determined by the ratio

$M_{1r}(A_1):M_{2r}(A_1):M_{3r}(A_1)=10^{-3}:1:10^{-1}$, i.e. the main contribution to the value of coordinates K_n is made by the completion of stage 2 by the formation of G_0^* and G_1^* .

Studies have shown that K_{m2} is a nonlinearly increasing function of r . The nonlinearity depends on the probability that for stage 2 $|V_1^*|>1$ (at $r=t^*/2$ this probability for the investigated structures is not lower than (0.15)). The dependence of K_{c2r} on r is close to a linear function $K_N r$. The coefficient K_N corresponds to the value obtained by modeling the value of the first coordinate K_{02} .

A typical sequence of execution of Algorithm 3 can be represented as follows. In most cases, the allocation of the connectivity component that makes up the kernel is performed on the first attempt. There is a possibility of one or two faulty modules present in this component. One of them is the root module of the subsystem. The structure of the component making up the kernel is in most cases a simple chain, which ensures the maximum number of kernel modules involved in determining the state of CCS. Usually, in stage 1, one or two connectivity components are formed $G_2^*; 1 \geq |V_2^*| \geq 2$.

The formation of the connectivity component in stage 1 is rarely completed by the formation of G_1^* and G_2^* . If such an event takes place, then $|V_1^*| \leq 3$. Other values for the examined graphs were not observed.

Conclusions

Thus, the conditions for automating the determination of the state of serviceability of durable modular CCS are investigated, which are t-diagnosed with repair. The way to determine their state is the modification of the known method of promotion. For CCS with a software-configurable structure, the concept of a diagnostic kernel is concretized, a heuristic decentralized kernel allocation algorithm is described. The efficiency of the proposed self-diagnostic algorithm is shown by the imitation-statistical modeling method. It has been analytically established that, without prejudice to the practical use of it, reverse testing for the connectivity components in stage 1 of the mark H transfer algorithm and the means associated with the formation of the structure G_2^* can be excluded from it.

References

1. Артамонов, Г.Т. Топология сетей ЭВМ и многопроцессорных систем / Г.Т. Артамонов, В.Д. Тюрин. — М.: Радио и связь, 1991. — 248 с.
2. Mariño, P.P. Optimization of Computer Networks: Modeling and Algorithms: A Hands-On Approach 2016. — Pp.1-375. DOI: 10.1002/9781119114840
3. Danilin I.S, Guseev L.I, Zagorodny Y.I. Quality assurance of REA by methods of diagnostics and forecasting. Moskva: Izd. Standards, 1983. — 224 p. [in Russian]
4. Wolf, M. The Physics of Computing, 2016, 1-264
5. Harari F. Theory of graphs. Mir: Moskva, 1973. — 300 p. [in Russian]
6. Koh, K.M., Dong, F., Ng, K.L., Tay, E.G. Graph Theory: Undergraduate Mathematics, 2015. — Pp. 1-479. DOI: 10.1142/9426
7. Gulyaev V.A. Organization of diagnostic systems for computers. Science. Dumka: Kyiv, 1979. — 116 p. [in Ukrainian]
8. Ligeza, A. Towards knowledge compilation for automated diagnosis: A qualitative, model-based approach with constraint programming. Advanced and Intelligent Computations in Diagnosis and Control, 2015. — Pp. 355-367. DOI: 10.1007/978-3-319-23180-8_26
9. Henley E.J., Kumamoto H. Reliability of technical systems and risk assessment. Mechanical Engineering: Moskva, 1984. — 528 p. [in Russian]

10. Verma, A.K., Ajit, S., Karanki, D.R. Reliability and Safety Engineering: Second Edition, 2015. — Pp. 1-571. DOI: 10.1007/978-1-4471-6269-8
11. Todinov, M. Reliability and Risk Models: Setting Reliability Requirements: Second Edition, 2015. — Pp. 1-419. DOI: 10.1002/9781118873199

САМОДІАГНОСТИКА КОМПЛЕКСНИХ СИСТЕМ З ПРОГРАМНО-КОНФІГУРОВАНОЮ СТРУКТУРОЮ

I.V. Коробійчук¹, Р.В. Гришук², В.О. Хорошко³, Ю.Є. Хохлачова³

¹Інститут автоматичного управління та робототехніки, Варшавський технологічний університет, пл. Політехніки, 1, Варшава, 00-661, Польща; e-mail: igor@mchtr.pw.edu.pl

²Житомирський військовий інститут ім. С. Корольова, просп. Миру, 22, корп. ЖВІ, Житомир, 10004, Україна; e-mail: dr.hry@i.ua

³Національний авіаційний університет, просп. Космонавта Комарова, 1, Київ, 03058, Україна; e-mail: professor_va@ukr.net, hohlachova@gmail.com

За допомогою методу імітаційно-статистичного моделювання встановлено працездатність і отримані оцінки ефективності запропонованого алгоритму самодіагностики складних обчислювальних систем з програмно-конфігурованою структурою. Структура складних обчислювальних систем є діаграмою Келі деякої кінцевої абелевих групи дуги в діаграмі Келі мають позначки утворюють групи. Описані графи названі КАІС-графами (графи для надання кінцевих абелевих ізотропних і симетричних структур). Через обмежену кількість міжмодульних зв'язків складних обчислювальних систем для визначення її стану необхідно використовувати робочі зв'язки між модулями. Отже, діагностичний граф G для складних обчислювальних систем є КАІС-граф. Властивість симетричності КАІС-графа означає, що сусідні модулі складних обчислювальних систем є взаємотестуємими. Вірне визначення стану складних обчислювальних систем можливо, якщо воно здійснюється свідомо справними засобами, що є складовими ядра складних обчислювальних систем. Визначимо діагностичне ядро як сукупність засобів, що мають більш високу надійність, ніж система, яка перевіряється з його допомогою. Надійність ядра забезпечується засобами, які не належать системі, що ним перевіряється. Одним з методів автоматизації визначення стану сучасних складних обчислювальних систем є метод розкрутки. На кожному кроці розкрутки справність одних вузлів визначається справністю інших, яка встановлена на попередніх кроках. Процес розкрутки завжди починається з ядра системи. Функціональні характеристики модулів складних обчислювальних систем такі, що будь-який з них можна використовувати в якості ядра. Якщо $t < N$, то завдання визначення стану системи можна уточнити: за результатами взаємної перевірки виявити хоча б один справний модуль. Виявлені модулі використовуються в якості ядра складних обчислювальних систем. Способом визначення стану справності модульних живучих складних обчислювальних систем є модифікація відомого методу розкрутки. Для складних обчислювальних систем з програмно-конфігурованою структурою конкретизується поняття діагностичного ядра, описаний евристичний децентралізований алгоритм виділення ядра.

Ключові слова: складні обчислювальні системи, діагностична модель, діагностичний граф, алгоритм, витрати на діагностику

**САМОДИАГНОСТИКА КОМПЛЕКСНЫХ СИСТЕМ
С ПРОГРАММНО-КОНФИГУРИРОВАННОЙ СТРУКТУРОЙ**И.В. Коробийчук¹, Р.В. Гришук², В.А. Хорошко³, Ю.Е. Хохлачева³¹Институт автоматического управления и робототехники, Варшавский технологический университет, пл. Политехники, 1, Варшава, 00-661, Польша; e-mail: igor@mchtr.pw.edu.pl²Житомирский военный институт им. С. Королева, просп. Мира, 22, корп. ЖВИ, Житомир, 10004, Украина; e-mail: dr.hry@i.ua³Национальный авиационный университет, просп. Космонавта Комарова, 1, Киев, 03058, Украина; e-mail: professor_va@ukr.net, hohlachova@gmail.com

С помощью метода имитационно-статистического моделирования установлена работоспособность и получены оценки эффективности предложенного алгоритма самодиагностики сложных вычислительных систем с программно-конфигурируемой структурой. Структура сложных вычислительных систем является диаграммой Кэли некоторой конечной абелевой группы дуги в диаграмме Кэли имеют отметки образующих группы. Описанные графы названы КАИС-графами (графы для предоставления конечных абелевых изотропных и симметричных структур). Из-за ограниченного числа межмодульных связей сложных вычислительных систем для определения ее состояния необходимо использовать рабочие связи между модулями. Следовательно, диагностический граф G для сложных вычислительных систем есть КАИС-граф. Свойство симметричности КАИС-графа означает, что соседние модули сложных вычислительных систем взаимотестируемы. Правильное определение состояния сложных вычислительных систем возможно, если оно осуществляется заведомо исправными средствами, составляющими ядро сложных вычислительных систем. Определим диагностическое ядро как совокупность средств, имеющих более высокую надежность, чем проверяемая с его помощью система. Надежность ядра обеспечивается средствами, которые не принадлежат проверяемой им системе. Одним из методов автоматизации определения состояния современных сложных вычислительных систем является метод раскрутки. На каждом шаге раскрутки исправность одних узлов определяется исправностью остальных, которая установлена на предыдущих шагах. Процесс раскрутки всегда начиналась с ядра системы. Функциональные характеристики модулей сложных вычислительных систем таковы, что любой из них можно использовать в качестве ядра. Если $t < N$, то задачу определения состояния системы можно уточнить: по результатам взаимной проверки выявить хотя бы один заведомо исправный модуль. Выявленные модули используются в качестве ядра сложных вычислительных систем. Способом определения состояния исправности модульных живучих сложных вычислительных систем является модификация известного метода раскрутки. Для сложных вычислительных систем с программно-конфигурируемой структурой конкретизируется понятие диагностического ядра, описан эвристический децентрализованный алгоритм выделения ядра.

Ключевые слова: сложные вычислительные системы, диагностическая модель, диагностический граф, алгоритм, затраты на диагностику