OPEN

International Conference KNOWLEDGE-BASED ORGANIZATION DE GRUYTER Vol. XXIV No 3 2018

## ALGORITHM TO AUTOMATICALLY DETERMINE THE LENGTH OF A **BIVALENT SYSTEM**

## Paul VASILIU

## "Mircea cel Bătrân"Naval Academy, Constanța, Romania p\_vasiliu@yahoo.com

**Abstract:** From the point of view of reliability theory, a system can have two stable states: functioning and defect (bivalent system). Any system is a set of elements. Each element in this set can be found in one of the following states: operating state and fault condition. A subset of elements in the running state is called a system link if they only ensure the system works. The length of a bivalent system is equal to the minimum number of elements that the system holds. In this paper we present an algorithm for the automatic determination of the length of a bivalent system, a Matlab script, a case study and subsequent development directions.

## Keywords: algorithm, link, length, reliability

## **1. Introduction**

The plurality of all elements of the assembly of elements that reflect how the state of an S system depends on the states of its components is called a structure.

In the reliability analysis, structures play an important role. In a first step, the structure of the equipment is analyzed, followed by establishing the algebraic expression of the structure function and then constructing a reliability network associated with that equipment.

Considerations regarding complex systems or equipment are based on the following assumptions:

The equipment considered can only be in one of the following states: operating state or fault condition (bivalent system). The considered equipment can be decomposed into ncomponents (elements) е,  $i = 1, 2, \dots, n$ . The set of all these elements is:  $E = \{e_1, e_2, \cdots, e_n\}$ .

Each component  $e_i$ ,  $i = 1, 2, \dots, n$ , is associated with a state variable  $\mathbf{x}_i$  defined

by:  $x_i = \begin{cases} 1 & \text{if } e_i \text{ is in operation} \\ 0 & \text{if } e_i \text{ does not work} \end{cases}$ 

The set of states  $X = \{x_1, x_2, \cdots, x_m\}$ characterizes the set of possible states of the elements. assembly of Obvious.  $card(X) = 2^n$ .

The system S is associated with a state variable **y** defined as follows:

 $y = \begin{cases} 1 & \text{if } S \text{ is in operation} \\ 2 & \text{otherwise} \end{cases}$ 

 $y = \{0 \text{ if } S \text{ does not work} \\ \text{Let } Y = \{y\} = \{0,1\}. \text{ The } y \text{ state variable} \}$ depends on the set of X states.

We can define a function  $\varphi : X \to Y$ , defined by:  $y = \varphi(x_1, x_2, \dots, x_n)$ . The function  $\varphi$  is called the function of the structure. Since the function  $\varphi$  depends on **n** independent variables, it is called function of the n order structure. The *S* system can be identified with the pair  $(E, \varphi)$ :  $S = (E, \varphi)$ . system  $S = (E, \varphi).$ Let the The subassembly

DOI: 10.1515/kbo-2018-0141

<sup>© 2015.</sup> This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 3.0 License.

$$\begin{split} L &= \left\{ e_i \mid i \in I \subset \{1, 2, \cdots, n\} \right\} \subset E \text{ is called} \\ \text{a link if for any } x_i = 1, \ i \in I \text{ and for any} \\ x_i &= 0, \qquad i \notin I \qquad \text{follows:} \\ y &= \varphi(x_1, x_2, \cdots, x_n) = 1. \end{split}$$

In other words, a connection of an operating system (y = 1) is a subset (subassembly) of elements in which all elements are in the running state, the rest of the elements being defective.

By reformulating, a link is a subset of elements that are in a state of operation and which ensure the system's operation if the other elements are defective.

Any system has at least one link.

To determine all system links, calculate the values of the structure function at all points  $(x_1, x_2, \dots, x_n)$ . If  $f(x_1, x_2, \dots, x_n) = 1$  for  $x_{i_1} = x_{i_2} = \dots = x_{i_p} = 1$  and  $x_i = 0$  for any  $i \in \{1, 2, \dots, n\} \setminus \{i_1, i_2, \dots, i_p\}$  then the subset of elements (subassembly of elements)  $\{e_{i_1}, e_{i_2}, \dots, e_{i_p}\}$  is a link.

Let  $l_k > 0$  the number of links with k elements,  $k \in N^*$ .

It is called the length of the system S number  $\lambda(S) = \min\{k \mid l_k > 0\}$ .

The study of the links and length of a system brings important information about the operation of the system under the condition that some of the component elements are defective.

The difficulty of manually determining links and the length of a system increases as the number of elements in the system increases. There is a need to simplify and automate the determination of system links and system length. For this reason, the authors wrote the Matlab functions presented in Section 2.

# 2. Matlab implementation

The Matlab main function with the signature *function* links(n) receives the **n**-value of the input. The function generates all links to the reliability network. The function calculates the length of the system. The function *function* w=conversion(n,v) converts the value of the input argument n

from base 10 to base 2 and stores the result of the conversion in the variable w.

The function *function* val=fstruct(x) receives the input vector x of the structure function variables and returns the value of the structure function.

The display of links is made by the function *function write(n,w,tip)*.

The function function w=inverse(v) inverts the vector v and stores the result in the vector w.

The function function printnl(nb) displays the number of links with k elements,  $k = 1,2, \cdots$ 

We present below the Matlab code for implementing the algorithm.

% The main function function links(n) clc; k=2^n;

for i=1:n v(i)=0;end ival=1; for i=0:k-1 w=conversion(i,v); val=fstruct(w); if val==1 vval(ival)=i; fprintf(' The value %d :n y = f(',i); nb(ival)=0; for j=1:length(w)-1 fprintf('%d,',w(j)); if w(j) == 1nb(ival)=nb(ival)+1; end end if w(length(w))==1 nb(ival)=nb(ival)+1; end fprintf('%d',w(length(w))); fprintf(') = %d (n',val);write(i,w,'l'); ival=ival+1; end end printnl(nb);

fprintf(' The length of the system is %d \n',min(nb));

end % Conversion function % from base 10 in base 2 function w=conversion(n,v) i=1: while  $n \sim = 0$ uc=mod(n,2);v(i)=uc; i=i+1;n = floor(n/2);end w=inverse(v); end % Function for reversing vector v function w=inverse(v) n=length(v); for i=1:n w(i)=v(n-i+1);end end % Function for displaying links function write(n,w,tip) p=length(w); if tip=='l' fprintf(' Subassembly L = { '); for i=1:p if w(i) == 1fprintf('e%d',i); end end fprintf(' } is a link n); end end % Function to display the number of links % with k elements % k=1,2,... function printnl(nb) kmin=min(nb); kmax=max(nb);

for i=kmin:1:kmax s=0; for j=1:length(nb) if nb(j)==i & nb(j)~=0 s=s+1; end end fprintf(' Number of links with %d elements = %d n',i,s); end end % Function of system structure function val=fstruct(x) val=x(5)\*(1-(1-x(3)\*x(4))\*(1-x(1)\*x(2))); end

## 3. Case study

Consider the system  $S = (E, \varphi)$  with the properties: n = 5,  $E = \{e_1, e_2, e_3, e_4, e_5\}$ ,  $X = \{x_1, x_2, x_3, x_4, x_5\}$ . The graph of the system is shown in Figure 1. Obviously, the structure function of the system is the function:  $\varphi: X \to \{0,1\}$ ,  $\varphi(x_1, x_2, x_3, x_4, x_5) = x_5 \cdot (1 - (1 - x_1 \cdot x_2) \cdot (1 - x_3 \cdot x_4))$ 

For manual determination of all system links, the values of the structure function are calculated at all points  $(x_1, x_2, x_3, x_4, x_5)$ . If for  $x_{i_1} = x_{i_2} = \dots = x_{i_p} = 1$  and  $x_i = 0$ for any  $i \in \{1, 2, \dots, n\} \setminus \{i_1, i_2, \dots, i_p\}$  and  $\varphi(x_1, x_2, \dots, x_n) = 1$  then the subassembly of elements  $\{e_{i_1}, e_{i_2}, \dots, e_{i_p}\}$  is a link. Table 1 is completed and all system links

are obtained.



Figure 1: The system

							$I  dble  I  System  links  \mathfrak{I} = [\mathcal{L}]$
Value	$\boldsymbol{x_1}$	$x_2$	$x_3$	$x_4$	$x_5$	φ	Links
0	0	0	0	0	0	0	
1	0	0	0	0	1	0	
2	0	0	0	1	0	0	
3	0	0	0	1	1	0	
4	0	0	1	0	0	0	
5	0	0	1	0	1	0	
6	0	0	1	1	0	0	
7	0	0	1	1	1	1	$\{e_3, e_4, e_5\}$
8	0	1	0	0	0	0	
9	0	1	0	0	1	0	
10	0	1	0	1	0	0	
11	0	1	0	1	1	0	
12	0	1	1	0	0	0	
13	0	1	1	0	1	0	
14	0	1	1	1	0	0	
15	0	1	1	1	1	1	$\{e_2, e_3, e_4, e_5\}$
16	1	0	0	0	0	0	
17	1	0	0	0	1	0	
18	1	0	0	1	0	0	
19	1	0	0	1	1	0	
20	1	0	1	0	0	0	
21	1	0	1	0	1	0	
22	1	0	1	1	0	0	
23	1	0	1	1	1	1	$\{e_1, e_3, e_4, e_5\}$
24	1	1	0	0	0	0	
25	1	1	0	0	1	1	$\{e_1, e_2, e_5\}$
26	1	1	0	1	0	0	
27	1	1	0	1	1	1	$\{e_1, e_2, e_4, e_5\}$
28	1	1	1	0	0	0	
29	1	1	1	0	1	1	$\{e_1, e_2, e_3, e_5\}$
30	1	1	1	1	0	0	
31	1	1	1	1	1	1	$\{e_1, e_2, e_3, e_4, e_5\}$

Table 1 System links  $S = (E, \varphi)$ 

To determine the length of the system, the numbers are determined:

 $l_3 = 2$  because there are two links each with three elements  $\{e_3, e_4, e_5\}$  and  $\{e_1, e_2, e_5\}$ ); (links:  $l_4 = 4$  because there are four links each with four elements (links:  $\{e_2,e_3,e_4,e_5\},\{e_1,e_3,e_4,e_5\},\{e_1,e_2,e_4,e_5\}$ and  $\{e_1, e_2, e_3, e_5\}$ ;  $l_5 = 1$  because there is only one link to five

elements (link:  $\{e_1, e_2, e_3, e_4, e_5\}$ ).

The *S* system length is equal to:  $\lambda(S) = \min\{k \mid l_k > 0\} = \min\{3, 4, 5\} = 3$ Using the Matlab function above, we will determine all the links of this system and the length of the system. The structure function denoted by  $\varphi$ coincides with the function denoted by f in the Matlab code. An example of execution is: >> links(5) The value 7 : y = f(0, 0, 1, 1, 1) = 1Subassembly  $L = \{ e3 e4 e5 \}$  is a link The value 15 : y = f(0, 1, 1, 1, 1) = 1Subassembly  $L = \{ e2 e3 e4 e5 \}$  is a link The value 23 : y = f(1, 0, 1, 1, 1) = 1Subassembly  $L = \{ e1 e3 e4 e5 \}$  is a link The value 25 : y = f(1, 1, 0, 0, 1) = 1Subassembly  $L = \{ e1 e2 e5 \}$  is a link The value 27 : y = f(1, 1, 0, 1, 1) = 1Subassembly  $L = \{ e1 e2 e4 e5 \}$  is a link The value 29 : y = f(1, 1, 1, 1, 0, 1) = 1Subassembly  $L = \{ e1 e2 e3 e5 \}$  is a link The value 31: y = f(1, 1, 1, 1, 1) = 1Subassembly  $L = \{ e1 e2 e3 e4 e5 \}$  is a link Number of links with 3 elements = 2Number of links with 4 elements = 4Number of links with 5 elements = 1The length of the system is 3

It is easy to see that the linkages generated by the Matlab function coincide with the links determined manually in Table 1.

Also, the length of the system determined with the Matlab function coincides with the one determined manually.

## 4. Conclusions

Let  $n \in N^*$ ,  $E = \{e_1, e_2, \dots, e_n\},$   $X = \{x_1, x_2, \dots, x_n\}$ , function  $\varphi: X \to \{0, 1\},$   $y = \varphi(x_1, x_2, \dots, x_n)$ , and the system  $S = (E, \varphi).$ It's called the dual system of the

system  $S = (E, \varphi)$  the system  $S^d = (E, \varphi^d)$ 

where the structure function  $\varphi^d : X \to \{0,1\}$  is defined by

$$\varphi^{d}(x_{1}, x_{2}, \dots, x_{n}) = 1 - \varphi(1 - x_{1}, 1 - x_{2}, \dots, 1 - x_{n})$$

One of the developments in this article is the automatic determination of the links and length of the dual system of a given system. Another development is the determination

of the minimal links of the  $S = (E, \varphi)$ 

system and its dual  $S^{d} = (E, \varphi^{d})$  system.

#### Acknowledgements

The author thanks this way to all those who contributed in one way or another to the elaboration of this work.

## References

[1] Cătuneanu, V.,M., Mihalache, A., *Bazele teoretice ale fiabilității*, Ed. Academiei, 1983, București, România.

- [2] Panaite, V., Popescu, M.,O., Calitatea produselor și fiabilitate, Ed. Matrix Rom, 2003, București, România.
- [3] Țârcolea, C., Filipoiu, A., Bontaș, S., *Tehnici actuale în teoria fiabilității*, Ed. Științifică și Enciclopedică, 1989, București, România.
- [4] Vasiliu P., Deliu F., *Funcție Matlab pentru determinarea legăturilor unei rețele de fiabilitate*, Buletinul AGIR nr. 4/2017, pag. 24-27.
- [5] Vasiliu, P., Programare în Matlab, Ed. ANMB, 2015, Constanța, România.