

Accepted Manuscript

Diagnosis of three types of constant faults in read-once contact networks over finite bases

Monther Busbait, Mikhail Moshkov

PII: S0304-3975(16)00234-6
DOI: <http://dx.doi.org/10.1016/j.tcs.2016.03.023>
Reference: TCS 10703

To appear in: *Theoretical Computer Science*

Received date: 13 September 2015
Revised date: 16 January 2016
Accepted date: 18 March 2016

Please cite this article in press as: M. Busbait, M. Moshkov, Diagnosis of three types of constant faults in read-once contact networks over finite bases, *Theoret. Comput. Sci.* (2016), <http://dx.doi.org/10.1016/j.tcs.2016.03.023>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



Diagnosis of Three Types of Constant Faults in Read-Once Contact Networks over Finite Bases

Monther Busbait, Mikhail Moshkov*

Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia

Abstract

We study the depth of decision trees for diagnosis of three types of constant faults in read-once contact networks over finite bases containing only indecomposable networks. For each basis and each type of faults, we obtain a linear upper bound on the minimum depth of decision trees depending on the number of edges in networks. For bases containing networks with at most 10 edges, we find sharp coefficients for linear bounds.

Keywords: read-once contact networks, constant faults, decision trees

1. Introduction

In this paper, we study read-once contact networks representing Boolean functions. A read-once contact network is an undirected graph with two poles in which pairwise different Boolean variables are assigned to edges. For a given tuple of variable values, the value of the function corresponding to the network is equal to 1 if and only if there exists a path between poles such that the value of each variable assigned to the edges in this path is equal to 1. Note that the contact networks are also known as contact schemes, contact circuits, and switching circuits [5, 10, 19].

We study read-once networks over a finite set B of networks which is called basis. These networks can be obtained by replacing some edges in a network from B with recursively constructed networks (including networks

*Corresponding author. Tel.: +966 12 808 0334; fax: +966 12 802 1291.

Email address: mikhail.moshkov@kaust.edu.sa (Mikhail Moshkov)

from B). We assume that all networks from B are indecomposable, i.e., cannot be obtained from two nontrivial (containing more than one edge) networks by replacing an edge in the first network with the second one.

Let $C \in \{\{0, 1\}, \{0\}, \{1\}\}$. We consider C -faults of read-once networks each of which consists in assigning constant values from C to some of the network variables. We study the problem of diagnosis of C -faults: for a given faulty network, we should recognize the function implemented by this network. To solve this problem we use decision trees with membership queries: we can ask about value of the function implemented by a given faulty network on an arbitrary tuple of values of the network variables. The depth of a decision tree is the maximum number of nonterminal nodes (queries) in a path from the root to a terminal node.

For a read-once network S over B , we denote by $L(S)$ the number of edges in S and by $h_C(S)$ – the minimum depth of a decision tree for diagnosis of C -faults for S . We prove that $h_C(S) \leq t_C(B)(L(S) - 1)$ for any read-once network S over B , where $t_C(B) = \max\{t_C(Q) : Q \in B\}$ and $t_C(Q) = \frac{h_C(Q)}{L(Q)-1}$ for any network Q from the set B . This result was presented for $C = \{0, 1\}$ in [13] without proof and in [4] – with proof.

We also consider the function $H_B^C(n)$ which characterizes the dependence in the worst case of the minimum depth of a decision tree for diagnosis of C -faults for a network over B on the number n of edges in the network. In [4], we proved that $H_B^{\{0,1\}}(n) = \Theta(n)$ for any basis B . In this paper, we prove that $H_B^{\{0\}}(n) = 1$ for $n \geq 2$ if $B = \{S_2^2\}$ (S_2^2 is a network consisting of simple path of length two connecting two poles), and $H_B^{\{0\}}(n) = \Theta(n)$ if $B \neq \{S_2^2\}$. We prove also that $H_B^{\{1\}}(n) = 1$ for $n \geq 2$ if $B = \{S_1^2\}$ (S_1^2 is a network with two poles and two parallel edges connecting the poles), and $H_B^{\{1\}}(n) = \Theta(n)$ if $B \neq \{S_1^2\}$.

Kuznetsov in [11] published the list of all nontrivial indecomposable networks with at most 10 edges. The value $h_{\{0,1\}}(Q)$ was studied in [7] for each indecomposable network Q containing at most 8 edges. In [4], we obtained upper and lower bounds for the value $h_{\{0,1\}}(Q)$ for each indecomposable network Q with at most 10 edges. The difference between lower and upper bounds on the value $h_{\{0,1\}}(Q)$ was at most one. In this paper, we find exact value of $h_C(Q)$ for each $C \in \{\{0, 1\}, \{0\}, \{1\}\}$ and for each indecomposable network Q with at most 10 edges. To find these values, we use simple lower bound, dynamic and greedy algorithms for decision tree construction, and a new algorithm which has some common points with branch and bound tech-

nique. The obtained results allow us to find the value $t_C(B)$ for an arbitrary basis B containing only indecomposable networks with at most 10 edges.

As it was mentioned in [4], there are three directions of research which are connected with the topic of our paper. The first direction is related to the study of diagnostic tests for constant faults in read-once networks [12]. A diagnostic test is a set of membership queries which allows us to solve the problem of diagnosis for constant faults. The second direction is the study of diagnostic tests and decision trees for constant faults in iteration-free combinatorial circuits (read-once formulas) over finite bases of Boolean functions [8, 9, 14, 15]. The third direction is connected with two problems of learning: learning of read-once contact networks [18], and learning of read-once formulas over some basis [3].

The remainder of this paper is organized as follows. In Section 2, we consider main notions and prove a simple lower bound on the depth of decision trees for fault diagnosis. In Section 3, we study the problem of diagnosis of read-once networks over an arbitrary basis B . In Section 4, we find value $h_C(Q)$ for each nontrivial indecomposable network Q with at most 10 edges and each $C \in \{\{0, 1\}, \{0\}, \{1\}\}$, and describe tools which are used to find $h_C(Q)$. Section 5 contains short conclusions.

2. Main Notions

In this section, we consider main notions connected with networks, constant faults and decision trees for fault diagnosis. We also prove a simple lower bound on the depth of decision trees for fault diagnosis.

A *network* is an undirected graph S with multiple edges and without loops in which two different nodes called *poles* are fixed. For each edge e in S , there is a *simple* path (without repeating nodes) between poles which contains e . We will assume that the set of edges of S is ordered. We denote by $L(S)$ the number of edges in the network S . A network containing only two nodes and one edge connecting these nodes is called *trivial*. Various examples of networks can be found in Tables 2-10.

For a natural number n , a function $f(x_1, \dots, x_n)$ such that $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is called a *Boolean* function. This function is called *monotone* if, for any n -tuples $a = (a_1, \dots, a_n)$ and $b = (b_1, \dots, b_n)$ from $\{0, 1\}^n$, if $a_1 \leq b_1, \dots, a_n \leq b_n$ then $f(a) \leq f(b)$. A variable x_i of the function f is called *essential* if there exist two tuples $a, b \in \{0, 1\}^n$, which are different only in the i -th digit and for which $f(a) \neq f(b)$.

Let S be a network with $L(S) = n$ edges e_1, \dots, e_n . We assign to these edges, variables x_1, \dots, x_n , respectively, and define a Boolean function $f_S(x_1, \dots, x_n)$ implemented by S . Let $a_1, \dots, a_n \in \{0, 1\}$ be values of variables x_1, \dots, x_n , respectively. Then $f_S(a_1, \dots, a_n) = 1$ if and only if there is a path between poles of S in which all variables corresponding to edges in the path have value 1. One can show that f_S is a monotone function, and all variables of this function are essential.

We consider three types of constant faults: $\{0, 1\}$ -faults, $\{0\}$ -faults, and $\{1\}$ -faults. Let $C \in \{\{0, 1\}, \{0\}, \{1\}\}$. A C -fault of S consists in assigning of constants from C to some variables of f_S . More formally, a C -fault of S is an n -tuple $\rho = (\rho_1, \dots, \rho_n) \in \{C \cup \{2\}\}^n$. For $i = 1, \dots, n$, the value ρ_i is called the value of the fault ρ for the variable x_i . The network S with the fault ρ implements the function

$$f_{S,\rho}(x_1, \dots, x_n) = f_S(x_1 \circ \rho_1, \dots, x_n \circ \rho_n)$$

where

$$x_i \circ \rho_i = \begin{cases} 0, & \rho_i = 0 \\ 1, & \rho_i = 1 \\ x_i, & \rho_i = 2 \end{cases}$$

for $i = 1, \dots, n$. Note that $f_{S,\rho}$ is a monotone function. A C -fault ρ is called *normal* if, for $i = 1, \dots, n$, $\rho_i = 2$ if and only if the variable x_i is an essential variable of the function $f_{S,\rho}$. One can show that, for any C -fault δ of S , there exists a normal C -fault ρ of S such that $f_{S,\rho} = f_{S,\delta}$.

The *problem of diagnosis of C -faults for S* : for the network S with C -fault δ of S , we should find a normal C -fault ρ of S such that $f_{S,\rho} = f_{S,\delta}$. To resolve this problem we can ask about values of the function $f_{S,\delta}$ on arbitrary tuples from $\{0, 1\}^n$.

Algorithms for solving the problem of diagnosis of C -faults can be represented in the form of *decision trees* each of which is a directed tree with root. Terminal nodes of the tree are labeled with normal C -faults of S . Each nonterminal node is labeled with an n -tuple from $\{0, 1\}^n$. Two edges start in this node that are labeled with 0 and 1, respectively. The *depth* $h(\Gamma)$ of a decision tree Γ is the maximum length of a path from the root to a terminal node (the *length* of a path is the number of edges in this path).

Let Γ be a decision tree. For the network S with a C -fault δ , this tree works in the following way. If the root of Γ is a terminal node then the outcome of Γ is the normal fault attached to the root. Otherwise, we find

the value of the function $f_{S,\delta}$ on the n -tuple attached to the root and pass along the edge which starts in the root and is labeled with this value, etc., until we reach a terminal node. The normal fault attached to this node is the outcome of Γ . We will say that Γ *solves* the problem of diagnosis of C -faults for S if, for any C -fault δ of S , the outcome of Γ is a normal C -fault ρ of S such that $f_{S,\rho} = f_{S,\delta}$. We denote by $h_C(S)$ the minimum depth of a decision tree which solves the problem of diagnosis of C -faults for S .

We consider now a lower bound on $h_C(S)$. We denote $F_C(S) = \{f_{S,\rho} : \rho \in (C \cup \{2\})^n\}$. The set $F_C(S)$ contains all functions defined on $\{0, 1\}^n$ that can be obtained from the function f_S implemented by S by assigning of constants from C to some variables of f_S . This set includes the function f_S as well.

Lemma 1. *For any $C \in \{\{0, 1\}, \{0\}, \{1\}\}$ and any read-once contact network S , the following inequality holds:*

$$h_C(S) \geq \lceil \log_2 |F_C(S)| \rceil.$$

PROOF. Let Γ be a decision tree which solves the problem of diagnosis of C -faults for S and for which $h(\Gamma) = h_C(S)$. For each function $f \in F_C(S)$, in the tree Γ there is a terminal node which is labeled with a fault ρ for S such that $f = f_{S,\rho}$. Therefore, the number of terminal nodes in Γ is at least $|F_C(S)|$. One can show that the number of terminal nodes in Γ is at most $2^{h(\Gamma)}$. Thus $|F_C(S)| \leq 2^{h_C(S)}$ and $h_C(S) \geq \log_2 |F_C(S)|$. Since $h_C(S)$ is an integer, we have $h_C(S) \geq \lceil \log_2 |F_C(S)| \rceil$. \square

3. Diagnosis of Decomposable Networks

In this section, we study the problem of diagnosis of C -faults for decomposable networks which can be constructed from a finite set of indecomposable networks.

Recall that a network containing only two nodes and one edge connecting these nodes is called trivial. A network S is called *decomposable* if there are two nontrivial networks P and Q , and an edge e in the network P such that S can be obtained by replacing the edge e with the network Q . During the replacement, the ends of e are identified with the poles of Q , the edge e is removed, and the poles of P become the poles of S . The edges of S are ordered in the following way: first there are all edges of P with the exception

of e in the same order as in P and after that all edges of Q in the same order as in Q . If a network is not decomposable, it is called *indecomposable*.

Let B be a finite nonempty set of indecomposable nontrivial networks. We denote $\text{Net}_1(B) = B$ and, for any natural $r \geq 2$, we denote by $\text{Net}_r(B)$ the set of all networks $S \notin \text{Net}_1(B) \cup \dots \cup \text{Net}_{r-1}(B)$ such that S can be obtained by replacing some edges in a network from B with networks from $\text{Net}_1(B) \cup \dots \cup \text{Net}_{r-1}(B)$. The set $\text{Net}(B) = \text{Net}_1(B) \cup \text{Net}_2(B) \cup \dots$ is called the set of *networks over B* .

We now consider an upper bound on the minimum depth of decision trees for diagnosis of C -faults for networks over B depending on the number of edges in the network. We denote $t_C(B) = \max\{t_C(Q) : Q \in B\}$ where $t_C(Q) = \frac{h_C(Q)}{L(Q)-1}$ for any $Q \in B$. Recall that $L(S)$ is the number of edges in a network S , and $h_C(S)$ is the minimum depth of a decision tree which solves the problem of diagnosis of C -faults for S .

Theorem 2. *For any $C \in \{\{0, 1\}, \{0\}, \{1\}\}$ and any network $S \in \text{Net}(B)$, the following inequality holds:*

$$h_C(S) \leq t_C(B)(L(S) - 1).$$

PROOF. We will prove the considered inequality by induction on index r such that $S \in \text{Net}_r(B)$. Let $S \in \text{Net}_1(B)$. Then $h_C(S) = \frac{h_C(S)}{L(S)-1}(L(S) - 1) = t_C(S)(L(S) - 1) \leq t_C(B)(L(S) - 1)$.

Let now $r \geq 2$ and, for any network from $\text{Net}_1(B) \cup \dots \cup \text{Net}_{r-1}(B)$, the considered inequality hold. Let us prove that the considered inequality holds for any $S \in \text{Net}_r(B)$. Since $r \geq 2$, there exist networks $P_0 \in B$ and $P_1, \dots, P_k \in \text{Net}_1(B) \cup \dots \cup \text{Net}_{r-1}(B)$, $k \leq L(S)$, such that S can be obtained by replacing k edges in the network P_0 with the networks P_1, \dots, P_k . It is clear that $L(S) = L(P_0) + (L(P_1) - 1) + \dots + (L(P_k) - 1)$.

We proved that $h_C(P_0) \leq t_C(B)(L(P_0) - 1)$. By induction hypothesis, $h_C(P_i) \leq t_C(B)(L(P_i) - 1)$ for $i = 1, \dots, k$. For $i = 0, 1, \dots, k$, let Γ_i be a decision tree which solves the problem of diagnosis of C -faults for P_i and for which $h_C(\Gamma_i) \leq t_C(B)(L(P_i) - 1)$. We consider now a decision tree Γ which solves the problem of diagnosis of C -faults for S . We describe the work of Γ for the network S with a C -fault δ . As a result, we obtain a normal C -fault ρ for S such that $f_{S,\rho} = f_{S,\delta}$.

First, the decision tree Γ simulates in some way the work of the decision tree Γ_0 . The network P_0 (as a skeleton of S) is analyzed at that. Let P_0

contain $m = L(P_0)$ edges, e_1, \dots, e_m . Let a nonterminal node of the decision tree Γ_0 be labeled with an m -tuple $(a_1, \dots, a_m) \in \{0, 1\}^m$. For $i = 1, \dots, m$, we assign the value a_i to some variables of S (we denote the set of these variables X_i). If the edge e_i was not replaced with a network then we assign the value a_i to the variable corresponding to e_i in S (X_i contains only this variable). If the edge e_i was replaced with a network P_j then we assign the value a_i to all variables corresponding to edges of the network P_j in S (X_i contains all these variables). The value of the function implemented by P_j with the fault δ will be equal a_i provided that this function is not constant.

Let a normal C -fault $\gamma = (\gamma_1, \dots, \gamma_m)$ for P_0 be the outcome of the decision tree Γ_0 . For $i = 1, \dots, m$, we fix values of ρ for some variables of f_S . To this end, the decision tree Γ will simulate the work of some decision trees Γ_j , $j \in \{1, \dots, k\}$.

If $\gamma_i \neq 2$, then the value of ρ is equal to γ_i for each variable from X_i . Let $\gamma_i = 2$. If X_i contains only one variable then the value of ρ for this variable is equal to 2. Let X_i contain two or more variables. Then the edge e_i was replaced in S with a network P_j , $j \in \{1, \dots, k\}$. Since $\gamma_i = 2$, there are values $b_1, \dots, b_{i-1}, b_{i+1}, \dots, b_m \in \{0, 1\}$ such that $f_{P_0, \gamma}(b_1, \dots, b_{i-1}, 0, b_{i+1}, \dots, b_m) \neq f_{P_0, \gamma}(b_1, \dots, b_{i-1}, 1, b_{i+1}, \dots, b_m)$. Now, the decision tree Γ simulates the work of the decision tree Γ_j . Let P_j contain $l = L(P_j)$ edges $\varepsilon_1, \dots, \varepsilon_l$. Let a nonterminal node of the decision tree Γ_j be labeled with an l -tuple $(a_1, \dots, a_l) \in \{0, 1\}^l$. We assign values a_1, \dots, a_l to variables corresponding in S to the edges $\varepsilon_1, \dots, \varepsilon_l$, respectively. For $s = 1, \dots, i-1, i+1, \dots, m$, we assign the value b_s to all variables from X_s . Let the outcome of the decision tree Γ_j be a normal C -fault $\alpha = (\alpha_1, \dots, \alpha_l)$ for P_j . Then $\alpha_1, \dots, \alpha_l$ are the values of ρ for the variables corresponding to the edges $\varepsilon_1, \dots, \varepsilon_l$, respectively.

The outcome of Γ is the tuple ρ which will be completely defined after the finishing of the work with γ_m . One can show that $f_{S, \delta} = f_{S, \rho}$. Therefore Γ solves the problem of diagnosis of C -faults for S . It is clear that $h(\Gamma) \leq h(\Gamma_0) + \dots + h(\Gamma_k) \leq t(B)(L(P_0) - 1) + \dots + t(B)(L(P_k) - 1) = t(B)(L(S) - 1)$. \square

Remark 1. The bound from Theorem 2 is in some sense sharp: the coefficient $t_C(B)$ is the minimum number a such that $h_C(S) \leq a(L(S) - 1)$ for any $S \in \text{Net}(B)$ (the equality $h_C(S) = t_C(B)(L(S) - 1)$ holds for at least one network S from B).

Remark 2. It is not necessary to design and memorize the decision tree Γ for diagnosis of C -faults for a network $S \in \text{Net}(B)$ whose depth satisfies the inequality from Theorem 2. The proof of Theorem 2 shows that we can simulate the work of this tree efficiently if we know decision trees with minimum depth for the diagnosis of C -faults for networks from B and if we know how the network S was built from the networks contained in B .

Theorem 2 gives us an upper bound on $h_C(S)$ which depends linearly on $L(S)$ and is true for any $S \in \text{Net}(B)$. We have no lower bound on $h_C(S)$ which is linear depending on $L(S)$ and holds for any $S \in \text{Net}(B)$. However, we can obtain such a bound in the worst case. To this end, we consider a function H_B^C which characterizes the dependence in the worst case of the minimum depth of a decision tree for diagnosis of C -faults for a network from $\text{Net}(B)$ on the number of edges in the network. For each natural n ,

$$H_B^C(n) = \max\{h_C(S) : S \in \text{Net}(B), L(S) \leq n\}.$$

Theorem 3. [4] *Let B be a finite nonempty set of indecomposable nontrivial networks. Then*

$$H_B^{\{0,1\}}(n) = \Theta(n).$$

We denote by S_1^2 a network with two poles and two parallel edges connecting the poles (see Table 2).

Theorem 4. *Let B be a finite nonempty set of indecomposable nontrivial networks. If $B = \{S_1^2\}$ then $H_B^{\{1\}}(n) = 1$ for $n \geq 2$. Otherwise, $H_B^{\{1\}}(n) = \Theta(n)$.*

PROOF. Let $B = \{S_1^2\}$. An arbitrary network S from the set $\text{Net}(B)$ consists of two poles and $L(S) \geq 2$ edges connecting the poles. Therefore $F_{\{1\}}(S) = \{x_1 \vee \dots \vee x_{L(S)}, 1\}$ and $h_{\{1\}}(S) = 1$. Hence $H_B^{\{1\}}(n) = 1$ for $n \geq 2$.

Let $B \neq \{S_1^2\}$. From Theorem 2, it follows that $H_B^{\{1\}}(n) = O(n)$. Let us show that $H_B^{\{1\}}(n) = \Omega(n)$. Since $B \neq \{S_1^2\}$, the set B contains a network S_2 with a simple path between poles whose length is at least 2 (it is easy to show that S_1^2 is the only indecomposable nontrivial network in which the length of each simple path between poles is equal to 1). Let $L(S_2) = t$. One can show that, for any natural $m \geq 2$, there exists a network S_m from $\text{Net}(B)$ such that $L(S_m) \leq (m-1)t$ and there is a simple path τ between poles of

S_m with length $p \geq m$. Let x_{j_1}, \dots, x_{j_p} be variables corresponded to edges in τ . Let $I \subseteq \{j_1, \dots, j_p\}$. We consider a $\{1\}$ -fault $\delta(I)$ of S_m whose value for each variable x_i of S_m is equal to 2 if $i \in \{j_1, \dots, j_p\} \setminus I$ or $i \notin \{j_1, \dots, j_p\}$ and is equal to 1 if $i \in I$.

Let $I_1, I_2 \subseteq \{j_1, \dots, j_p\}$ and $I_1 \neq I_2$. Then there is $k \in I_1 \cup I_2$ such that $k \notin I_1 \cap I_2$. Let us consider a tuple α of values of variables of S_m in which $x_k = 0$, $x_i = 1$ for any $i \in \{j_1, \dots, j_p\} \setminus \{k\}$ and $x_i = 0$ for any $i \notin \{j_1, \dots, j_p\}$. It is easy to show that $f_{S_m, \delta(I_1)}(\alpha) \neq f_{S_m, \delta(I_2)}(\alpha)$. Therefore $|F_{\{1\}}(S_m)| \geq 2^p \geq 2^m$ and, by Lemma 1, $h_{\{1\}}(S_m) \geq m$.

Let n be a natural number such that $n \geq t$ and $m = \lfloor \frac{n}{t} \rfloor + 1$. It is clear that $L(S_m) \leq (m-1)t \leq n$. Since $h_{\{1\}}(S_m) \geq m \geq \frac{n}{t}$, we have $H_B^{\{1\}}(n) \geq \frac{n}{t}$ for any natural $n \geq t$. Therefore $H_B^{\{1\}}(n) = \Omega(n)$. \square

We denote by S_2^2 a network consisting of simple path of length two connecting two poles (see Table 2).

Theorem 5. *Let B be a finite nonempty set of indecomposable nontrivial networks. If $B = \{S_2^2\}$ then $H_B^{\{0\}}(n) = 1$ for $n \geq 2$. Otherwise, $H_B^{\{0\}}(n) = \Theta(n)$.*

PROOF. Let $B = \{S_2^2\}$. An arbitrary network S from the set $\text{Net}(B)$ consists of simple path of length $L(S) \geq 2$ connecting two poles. Therefore $F_{\{0\}}(S) = \{x_1 \wedge \dots \wedge x_{L(S)}, 0\}$ and $h_{\{0\}}(S) = 1$. Hence $H_B^{\{0\}}(n) = 1$ for $n \geq 2$.

Let $B \neq \{S_2^2\}$. From Theorem 2 it follows that $H_B^{\{0\}}(n) = O(n)$. Let us show that $H_B^{\{0\}}(n) = \Omega(n)$. Since $B \neq \{S_2^2\}$, the set B contains a network S_2 with at least 2 edges connected to a pole (it is easy to show that S_2^2 is the only indecomposable nontrivial network in which exactly one edge is connected to each pole). Let $L(S_2) = t$. One can show that, for any natural $m \geq 2$, there exists a network S_m from $\text{Net}(B)$ such that $L(S_m) \leq (m-1)t$ and there are $p \geq m$ edges connected to a pole. Let x_{j_1}, \dots, x_{j_p} be variables corresponded to these edges. Let $I \subseteq \{j_1, \dots, j_p\}$. We consider a $\{0\}$ -fault $\delta(I)$ of S_m whose value for each variable x_i of S_m is equal to 2 if $i \in \{j_1, \dots, j_p\} \setminus I$ or $i \notin \{j_1, \dots, j_p\}$ and is equal to 0 if $i \in I$.

Let $I_1, I_2 \subseteq \{j_1, \dots, j_p\}$ and $I_1 \neq I_2$. Then there is $k \in I_1 \cup I_2$ such that $k \notin I_1 \cap I_2$. Let us consider a tuple α of values of variables of S_m in which $x_k = 1$, $x_i = 0$ for any $i \in \{j_1, \dots, j_p\} \setminus \{k\}$ and $x_i = 1$ for any $i \notin \{j_1, \dots, j_p\}$. It is easy to show that $f_{S_m, \delta(I_1)}(\alpha) \neq f_{S_m, \delta(I_2)}(\alpha)$. Therefore $|F_{\{0\}}(S_m)| \geq 2^p \geq 2^m$ and, by Lemma 1, $h_{\{0\}}(S_m) \geq m$.

Let n be a natural number such that $n \geq t$ and $m = \lfloor \frac{n}{t} \rfloor + 1$. It is clear that $L(S_m) \leq (m-1)t \leq n$. Since $h_{\{0\}}(S_m) \geq m \geq \frac{n}{t}$, we have $H_B^{\{0\}}(n) \geq \frac{n}{t}$ for any natural $n \geq t$. Therefore $H_B^{\{0\}}(n) = \Omega(n)$. \square

4. Diagnosis of Indecomposable Networks

Let $C \in \{\{0, 1\}, \{0\}, \{1\}\}$. In this section, we find value of $h_C(S)$ for each nontrivial indecomposable network S with at most 10 edges. We describe now tools which were used to obtain these values. The value $t_C(S) = \frac{h_C(S)}{L(S)-1}$ can be found in an easy way if we know the value $h_C(S)$.

The first tool is a dynamic programming algorithm [1, 6, 16]. Let S contain n edges. For $\alpha_1, \dots, \alpha_m \in \{0, 1\}^n$ and $b_1, \dots, b_m \in \{0, 1\}$, we denote by $F_C(S)(\alpha_1, b_1) \cdots (\alpha_m, b_m)$ the set of all functions f from $F_C(S)$ for which $f(\alpha_1) = b_1, \dots, f(\alpha_m) = b_m$. All nonempty subsets of $F_C(S)$ which can be represented in the form $F_C(S)(\alpha_1, b_1) \cdots (\alpha_m, b_m)$ including the set $F_C(S)$ are called *separable* subsets of the set $F_C(S)$. Let $G \subseteq F_C(S)$. Then

$$G(\alpha_1, b_1) \cdots (\alpha_m, b_m) = F_C(S)(\alpha_1, b_1) \cdots (\alpha_m, b_m) \cap G.$$

First, the dynamic programming algorithm constructs the set $F_C(S)$. Next, this algorithm constructs the set $\Delta_C(S)$ of all separable subsets of the set $F_C(S)$. Each subset $G \in \Delta_C(S)$ is considered as a subproblem of the initial problem $F_C(S)$: for a given function $f \in G$ we should find a normal C -fault ρ for S such that $f = f_{S,\rho}$. We denote by $h(G)$ the minimum depth of a decision tree solving the problem G . For a problem G , we denote by $E(G)$ the set of tuples $\alpha \in \{0, 1\}^n$ such that $|G(\alpha, 0)| \geq 1$ and $|G(\alpha, 1)| \geq 1$. The considered algorithm is based on the following equalities: $h(G) = 0$ if $|G| = 1$, and

$$h(G) = \min\{1 + \max\{h(G(\alpha, 0)), h(G(\alpha, 1))\} : \alpha \in E(G)\}$$

if $|G| \geq 2$. The algorithm begins the computation of the value $h(G)$ from smallest separable subsets and finishes when the value $h(F_C(S))$ is computed.

For networks containing at most five edges, we used the dynamic programming algorithm which allows us to find exact value of $h_{\{0,1\}}(S)$ (these results were described in [4]). For networks containing at most seven edges, we used the dynamic programming algorithm which allows us to find exact values of $h_{\{0\}}(S)$ and $h_{\{1\}}(S)$.

For the networks containing more than five edges for $C = \{0, 1\}$ and networks containing more than seven edges for $C \in \{\{0\}, \{1\}\}$, we used lower bound on $h_C(S)$ from Lemma 1. To obtain upper bound on $h_C(S)$, we used greedy algorithms for decision tree construction considered in [2]. We describe here only two greedy algorithms which allowed us to obtain best results for each of the considered networks. The description is adapted to the problem of diagnosis.

We now consider the first step of each algorithm – the choice of n -tuple from $E(F_C(S))$ which will be attached to the root of the tree, where $n = L(S)$.

For a given n -tuple α from $E(F_C(S))$ we denote $N_0(\alpha) = |F_C(S)(\alpha, 0)|$ and $N_1(\alpha) = |F_C(S)(\alpha, 1)|$. The first greedy algorithm (similar to the one considered in [15]) chooses a tuple α from $E(F_C(S))$ which minimizes the value

$$\max\{N_0(\alpha), N_1(\alpha)\}.$$

The second greedy algorithm (similar to the one considered in [17]) chooses a tuple α from $E(F_C(S))$ which minimizes the value

$$N_0(\alpha) \log_2 N_0(\alpha) + N_1(\alpha) \log_2 N_1(\alpha).$$

Later, each greedy algorithm works in the same way with separable subsets $F_C(S)(\alpha, 0)$ and $F_C(S)(\alpha, 1)$ corresponding to the children of the root, etc. The construction of the decision tree is finished when separable subsets corresponding to terminal nodes are singletons.

For each of the considered nontrivial indecomposable network S with at most 10 edges and for each $C \in \{\{0, 1\}, \{0\}, \{1\}\}$, the difference between lower bound on $h_C(S)$ from Lemma 1 and upper bound – the depth of a decision tree constructed by one of the considered greedy algorithms – is at most one. For $C = \{0, 1\}$, the obtained results were described in [4].

To find the exact value of $h_C(S)$ for each nontrivial indecomposable network S with at most 10 edges and for each $C \in \{\{0, 1\}, \{0\}, \{1\}\}$, we created new algorithm called Bounded Depth Algorithm.

Let $C \in \{\{0, 1\}, \{0\}, \{1\}\}$, d be a nonnegative integer, and $S \in \text{Net}(B)$. Let G be a subset of $F_C(S)$ with $|G| \geq 2$. We will describe a sequence $T(G, d)$ of tuples from $\{0, 1\}^n$ where $n = L(S)$.

We will say that two tuples $\alpha, \beta \in E(G)$ are equivalent if, for any $f \in G$, $f(\alpha) = f(\beta)$. This equivalence relation divides $E(G)$ into classes of equivalence E_1, \dots, E_p . We choose one representative from each class

and order these representatives $\alpha_1, \dots, \alpha_p$ such that, for $i = 1, \dots, p-1$, $I(\alpha_i) \leq I(\alpha_{i+1})$ where, for $j = 1, \dots, p$, $I(\alpha_j) = \max\{|G(\alpha_j, 0)|, |G(\alpha_j, 1)|\}$. If $\lceil \log_2 I(\alpha_1) \rceil > d-1$, the sequence $T(G, d)$ is empty. Let $\lceil \log_2 I(\alpha_1) \rceil \leq d-1$ and k be the maximum number from $\{1, \dots, p\}$ such that $\lceil \log_2 I(\alpha_k) \rceil \leq d-1$. Then $T(G, d) = (\alpha_1, \dots, \alpha_k)$.

Bounded Depth Algorithm. Let t be a nonnegative integer. We now describe an algorithm which recognizes if there exists a tree Γ for diagnosis of C -faults for the network S with $h(\Gamma) \leq t$, and if such a tree exists, it constructs one. The inputs of this algorithm are the set $F_C(S)$ and the number t . During each step, the algorithm will work with a tree D which is a prefix of some decision tree for diagnosis of C -faults for S . Each node v of the tree D will be labeled with an integer $d(v)$ and a sequence $T(v)$ of tuples from $\{0, 1\}^n$ where $n = L(S)$. We will correspond also a subset $G(v)$ of the set $F_C(S)$ to the node v .

We will assume that in each nonterminal node of D , the edge starting in this node and labeled with 0 is going to the left and the edge labeled with 1 is going to the right. In this case, we have a natural ordering of terminal nodes of D from the left to the right.

At the first step, we form the root node r of D for which $d(r) = t$, $T(r) = T(F_C(S), t)$, and $G(r) = F_C(S)$. Let us assume that we already made $m \geq 1$ steps and constructed a tree D . We now describe the step number $m+1$. If each terminal node of D is labeled with a normal C -fault for S , then D is a decision tree for diagnosis of C -faults for the network S with $h(D) \leq t$. Otherwise, we choose the leftmost terminal node v of D which is not labeled with a normal fault.

Let $T(v)$ be empty. If v be the root of D , then there is no decision tree Γ for diagnosis of C -faults for S such that $h(\Gamma) \leq t$, and the algorithm finishes its work. If v is not a root of D , then we remove v and the edge entering v . Also, we remove the subtree with root at the sibling v' of v and remove the edge entering v' . Moreover, we remove the n -tuple α attached to the parent of v . Then, we proceed to step number $m+2$.

Let $T(v)$ be nonempty and α be the first n -tuple in $T(v)$. We remove α from $T(v)$ and we attach it to the node v . We add to D two nodes v_0, v_1 and we draw edges from v to v_0 and v_1 labeled with 0 and 1, respectively. For v_0 , $d(v_0) = d(v) - 1$, $G(v_0) = G(v)(\alpha, 0)$ and $T(v_0) = T(G(v_0), d(v_0))$. For v_1 , $d(v_1) = d(v) - 1$, $G(v_1) = G(v)(\alpha, 1)$ and $T(v_1) = T(G(v_1), d(v_1))$. If $|G(v_0)| = 1$, we attach to the node v_0 the corresponding normal fault for the

function from $G(v_0)$. Similarly, if $|G(v_1)| = 1$, we attach to the node v_1 the corresponding normal fault for the function from $G(v_1)$. Now we proceed to step number $m + 2$.

If, for some $C \in \{\{0, 1\}, \{0\}, \{1\}\}$ and some network S with at most 10 edges, the difference between the lower and the upper bounds for $h_C(C)$ is equal to 1, we apply the Bounded Depth Algorithm to the set $F_C(S)$ and to the number t which is equal to the obtained lower bound for $h_C(S)$. If the algorithm constructs a decision tree Γ for diagnosis of C -faults for the network S with $h(\Gamma) = t$, then $h_C(S) = t$. Otherwise, $h_C(S) = t + 1$.

We consider now an example of the Bounded Depth Algorithm work.

Example 1. Let S_1^2 be a network with two edges as in Figure 1. This network implements the function $f_{S_1^2}(x_1, x_2) = x_1 \vee x_2$. We are interested in finding a decision tree with minimum depth that solves the *problem of diagnosis of $\{0, 1\}$ -faults for S_1^2* .

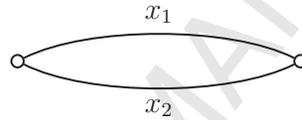


Figure 1: Network S_1^2

We found that $F_{\{0,1\}}(S_1^2) = \{f_{S_1^2, \rho_1}, f_{S_1^2, \rho_2}, f_{S_1^2, \rho_3}, f_{S_1^2, \rho_4}, f_{S_1^2, \rho_5}\}$ where $\rho_1 = (0, 0)$, $\rho_2 = (1, 1)$, $\rho_3 = (0, 2)$, $\rho_4 = (2, 0)$, and $\rho_5 = (2, 2)$ are normal faults for the network S_1^2 . Table 1 describes functions from the set $F_{\{0,1\}}(S_1^2)$.

Input	$f_{S_1^2, \rho_1}$	$f_{S_1^2, \rho_2}$	$f_{S_1^2, \rho_3}$	$f_{S_1^2, \rho_4}$	$f_{S_1^2, \rho_5}$
00	0	1	0	0	0
01	0	1	1	0	1
10	0	1	0	1	1
11	0	1	1	1	1

Table 1: Outputs of functions $f_{S_1^2, \rho_i}$, $i = 1, 2, 3, 4, 5$

We know that $|F_{\{0,1\}}(S_1^2)| = 5$. According to Lemma 1, the minimum depth of a decision tree that solves the *problem of diagnosis of $\{0, 1\}$ -faults*

for S_1^2 is at least 3. In this example, we use Bounded Depth Algorithm to verify if a tree with depth of 3 exists.

Note that from each class of equivalent 2-tuples, we only choose one representative. Also, tuples that have the same output for all faults are excluded. As a result, at the beginning of the algorithm work we have the whole set $\{0, 1\}^2$ of input tuples (see Table 1).

Let $t = 3$. One can show that

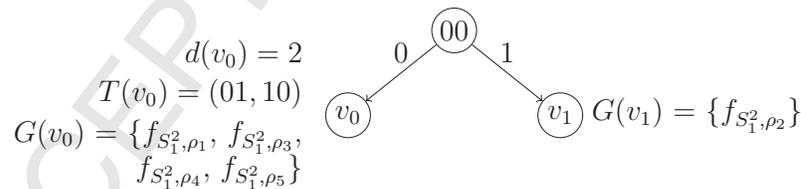
$$T(F_{\{0,1\}}(S_1^2), t) = (00, 01, 10, 11).$$

We start constructing a tree D with a root node r as following:

$$\begin{aligned} d(r) &= 3 \\ \textcircled{r} \quad T(r) &= (00, 01, 10, 11) \\ G(r) &= \{f_{S_1^2, \rho_1}, f_{S_1^2, \rho_2}, \\ &\quad f_{S_1^2, \rho_3}, f_{S_1^2, \rho_4}, f_{S_1^2, \rho_5}\} \end{aligned}$$

We start with the root node r (the leftmost terminal node not labeled with a normal fault). We remove the first tuple 00 from $T(r)$ and we attach it to the node r . We add two nodes v_0, v_1 and we draw edges from r to v_0 and v_1 labeled with 0 and 1, respectively.

For v_0 , $d(v_0) = d(r) - 1$, $G(v_0)$ contains all faulty functions that give an output of 0 with the input tuple 00 and $T(v_0) = T(G(v_0), d(v_0))$. For v_1 , $d(v_1) = d(r) - 1$, $G(v_1)$ contains all faulty functions that give an output of 1 with the input tuple 00 and $T(v_1) = T(G(v_1), d(v_1))$. The drawing is as following:



Since $|G(v_1)| = 1$, we attach to the node v_1 the fault ρ_2 corresponding to the function in $G(v_1)$. For v_0 , we remove the first tuple 01 from $T(v_0)$ and we attach it to the node v_0 . We add two nodes v_{00} and v_{01} and we draw edges from v_0 to v_{00} and v_{01} labeled with 0 and 1, respectively.

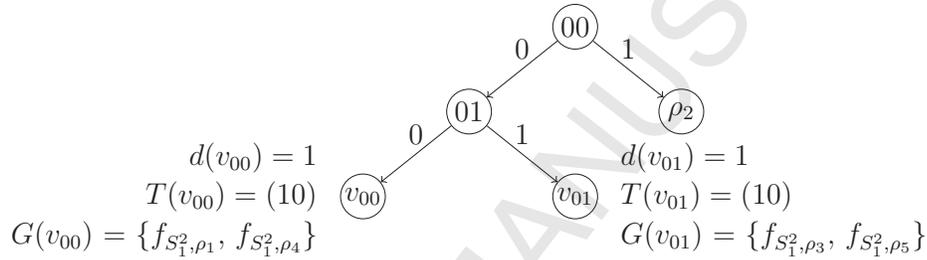
For v_{00} , $d(v_{00}) = d(v_0) - 1$, $G(v_{00})$ contains all faulty functions within $G(v_0)$ that give an output of 0 with the input tuple 01 and

$$T(v_{00}) = T(G(v_{00}), d(v_{00})).$$

For v_{01} , $d(v_{01}) = d(v_0) - 1$, $G(v_{01})$ contains all faulty functions within $G(v_0)$ that give an output of 1 with the input tuple 01 and

$$T(v_{01}) = T(G(v_{01}), d(v_{01})).$$

The drawing is as following:



By repeating the same sequence of steps for nodes v_{00} and v_{01} , the following tree will be constructed with depth of tree not exceeding the set threshold $t = 3$. Since the depth of this tree equals to the value obtained by Lemma 1, one can say that this tree solves the *problem of diagnosis of $\{0, 1\}$ -faults for S_1^2* and has minimum depth.

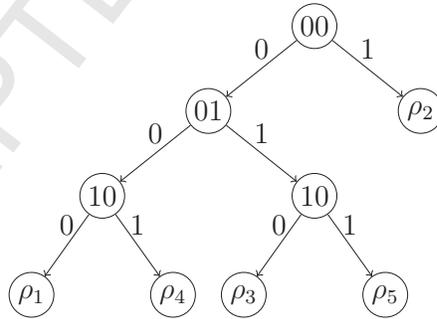


Figure 2: A decision tree with minimum depth for diagnosing $\{0, 1\}$ -faults for S_1^2 constructed using Bounded Depth Algorithm

The obtained results are shown in Tables 2-10. These tables contain, for each nontrivial indecomposable network S with at most 10 edges (the list of these networks was published by Kuznetsov in [11]),

- the name of the network S in the form S_j^i where i is the number of edges in the network and j is the index of the network in the list of networks with i edges;
- graphical representation of the network S in which two empty circles are the poles and the filled circles are usual nodes of the network;
- the values $h_{\{0,1\}}(S)$, $h_{\{0\}}(S)$, and $h_{\{1\}}(S)$.

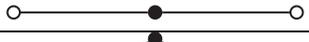
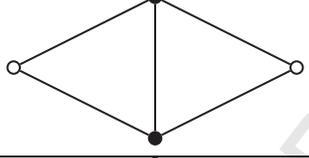
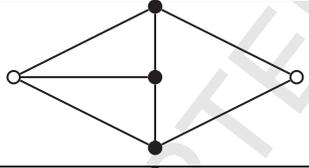
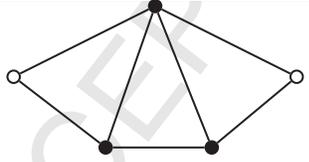
Network S		$h_{\{0,1\}}(S)$	$h_{\{0\}}(S)$	$h_{\{1\}}(S)$
S_1^2		3	2	1
S_2^2		3	1	2
S_1^5		7	4	4
S_1^7		9	5	5
S_2^7		9	5	5

Table 2: Results for networks with 2, 5, and 7 edges

Network S		$h_{\{0,1\}}(S)$	$h_{\{0\}}(S)$	$h_{\{1\}}(S)$
S_1^8		10	7	6
S_2^8		10	7	5
S_3^8		10	5	7
S_4^8		10	6	7
S_5^8		10	6	7
S_1^9		11	7	7
S_2^9		11	6	7

Table 3: Results for networks with 8 and 9 edges

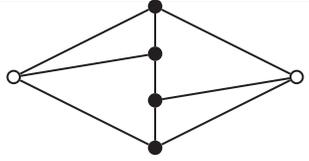
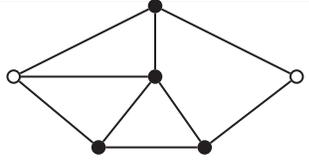
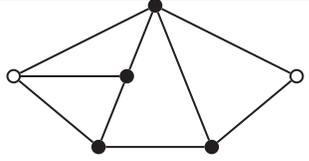
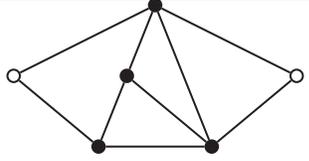
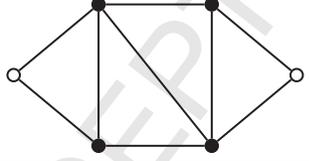
	Network S	$h_{\{0,1\}}(S)$	$h_{\{0\}}(S)$	$h_{\{1\}}(S)$
S_3^9		11	7	7
S_4^9		11	7	7
S_5^9		11	7	7
S_6^9		11	7	7
S_7^9		11	7	6
S_8^9		11	7	7
S_9^9		12	8	6

Table 4: Results for networks with 9 edges

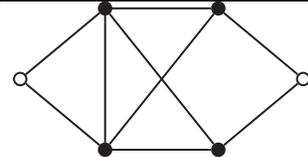
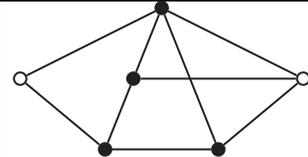
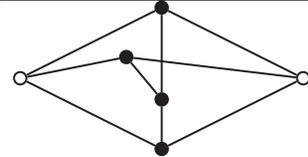
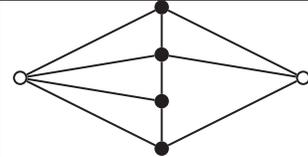
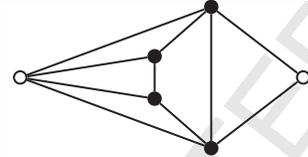
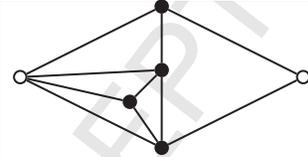
	Network S	$h_{\{0,1\}}(S)$	$h_{\{0\}}(S)$	$h_{\{1\}}(S)$
S_{10}^9		12	7	7
S_{11}^9		12	7	7
S_{12}^9		12	7	7
S_1^{10}		13	8	7
S_2^{10}		13	8	7
S_3^{10}		13	8	7
S_4^{10}		13	8	7

Table 5: Results for networks with 9 and 10 edges

	Network S	$h_{\{0,1\}}(S)$	$h_{\{0\}}(S)$	$h_{\{1\}}(S)$
S_5^{10}		13	8	7
S_6^{10}		13	8	7
S_7^{10}		13	8	7
S_8^{10}		13	8	7
S_9^{10}		13	8	7
S_{10}^{10}		13	8	7
S_{11}^{10}		13	8	7

Table 6: Results for networks with 10 edges

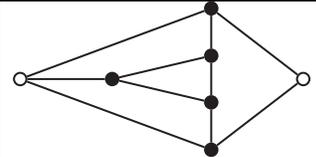
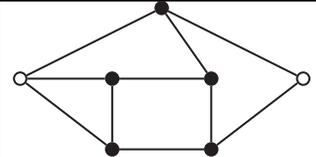
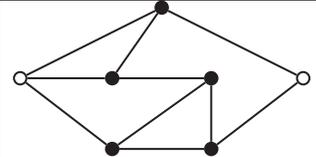
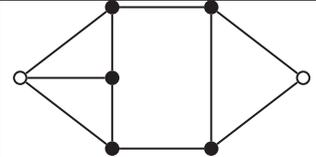
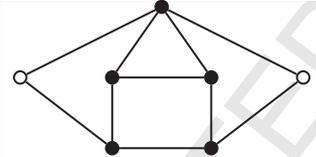
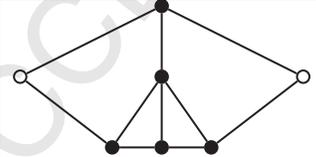
	Network S	$h_{\{0,1\}}(S)$	$h_{\{0\}}(S)$	$h_{\{1\}}(S)$
S_{12}^{10}		13	7	8
S_{13}^{10}		13	7	8
S_{14}^{10}		13	7	8
S_{15}^{10}		13	7	8
S_{16}^{10}		13	7	8
S_{17}^{10}		13	7	8
S_{18}^{10}		13	7	8

Table 7: Results for networks with 10 edges

	Network S	$h_{\{0,1\}}(S)$	$h_{\{0\}}(S)$	$h_{\{1\}}(S)$
S_{19}^{10}		13	7	8
S_{20}^{10}		13	7	8
S_{21}^{10}		13	7	8
S_{22}^{10}		13	7	8
S_{23}^{10}		13	8	7
S_{24}^{10}		13	8	7
S_{25}^{10}		13	8	7

Table 8: Results for networks with 10 edges

	Network S	$h_{\{0,1\}}(S)$	$h_{\{0\}}(S)$	$h_{\{1\}}(S)$
S_{26}^{10}		13	8	7
S_{27}^{10}		13	8	7
S_{28}^{10}		13	8	7
S_{29}^{10}		13	8	7
S_{30}^{10}		13	7	8
S_{31}^{10}		13	7	8
S_{32}^{10}		13	7	8

Table 9: Results for networks with 10 edges

	Network S	$h_{\{0,1\}}(S)$	$h_{\{0\}}(S)$	$h_{\{1\}}(S)$
S_{33}^{10}		13	7	8
S_{34}^{10}		13	7	8

Table 10: Results for networks with 10 edges

Using these results we can obtain a number of corollaries. For $i = 2, 5, 7, 8, 9, 10$, we denote by $B(i)$ the set of all indecomposable networks with i edges. For $C \in \{\{0, 1\}, \{0\}, \{1\}\}$ and $i = 2, 5, 7, 8, 9, 10$, the value $t_C(B(i))$ can be found in Table 11 at the intersection of the row C and the column i .

$C \backslash i$	2	5	7	8	9	10
$\{0, 1\}$	3	1.75	1.5	≈ 1.43	1.5	≈ 1.44
$\{0\}$	2	1	≈ 0.83	1	1	≈ 0.89
$\{1\}$	2	1	≈ 0.83	1	≈ 0.88	≈ 0.89

Table 11: Values of $t_C(B(i))$ for $C \in \{\{0, 1\}, \{0\}, \{1\}\}$ and $i = 2, 5, 7, 8, 9, 10$

5. Conclusions

We studied the depth of decision trees for diagnosis of three types of constant faults in undirected read-once contact networks over finite bases. In the future, we are planning to study the diagnosis of constant faults in directed read-once contact networks.

Acknowledgements

Research reported in this publication was supported by the King Abdullah University of Science and Technology (KAUST).

The authors are greatly indebted to reviewers for useful comments.

- [1] A. Alkhalid, I. Chikalov, S. Hussain, M. Moshkov, Extensions of dynamic programming as a new tool for decision tree optimization, in: S. Ramanna, R.J. Howlett, L.C. Jain (Eds.), *Emerging Paradigms in Machine Learning, Smart Innovation, Systems and Technologies*, vol. 13, Springer, Heidelberg, 2013, pp. 11–29.
- [2] A. Alkhalid, I. Chikalov, M. Moshkov, Comparison of greedy algorithms for decision tree optimization, in: A. Skowron, Z. Suraj (Eds.), *Rough Sets and Intelligent Systems – Professor Zdzislaw Pawlak in Memoriam*, Intelligent Systems Reference Library, vol. 43, Springer, Heidelberg, 2013, pp. 21–40.
- [3] D. Angluin, L. Hellerstein, M. Karpinski, Learning read-once formulas with queries, *Journal of the ACM*, 40 (1993) 185–210.
- [4] M. Busbait, I. Chikalov, S. Hussain, M. Moshkov, Diagnosis of constant faults in read-once contact networks over finite bases, *Discrete Applied Mathematics*, 184 (2015) 76–82.
- [5] I.A. Chegis, S.V. Yablonskii, Logical methods for testing the operation of electrical circuits, *Trudy Mat. Inst. Steklov.*, 51 (1958) 270–360 (in Russian).
- [6] M.R. Garey, Optimal binary identification procedures, *SIAM Journal on Applied Mathematics*, 23 (1972) 173–186.
- [7] G.B. Goduhina, Experimental study of conditional tests for diagnosis of constant faults in iteration-free contact networks, in: A.I. Markov (Ed.), *Combinatorial-Algebraic and Probabilistic Methods in Applied Mathematics*, Gorky State University, Gorky, 1988, pp. 16–21 (in Russian).
- [8] R.S. Goldman, V.P. Chipulis, Diagnosis of iteration-free combinatorial circuits, *Diskret. Analiz* 14 (1969) 3–15 (in Russian).

- [9] M.F. Karavai, Diagnosis of treelike circuits of arbitrary basis, *Avtomat. i Telemekhan.*, 34 (1973) 173-181 (in Russian).
- [10] N.A. Karpova, Contact scheme, *Encyclopedia of Mathematics*, http://www.encyclopediaofmath.org/index.php/Contact_scheme
- [11] A.V. Kuznetsov, Non-repeating contact schemes and non-repeating superpositions of functions of algebra of logic, *Trudy Mat. Inst. Steklov.*, 51 (1958) 186-225 (in Russian).
- [12] Kh.A. Madatyan, On complete tests for contact circuits without repetitions, *Problemy Kibernet.*, 23 (1970) 103-118 (in Russian).
- [13] M. Moshkov, Diagnosis of 0-1 faults in iteration-free contact networks, in: *Proc. 6th Scientific Conference of Young Scientists of Volga-Vyatka Region, Gorky, 1986*, pp. 103-104 (in Russian).
- [14] M. Moshkov, Diagnosis of constant faults in circuits, in: O.B. Lupanov (Ed.), *Mathematical Problems of Cybernetics*, vol. 9, Nauka Publishers, Moscow, 2000, pp. 79-100 (in Russian).
- [15] M. Moshkov, Time complexity of decision trees, in: J.F. Peters, A. Skowron (Eds.), *Transactions on Rough Sets III, Lecture Notes in Computer Science*, vol. 3400, Springer, Heidelberg, 2005, pp. 244-459.
- [16] M. Moshkov, I. Chikalov, On algorithm for constructing of decision trees with minimal depth, *Fundamenta Informaticae*, 41 (2000) 295-299.
- [17] J.R. Quinlan, Induction of decision trees, *Machine Learning*, 1 (1986) 81-106.
- [18] V. Raghavan, S. Schach, Learning switch configurations, in: *Proc. 3rd Annual Workshop on Computational Learning Theory*, Morgan Kaufmann Publishers, 1990, pp. 38-51.
- [19] C.E. Shannon, A symbolic analysis of relay and switching circuits, *Trans. AIEE*, 57 (1938) 713-723.