INFORMATION TECHNOLOGY

This paper has determined the relevance of the issue related to improving the accuracy of the results of mathematical modeling of the software security testing process. The fuzzy GERT-modeling methods have been analyzed. The necessity and possibility of improving the accuracy of the results of mathematical formalization of the process of studying software vulnerabilities under the conditions of fuzziness of input and intermediate data have been determined. To this end, based on the mathematical apparatus of fuzzy network modeling, a fuzzy GERT model has been built for investigating software vulnerabilities. A distinctive feature of this model is to take into consideration the probabilistic characteristics of transitions from state to state along with time characteristics. As part of the simulation, the following stages of the study were performed. To schematically describe the procedures for studying software vulnerabilities, a structural model of this process has been constructed. A "reference GERT model" has been developed for investigating software vulnerabilities. The process was described in the form of a standard GERT network. The algorithm of equivalent transformations of the GERT network has been improved, which differs from known ones by considering the capabilities of the extended range of typical structures of parallel branches between neighboring nodes. Analytical expressions are presented to calculate the average time spent in the branches and the probability of successful completion of studies in each node. The calculation of these probabilistic-temporal characteristics has been carried out in accordance with data on the simplified equivalent fuzzy GERT network for the process of investigating software vulnerabilities. Comparative studies were conducted to confirm the accuracy and reliability of the results obtained. The results of the experiment showed that in comparison with the reference model, the fuzziness of the input characteristic of the time of conducting studies of software vulnerabilities was reduced, which made it possible to improve the accuracy of the simulation results

Keywords: *software, security testing, fuzzy GERT-model, cyber threat, software vulnerability*

UDC 004.415.53: 519.711
DOI: 10.15587/1729-4061.2021.243715

DEVELOPMENT OF A FUZZY GERT MODEL FOR INVESTIGATING COMMON SOFTWARE VULNERABILITIES

Serhii Semenov
Doctor of Technical Sciences, Professor*
Liqiang Zhang
Postgraduate Student
College of Computer Science**
Weiling Cao
Postgraduate Student
Department of IT Information Centre**
Serhii Bulba
PhD*
Vira Babenko
Doctor of Technical Sciences, Associate Professor
Department of Informational Security and Computer Engineering
Cherkasy State Technological University
Shevchenka blvd., 460, Cherkasy, Ukraine, 18006
Viacheslav Davydov
Corresponding author
PhD*
E-mail: vyacheslav.v.davydov@gmail.com
*Department of Computer Engineering and Programming
National Technical University "Kharkiv Polytechnic Institute"
Kyrpychova str., 2, Kharkiv, Ukraine, 61002
**Neijiang Normal University
705 Dongtong Rd, Dongxing District, Neijiang, Sichuan, China

Received date 28.09.2021
Accepted date 08.11.2021
Published date 29.12.2021

How to Cite: *Semenov, S., Zhang, L., Cao, W., Bulba, S., Babenko, V., Davydov, V. (2021). Development of a fuzzy GERT model for investigating common software vulnerabilities. Eastern-European Journal of Enterprise Technologies, 6 (2 (114)), 6–18. doi: https://doi.org/10.15587/1729-4061.2021.243715*

1. Introduction

The current level of threats to the security of software and the increasing requirements of customers for its provision predetermine the need for a number of specialized measures (security testing procedures). Most of these activities are carried out in accordance with procedures [1] that minimize individual risks of cyber threats.

The process of security testing implies the implementation of a complex set of algorithms and procedures that take into consideration the various modes of operation of computer systems and software, as well as subjective factors of interaction in human-machine systems. At the same time, it is known that the main tool for reducing the time of research and obtaining results, as well as the possibility of their repeated and rapid repetition or clarification, are methods of mathematical modeling.

One of the necessary conditions for the application of a mathematical model is the sufficient accuracy of the results obtained. At the same time, improving the accuracy of calculations can be achieved in various ways: the construction of schemes of increased order, highlighting the main features of the solution, the extrapolation of numerical solutions obtained on a sequence of steps, etc. In each of these techniques, it is advisable to consider the factor of fuzziness of input data and uncertainty of external influences. Neglecting this factor, most often, leads to a decrease in the accuracy of the results in assessing the performance of the system. In the problems of mathematical formalization of software security testing processes, this factor becomes even more relevant.

Thus, improving the accuracy of the results of mathematical modeling of the security testing process is a relevant task. It can be resolved by improving and building a mathematical model for studying the vulnerability of software, taking into consideration the uncertainty factor of the input and intermediate test results.

Paper [2] reports a mathematical model of the first stage of identifying software vulnerabilities, the results of which can be used in the second, main, stage – investigating software vulnerabilities. At the same time, taking into consideration the uncertainties of input data and intermediate results is one of the innovative components of modeling.

2. Literature review and problem statement

Probabilistic network modeling methods remain popular among modern approaches to mathematical formalization. This is largely due to new developments of scientists and the improvement of known network approaches to modeling. As an example, we can cite the dynamic advancement of GERT models, which have become popular due to the developments reported in [3]. This is largely due to the availability of the mathematical apparatus for finding a continuous probability distribution density of the time of passage of the GERT network. One of the conditions, in this case, is that the set of distributions that can characterize the individual arcs of the model includes known (uniform, exponential, gamma, normal, etc.) distributions. In addition, it is possible to find and use continuous distributions of arbitrary types. This makes it possible to improve the accuracy of the simulation results in comparison with other network methods.

A series of improvements [4] of GERT models are related to the initial need to predict probabilistic distributions. That limited the possibilities of mathematical description of intermediate processes in this network concept and, accordingly, reduced the accuracy of the simulation results.

In work [5], an attempt was made to develop GERT models in order to unify the problems by using the Erlang distribution with different coefficients. However, that solution did not make it possible to avoid errors in the simulation results under the conditions of uncertainty of input or intermediate data.

One of the many attempts to solve the problem of analysis of fuzzy data was carried out in work [6]. At the same time, that approach did not provide for the use of fuzzy logic in network modeling structures.

Adaptation of the provisions of fuzzy mathematics in the application to the network modeling method is reported in [7]. The authors proposed to replace the probabilistic parameters of network transitions with fuzzy ones. At the same time, the weakest t-norm was used in the descriptive part of the GERT network transitions. The authors proved the effectiveness of this modeling approach in comparison with interval mathematics. However, the study of only individual fuzzy parameters (for example, only temporary) did not make it possible to unify these models and use them in cases where it is necessary to take into consideration probabilistic indicators. At the same time, it is the set of time and probabilistic indicators that makes it possible to comprehensively assess the accuracy of the simulation results.

A similar approach is used in work [8], where a fuzzy GERT model using a *z*-tag was developed. In addition, its special case of application in the formalization of the process of weapons management was considered. However, the authors did not consider the probabilistic characteristics.

Similar and other restrictions are inherent in a number of scientific articles. Thus, in work [9], the researchers conducted a fuzzy GERT simulation of the software design process. However, the authors used only the Exclusive-or nodes. That, in the end, limited the scope of practical use of the model and reduced accuracy.

In work [10], an attempt was made to eliminate the noted drawback, with the mathematical formalization of the process of assessing the complexity of technical works of architectural construction. The authors expanded the descriptive part of the internal fuzzy processes and were not limited to the Exclusive-or nodes. The results of the simulation once again emphasized the effectiveness of the use of the mathematical apparatus of fuzzy GERT-networks in the formalization of complex, ambiguous, and integrated processes.

In [11], transitions from state to state are described by a positive trapezoidal fuzzy node. However, the cited paper does not take into consideration the impact and possibilities of feedback and cycles. That, in turn, increases the complexity of the resulting models. The issue of reducing the complexity and effect of this negative factor is considered in work [12]. However, the authors also neglected the study of probabilistic characteristics.

GERT-modeling of a complex technological process to produce carbon fiber was performed by the authors of work [13]. It confirms the fact of the effectiveness of the use of the main approaches of fuzzy mathematics in network formalization schemes. However, the integrated use of fuzzy and probabilistic modeling methods was not considered in the cited work, although it is very important in the study of complex technical and technological processes. Such processes include the process of software security testing research.

Paper [14] reports a GERT model of the software penetration testing process. A given model was designed considering the ability to simplify network transformations. However, it just does not take into consideration the factor of fuzziness of internal data and processes, which introduces an error in the results of mathematical modeling.

Thus, it becomes obvious that there is a need to use fuzzy GERT networks in the mathematical formalization of the process of investigating software vulnerabilities.

3. The aim and objectives of the study

The purpose of this study is to improve the accuracy of the results of mathematical formalization of the process of investigating software vulnerabilities under the conditions of fuzziness of input and intermediate data. This will make it possible to improve the security of the software.

To accomplish the aim, the following tasks have been set:

– to build a structural model for conducting software vulnerability studies and develop an algorithm for investigating software vulnerabilities, taking into consideration such indicators as the time of the study, the probability of starting the study, the probability of successful completion of the research;

– to construct a fuzzy GERT-model for investigating software vulnerabilities;

– to develop an improved algorithm of equivalent transformations of the GERT-network;

– based on the algorithm, to improve the fuzzy GERT-model of investigating software vulnerabilities;

– to conduct comparative studies to confirm the reliability of the results obtained.

4. Research methods

A series of methods were used to solve our tasks. To build a structural model for conducting research on software vulnerabilities, methods of expert evaluation and composition, which are part of the complex of methods of system analysis, were applied. This has made it possible to synthesize the knowledge of experts in the field of software security testing into a general structure of investigating software vulnerabilities.

The development of a fuzzy GERT model for investigating software vulnerabilities was based primarily on the probabilistic method of network planning (GERT-networks). They make it possible to effectively formalize complex design processes in cases where it is difficult or impossible to unambiguously determine which activities and in what sequence should be performed to achieve the goal of the project. We have improved the GERT model based on the formalization of the provisions of the theory of fuzzy logic and their introduction into the method of network planning.

When describing the types of uncertainties of the time of vulnerability research, trapezoidal fuzzy sets (fuzzy numbers) were used.

Modernization of the GERT network was carried out using approaches for simplifying equivalent transformations that reduce the computational complexity of the mathematical model.

Comparative evaluation of the GERT model for investigating software vulnerabilities was carried out on the basis of the experimental results using the engineering mathematical software Mathcad.

5. Model for investigating software vulnerabilities

5. 1. The scheme of software vulnerability research

To schematically describe procedures for investigating software vulnerabilities, a structural model of this process has been built (Fig. 1). It should be noted that the implementation of the set of analysis methods shown in Fig. 1 in full is advisable to carry out for testing the security of software systems of critical application. In cases of less budgetary projects, it is possible to neglect certain methods of analysis, for example, the method of manual analysis in the presence of expert and dynamic analyses results.

The structural model shown in Fig. 1 could identify the following vulnerabilities recommended by MITRE:

– errors in processing user input/output data (CWE – 78, 79, 89, 119, 134, 189, 352, 434);
– security function errors (CWE – 21, 200, 255, 264, 287, 310);
– synchronization errors (CWE – 162, 399, 829, 834);
– errors of using programming interfaces (CWE – 583, 684);
– errors in environment (CWE – 16, 733);
– disadvantages of error handling (CWE – 703);
– encapsulation errors (CWE – 653);
– poor code quality (CWE – 477).

In work [14], the "reference GERT-model" for investigating software vulnerabilities was presented. At the same time, this process was described in the form of a standard GERT network, Fig. 2.

This model can be interpreted as follows. Node 1 corresponds to the initial status "The preliminary stage of preparation for investigating software vulnerabilities was passed. The necessary package of documentation, source and executable codes have been collected." Node 2 interprets the status "Expert analysis was conducted". Node 3 – the status "Static analysis was carried out". Node 4 corresponds to the status "Dynamic analysis was performed". Node 5 – the status "Manual analysis was carried out". Node 6 – the status "Procedures for decision-making and confirmation of software vulnerabilities have been carried out".

The corresponding branches of the model formalize the direct implementation of the planned algorithms and procedures for software research, as well as decision-making about software vulnerabilities. In particular, the transition (1–2) formalizes the process of expert analysis. Transitions (1–3) and (2–3) correspond to the procedures for static analysis of software vulnerabilities. Transitions (1–4), (2–4), (3–4) formalize algorithms and procedures for dynamic analysis and evaluation of the test object. It should be noted that these procedures should take into consideration the fuzziness of the input and output data. Transitions (1–5) and (3–5) characterize the process of manual software analysis. Transitions (2–6), (3–6), (4–6), and (5–6) describe one of the most complex processes in terms of mathematical formalization, the decision-making process, and confirmation of software vulnerabilities. Transitions (3–1), (4–1), and (5–1) are possible if the input data are insufficient and formalize the processes of requests for their repetition.

It should be noted that a given model does not take into consideration the procedures for re-examination after correcting possible security errors.

The equivalent *W*-function of the process of preparing for vulnerability studies can be represented as the following expression:

$$W_E(s)=\frac{\begin{pmatrix}W_{12}W_{26}+W_{12}W_{24}W_{46}+W_{12}W_{23}W_{36}+W_{12}W_{23}W_{34}W_{46}+W_{12}W_{23}W_{35}W_{56}+\\+W_{13}W_{34}W_{46}+W_{13}W_{35}W_{56}W_{45}+W_{13}W_{36}+W_{14}W_{46}+W_{15}W_{56}\end{pmatrix}}{\begin{pmatrix}1-W_{12}W_{24}W_{41}-W_{12}W_{23}W_{31}-W_{12}W_{23}W_{34}W_{41}-W_{12}W_{23}W_{35}W_{51}-\\-W_{13}W_{34}W_{41}-W_{13}W_{35}W_{51}-W_{13}W_{31}-W_{14}W_{41}-W_{15}W_{51}\end{pmatrix}}. \quad (1)$$

In accordance with expression (1), the characteristics of the branches and the distribution parameters are given in the form of Table 1.

Table 1

Characteristics of the software vulnerability research model branches

No. of entry	Branch	*W*-function	Probability	Generating function of moments
1	(1,2)	W_{12} (1)		
2	(1,3)	W_{13}	p_1	$\lambda_1/(\lambda_1-s)$
3	(2,3)	W_{23}	p_1	$\lambda_1/(\lambda_1-s)$
4	(3,4)	W_{34}	p_2	$\lambda_2/(\lambda_2-s)$
5	(4,5)	W_{45}	p_3	$\lambda_3/(\lambda_3-s)$
6	(4,2)	W_{42}	p_4	$\lambda_4/(\lambda_4-s)$
7	(4,1)	W_{41}	p_5	$\lambda_5/(\lambda_5-s)$

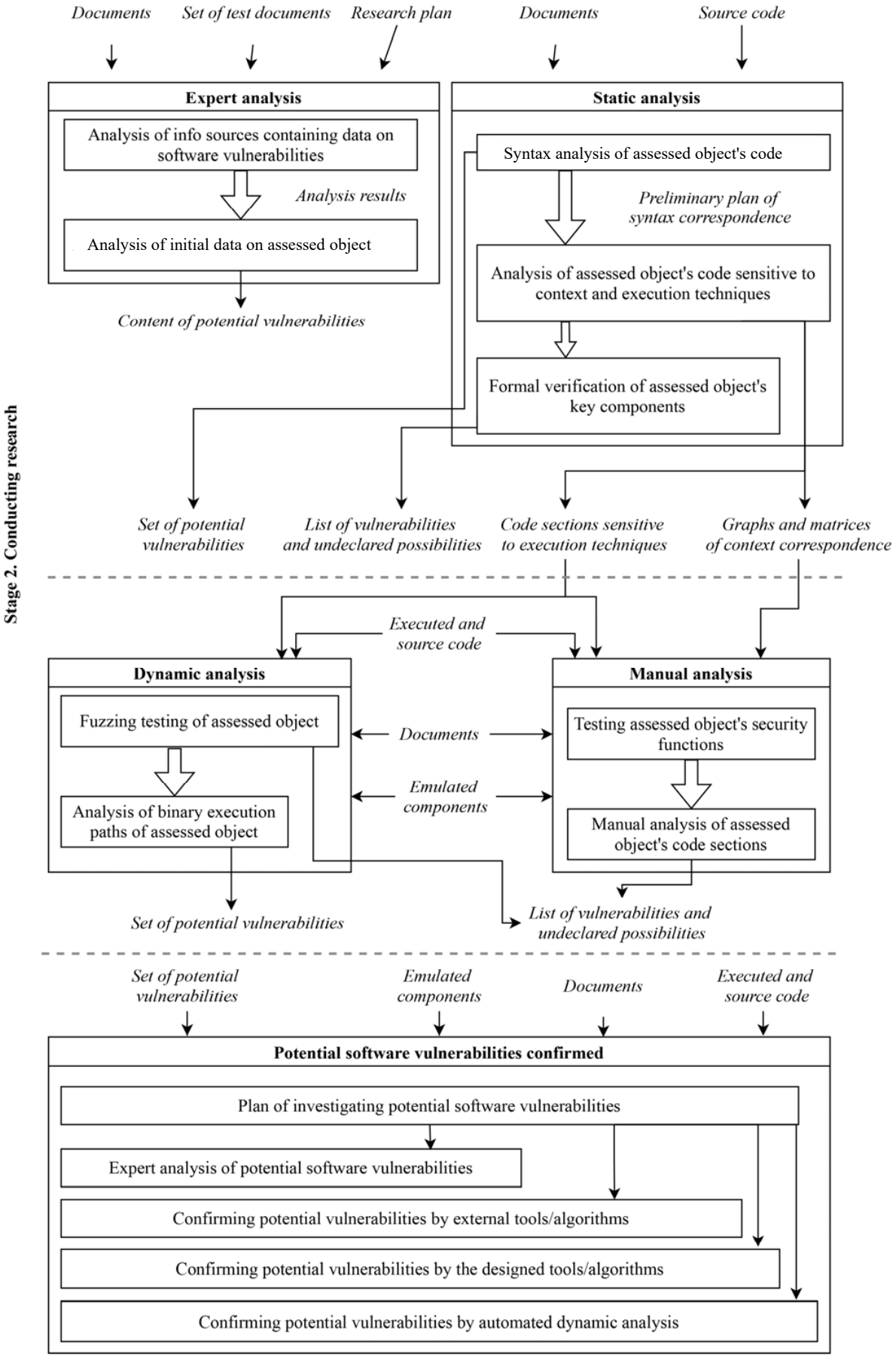


Fig. 1. Software vulnerability research scheme

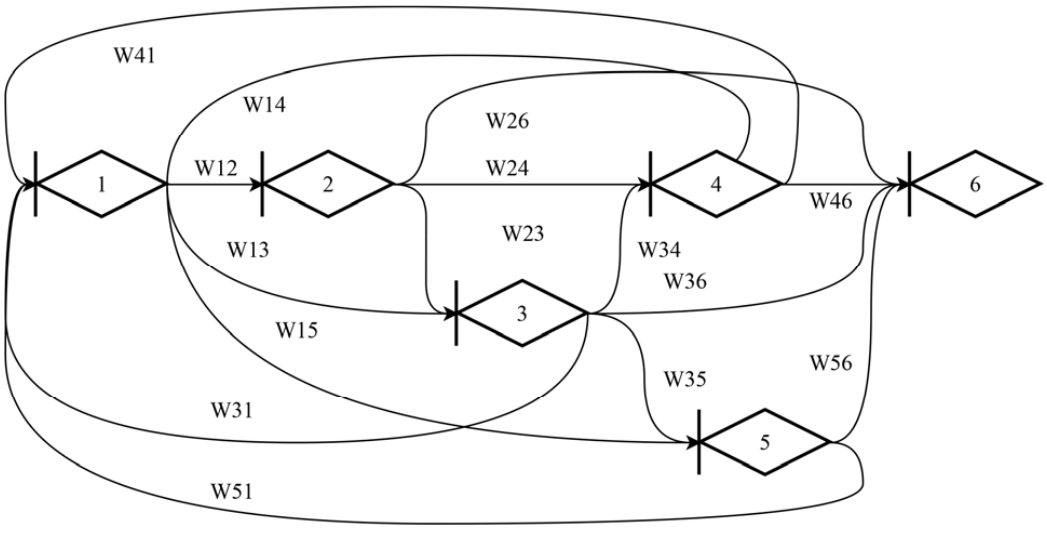


Fig. 2. Diagram of the GERT network of the software vulnerability research process

Then

$$W_E(s)= \frac{\frac{p_1\lambda_1 p_2\lambda_2 p_3\lambda_3\left(Wk_E(s)+1\right)}{\left(\lambda_1-s\right)\left(\lambda_2-s\right)\left(\lambda_3-s\right)}}{1-\frac{\left\{\begin{array}{l}\left(2p_1\lambda_1 p_2\lambda_2 p_4\lambda_4\left(\lambda_1-s\right)\left(\lambda_5-s\right)\left(Wk_E(s)+1\right)\right)+\\ +\left(2p_1\lambda_1 p_2\lambda_2 p_5\lambda_5\left(\lambda_1-s\right)\left(\lambda_4-s\right)\right)\end{array}\right.}{\left(\lambda_1-s\right)\left(\lambda_2-s\right)\left(\lambda_4-s\right)\left(\lambda_5-s\right)}}. \quad (2)$$

Table 1 shows that the generating function of moments of almost all transitions is described by the exponential law of distribution. At the same time, the totality of the presented steps and their interpretation can make it possible to form an arbitrary equivalent function.

5. 2. A fuzzy GERT model for investigating software vulnerabilities

It is advisable to introduce several restrictions and assumptions related to the structure of the GERT network and the formalization of its branches:

1. Trapezoidal fuzzy numbers are used when estimating the time of investigating software vulnerabilities. This assumption is due to the convenience of representing and calculating this indicator, as well as the clarity of the linear membership function.

2. The structural elements of the GERT network are characterized by the following features: when describing the input parts, typical structures are used in accordance with Table 2; when describing the output parts, probabilistic characteristics are used.

3. The uncertainty of the input and resulting data is characterized by a probabilistic type.

4. The maximum number of parallel branches is three.

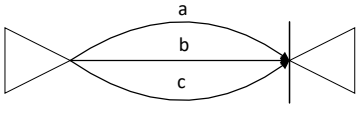
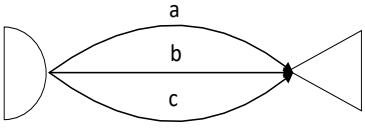
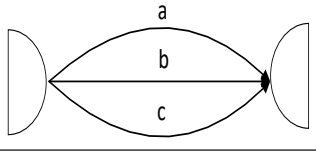
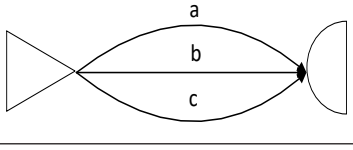
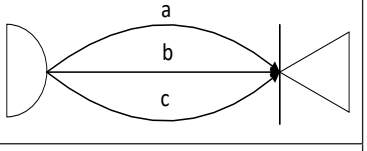
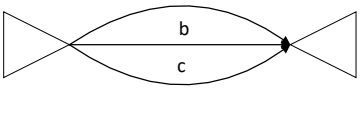
We also introduce definitions, limitations, and assumptions that relate to the mathematical descriptive component of the software vulnerability research model.

5. Evaluated parameters for investigating software vulnerabilities: study time t_{ij}, the probability of starting the analysis $p_{i,j}^{(\text{starting analysis})}$, the probability of successful study completion $p_{i,j}^{(\text{useful conclusion})}$.

6. A fuzzy set $\tilde{S}$ is the set of pairs $\left\{\left(x,\mu_{\tilde{S}}(x)\right)\middle|x\in X\right\}$, where $\mu_{\tilde{S}}(x):R\to[0,1]$ is the fuzzy set membership function.

Table 2

Typical structures of parallel branches between neighboring nodes

No.	Description	Representation
1	Parallel transitions between two nodes with «probabilistic» output and input «Exclusive-Or»	
2	Parallel transitions between two nodes with «deterministic» output and input «Inclusive-Or»	
3	Parallel transitions between two nodes with «deterministic» output and input «And»	
4	Parallel transitions between two nodes with a «probabilistic» output and an input «And»	
5	Parallel transitions between two nodes with «deterministic» output and input «Exclusive-Or»	
6	Parallel transitions between two nodes with «probabilistic» output and input «Inclusive-Or»	

7. A convex fuzzy set $\tilde{S}$ is such a fuzzy set in which:

$$\forall x, y \in R, \forall \lambda \in [0,1],$$

$$\mu_{\tilde{S}}(\lambda x + (1-\lambda) y) \geq$$

$$\geq \min\left(\mu_{\tilde{S}}(x), \mu_{\tilde{S}}(y)\right).$$

8. A fuzzy set $\tilde{S}$ is denoted positive if its membership function is such that: $\mu_{\tilde{S}}(x) = 0, \forall x \leq 0$.

9. A trapezoidal fuzzy number is a convex fuzzy set that is defined as $\tilde{S} = \left(x, \mu_{\tilde{S}}(x)\right)$, where

$$\mu_{\tilde{S}} = \begin{cases} 0, & x \leq a, \\ \dfrac{x-a}{b-a}, & a < x \leq b, \\ 1, & b < x \leq c, \\ \dfrac{x-d}{c-d}, & c < x \leq d, \\ 0, & x > d. \end{cases}$$

10. A trapezoidal fuzzy number $\tilde{S} = (a, b, c, d)$ is denoted a positive trapezoidal fuzzy number if: $0 \leq a \leq b \leq c \leq d$.

11. Considering $S_1 = (a_1, b_1, c_1, d_1)$ and $S_2 = (a_2, b_2, c_2, d_2)$ as two positive trapezoidal fuzzy numbers, the fuzzy operators are determined as follows:

– addition $\oplus$: $S_1 \oplus S_2 = (a_1 + a_2, b_1 + b_2, c_1 + c_2, d_1 + d_2)$;

– subtraction Θ: $S_1 \Theta S_2 = (a_1 - d_2, b_1 - c_2, c_1 - b_2, d_1 - a_2)$;

– multiplication $\otimes$: $S_1 \otimes S_2 \approx (h \times a_2, b_1 \times b_2, c_1 \times c_2, d_1 \times d_2)$;

– $h \otimes S_2 \approx (h \times a_2, h \times b_2, h \times c_2, h \times d_2)$;

– division $\div$: $\tilde{S}_1 \div \tilde{S}_2 \approx \left(\dfrac{a_1}{d_2}, \dfrac{b_1}{c_2}, \dfrac{c_1}{b_2}, \dfrac{d_1}{a_2}\right)$;

– maximum:

$$\max\left\{\tilde{S}_1, \tilde{S}_2\right\} \approx \left((a_1 \vee a_2), (b_1 \vee b_2), (c_1 \vee c_2), (d_1 \vee d_2)\right).$$

The improved algorithm for simplifying equivalent transformations.

Considering the scheme shown in Fig. 2, and taking it as a basis, it must be remembered that the ultimate goal of this stage of software security research is to form a set of vulnerabilities and undeclared software capabilities, as well as tools and algorithms for confirming vulnerabilities. At the same time, its distinctive feature is the use of the mathematical apparatus of fuzzy data to confirm software vulnerabilities.

The factor of the presence of a separate class of fuzzy input data determines the need to use the appropriate mathematical apparatus in modeling. At the same time, the choice of dynamic analysis techniques is based on taking into consideration the logic of fuzzy data. Therefore, it is advisable to transform the scheme of investigating software vulnerabilities (Fig. 1) and the GERT network of the software vulnerability research process (Fig. 2) into a fuzzy GERT network and represent it in the form shown in Fig. 3.

Fig. 3 shows that the structure is complex and has a number of elements that are subject to simplifying equivalent transformations.

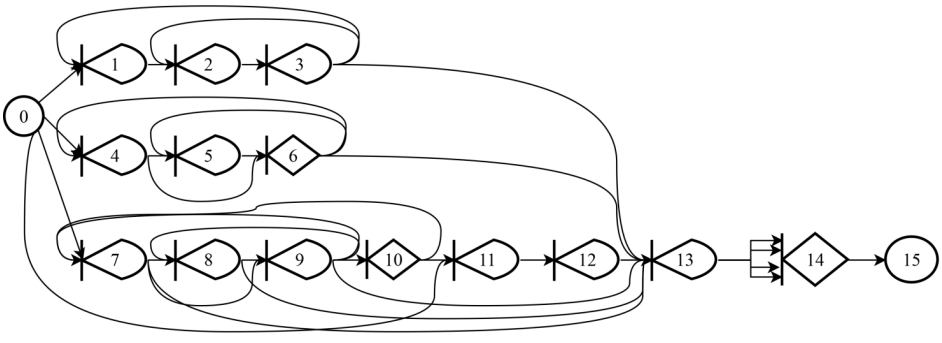


Fig. 3. A fuzzy GERT network of software vulnerability research process

5. 3. The advanced algorithm of equivalent transformations of GERT-network

To carry out simplifying equivalent transformations of the GERT network for investigating software vulnerabilities, the main ideas from scientific research [15] were used. At the same time, we shall improve the algorithm of simplifying equivalent transformations by taking into consideration the three evaluated parameters of software vulnerability research: the study time t_{ij}, the probability of the beginning of the analysis $p_{i,j}^{(\text{starting analysis})}$, and the probability of successful study completion $p_{i,j}^{(\text{useful conclusion})}$.

The block diagram of the improved algorithm of simplifying equivalent transformations is shown in Fig. 4. Take a closer look at some of the main stages and steps of the algorithm.

At the initial stage of the study, for a reasoned assessment of reversible transitions and their description, three parameters are considered: the fuzzy study time t_{ij}, the probability of starting the analysis $p_{i,j}^{(\text{starting analysis})}$, and the probability of successful study completion $p_{i,i}^{(\text{useful conclusion})}$. The existence of uncertainty in the number of repetitions of the passage of the input and output points of the nodes (i) can be formalized using a geometrically estimated value

$$Ge\left(x; p_{i,j}^{(\text{starting analysis})}\right) = \left(p_{i,i}^{(\text{starting analysis})}\right)^x \left(1 - p_{i,i}^{(\text{starting analysis})}\right),$$

where x is the number of repetitions of branches (i, i).

This expression is the basis for calculating the temporal and probabilistic characteristics of investigating software vulnerabilities. The calculation data are given in Tables 3, 4.

Analytical expressions for calculating the average time spent in branches not associated with a node (i) can be represented by steps using Table 3.

Table 3

Analytical expressions for calculating the average time spent in branches not associated with a node (i) (added average time spent in branches)

No.	Probability of successful start of analysis	Added average time spent in branches
0	$1 - p_{i,i}^{(\text{starting analysis})}$	0
1	$p_{i,i}^{(\text{starting analysis})}\left(1 - p_{i,i}^{(\text{starting analysis})}\right)$	$\tilde{t}_{i,i}$
2	$\left(p_{i,i}^{(\text{starting analysis})}\right)^2\left(1 - p_{i,i}^{(\text{starting analysis})}\right)$	$2\tilde{t}_{i,i}$
⋮	⋮	⋮
N	$\left(p_{i,i}^{(\text{starting analysis})}\right)^n\left(1 - p_{i,i}^{(\text{starting analysis})}\right)$	$n\tilde{t}_{i,i}$
⋮	⋮	⋮

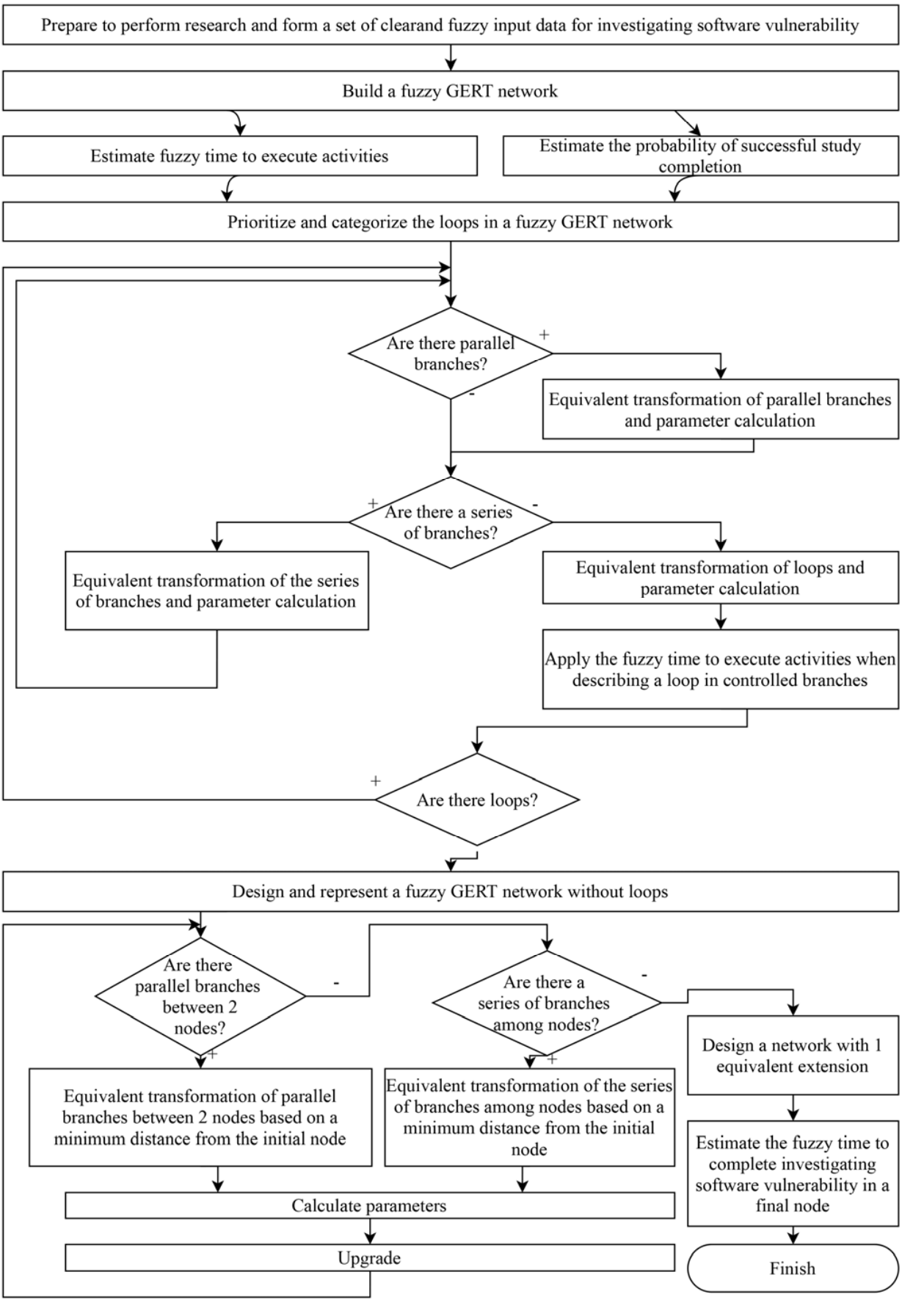


Fig. 4. Block diagram of the improved algorithm for simplifying equivalent transformations

Table 4

Analytical expressions for calculating the probability of successful completion of the investigation in each node *i*

No.	Probability of successful start of analysis	Probability of successful branch transition from node *i* to node *j*
0	$1-p_{i,j}^{(\text{starting analysis})}$	$p_{i,j}^{(\text{useful conclusion})}$
1	$p_{i,j}^{(\text{starting analysis})}\left(1-p_{i,j}^{(\text{starting analysis})}\right)$	$p_{i,j}^{(\text{useful conclusion})}\cdot p_{i,i}^{(\text{useful conclusion})}$
2	$\left(p_{i,j}^{(\text{starting analysis})}\right)^2\left(1-p_{i,j}^{(\text{starting analysis})}\right)$	$p_{i,j}^{(\text{useful conclusion})}\cdot\left(p_{i,i}^{(\text{useful conclusion})}\right)^2$
⋮	⋮	⋮
N	$\left(p_{i,j}^{(\text{starting analysis})}\right)^n\left(1-p_{i,j}^{(\text{starting analysis})}\right)$	$p_{i,j}^{(\text{useful conclusion})}\cdot\left(p_{i,i}^{(\text{useful conclusion})}\right)^n$
⋮	⋮	⋮

In accordance with the laws in Table 3, the value of the added fuzzy investigation time $\vec{t}_{i,i}$ is equal to

$$\vec{t}_{i,j}=\left(\begin{array}{l}0\left(1-\\-p_{i,i}^{(\text{starting analysis})}\right)\oplus\\\oplus\left(\begin{array}{l}p_{i,i}^{(\text{starting analysis})}\times\\\times\left(\begin{array}{l}1-\\-p_{i,i}^{(\text{starting analysis})}\end{array}\right)\otimes\tilde{t}_{i,i}\end{array}\right)\end{array}\right)\oplus$$
$$\oplus\left(\begin{array}{l}2\left(p_{i,i}^{(\text{starting analysis})}\right)^2\times\\\times\left(\begin{array}{l}1-\\-p_{i,i}^{(\text{starting analysis})}\end{array}\right)\otimes\left(\tilde{t}_{i,i}\right)\end{array}\right)\oplus\ldots$$
$$\oplus\left(\begin{array}{l}n\left(p_{i,i}^{(\text{starting analysis})}\right)^n\times\\\times\left(\begin{array}{l}1-\\-p_{i,i}^{(\text{starting analysis})}\end{array}\right)\otimes\left(\tilde{t}_{i,i}\right)\end{array}\right)\oplus\ldots=$$
$$=\sum_{x=0}^{\infty}\oplus\left(\begin{array}{l}x\left(p_{i,i}^{(\text{starting analysis})}\right)^x\times\\\times\left(\begin{array}{l}1-\\-p_{i,i}^{(\text{starting analysis})}\end{array}\right)\otimes\left(\tilde{t}_{i,i}\right)\end{array}\right)=$$
$$=\sum_{x=0}^{\infty}\left(\begin{array}{l}x\left(p_{i,i}^{(\text{starting analysis})}\right)^{x-1}\times\\\times p_{i,i}^{(\text{starting analysis})}\cdot\left(1-p_{i,i}^{(\text{starting analysis})}\right)\otimes\left(\tilde{t}_{i,i}\right)\end{array}\right).\qquad(3)$$

By assuming

$$\sum_{x=0}^{\infty}\left(x\left(p_{i,i}^{(\text{starting analysis})}\right)^{x-1}\right)\approx\frac{1}{\left(1-p_{i,i}^{(\text{starting analysis})}\right)^2},$$

expression (3) is simplified to obtain

$$\vec{t}_{i,j}=\frac{p_{i,i}^{(\text{starting analysis})}}{\left(1-p_{i,i}^{(\text{starting analysis})}\right)}\otimes\tilde{t}_{i,i}.\qquad(4)$$

After calculating the fuzzy time for a node (*i*), the value of expression (4) can be synthesized with the time values of all output branches from the node (*i*) according to the following expression

$$\vec{t}_r=\tilde{t}_r\oplus\left(\frac{p_{i,i}^{(\text{starting analysis})}}{\left(1-p_{i,i}^{(\text{starting analysis})}\right)}\otimes\tilde{t}_{i,i}\right);\ \ \forall r\in Z_{Y_i}.\qquad(5)$$

Consider the next step in the equivalent transformation of a fuzzy GERT network – determining the change in the probability of successful completion of a particular investigation that does not belong to node (*i*) by excluding the branch (*i, i*). To this end, assume that the branch (*i–j*) with the parameters t_{ij}, $p_{i,j}^{(\text{starting analysis})}$ and $p_{i,i}^{(\text{useful conclusion})}$ is a branch of the node (*i*). Using expressions to calculate the probability of successful transition to branches from node *i* to node *j*, given in Table 4, we obtain the following analytic expressions

$$\hat{p}_{i,j}^{(\text{useful conclusion})}=p_{i,j}^{(\text{useful conclusion})}\cdot\left(1-p_{i,i}^{(\text{starting analysis})}\right)+$$
$$+p_{i,i}^{(\text{useful conclusion})}\cdot p_{i,i}^{(\text{starting analysis})}\cdot p_{i,i}^{(\text{starting analysis})}\times$$
$$\times\left(\begin{array}{l}1-\\-p_{i,i}^{(\text{starting analysis})}\end{array}\right)+p_{i,j}^{(\text{useful conclusion})}\times$$
$$\times\left(p_{i,i}^{(\text{useful conclusion})}\right)^2\times$$
$$\times\left(p_{i,i}^{(\text{starting analysis})}\right)^2\cdot\left(\begin{array}{l}1-\\-p_{i,i}^{(\text{starting analysis})}\end{array}\right)+$$
$$+\ldots+p_{i,i}^{(\text{useful conclusion})}\cdot\left(p_{i,i}^{(\text{useful conclusion})}\right)^n\times$$
$$\times\left(p_{i,i}^{(\text{starting analysis})}\right)^n\cdot\left(1-p_{i,i}^{(\text{starting analysis})}\right)+\ldots=$$
$$=\sum_{n=0}^{\infty}p_{i,j}^{(\text{useful conclusion})}\cdot\left(p_{i,i}^{(\text{useful conclusion})}\right)^n\times$$
$$\times\left(p_{i,i}^{(\text{starting analysis})}\right)^n\cdot\left(1-p_{i,i}^{(\text{starting analysis})}\right)=$$
$$=p_{i,j}^{(\text{useful conclusion})}\cdot\left(1-p_{i,i}^{(\text{starting analysis})}\right)\times$$
$$\times\sum_{n=0}^{\infty}\left(p_{i,i}^{(\text{useful conclusion})}\right)^n\cdot\left(p_{i,i}^{(\text{starting analysis})}\right)^n.\qquad(6)$$

Thru the simplification transformation similar to expression (5), we obtain

$$\hat{p}_r^{(\text{useful conclusion})}=$$
$$=p_r^{(\text{useful conclusion})}\cdot\left(1-p_{i,i}^{(\text{starting analysis})}\right)\times$$
$$\times\left(1+\frac{p_{i,i}^{(\text{useful conclusion})}\cdot p_{i,i}^{(\text{starting analysis})}}{1-\left(p_{i,i}^{(\text{useful conclusion})}\cdot p_{i,i}^{(\text{starting analysis})}\right)}\right);\ \forall r\in Z_{Y_i}.\qquad(7)$$

Calculating the change in the probability of the beginning of the analysis in branches that do not have a relationship with node (*i*) by excluding branch (*i, i*) can be done as follows. The exclusion of the branch (*i, i*) entails multiplying the probability of the beginning of the analysis by the value

$$1+\frac{p_{i,i}^{(\text{starting analysis})}}{\left(1-p_{i,i}^{(\text{starting analysis})}\right)}.$$

Taking into consideration the assumption of a maximum of three branches *a*, *b*, and *c*, characterized by fuzzy execution times $\tilde{t}_a=(a_1,a_2,a_3,a_4)$, $\tilde{t}_b=(b_1,b_2,b_3,b_4)$, $\tilde{t}_c=(c_1,c_2,c_3,c_4)$, we investigate the existing rules for the fuzzy description of parallel branches between neighboring nodes (Table 1).

Considering the first example from Table 1 (parallel transitions between two nodes with a "probabilistic" output and an "Exclusive-Or" input), it should be noted that there is only one way to perform these actions. To determine the equivalent time to complete the transition, it is advisable to use the average time indicator, taking into consideration the probabilities of the beginning of the analysis and the successful completion

of the investigation. One can calculate this metric for a single branch (for example, *a*) by using an expression

$$\tilde{t}_a = \frac{p_a^{(\text{starting analysis})}}{p_a^{(\text{starting analysis})} + p_b^{(\text{starting analysis})} + p_c^{(\text{starting analysis})}} \otimes \left(\begin{array}{l} \tilde{t}_a \otimes p_a^{(\text{useful conclusion})} \oplus \left(1 - p_a^{(\text{useful conclusion})}\right) \otimes \\ \otimes \left(\begin{array}{l} \frac{p_b^{(\text{starting analysis})}}{p_b^{(\text{starting analysis})} + p_c^{(\text{starting analysis})}} \otimes \\ \otimes \left(\begin{array}{l} p_b^{(\text{useful conclusion})} \otimes \left(\tilde{t}_a \oplus \tilde{t}_b\right) \oplus \\ \oplus \left(\begin{array}{l} 1 - \\ - p_b^{(\text{useful conclusion})} \end{array} \right) \otimes \left(\tilde{t}_a \oplus \tilde{t}_b \oplus \tilde{t}_c\right) \end{array} \right) \oplus \\ \oplus \frac{p_b^{(\text{starting analysis})}}{p_b^{(\text{starting analysis})} + p_c^{(\text{starting analysis})}} \otimes \\ \otimes \left(\begin{array}{l} p_c^{(\text{useful conclusion})} \otimes \left(\tilde{t}_a \oplus \tilde{t}_c\right) \oplus \\ \oplus \left(\begin{array}{l} 1 - \\ - p_c^{(\text{useful conclusion})} \end{array} \right) \otimes \left(\tilde{t}_a \oplus \tilde{t}_b \oplus \tilde{t}_c\right) \end{array} \right) \end{array} \right) \end{array} \right). \quad (8)$$

Similarly, fuzzy indicators of the time of passage of branches *b* and *c* are described.

$$\tilde{t}_{e_r} = \left(\begin{array}{l} \frac{p_a^{(\text{starting analysis})}}{p_a^{(\text{starting analysis})} + p_b^{(\text{starting analysis})} + p_c^{(\text{starting analysis})}} \otimes \\ \otimes \left(\begin{array}{l} \tilde{t}_a \oplus \left(1 - p_a^{(\text{useful conclusion})}\right) \otimes \\ \otimes \left(\begin{array}{l} \frac{p_b^{(\text{starting analysis})}}{p_b^{(\text{starting analysis})} + p_c^{(\text{starting analysis})}} \otimes \\ \otimes \left(\tilde{t}_b \oplus \left(1 - p_b^{(\text{useful conclusion})}\right) \otimes \tilde{t}_c\right) \oplus \\ \frac{p_c^{(\text{starting analysis})}}{p_b^{(\text{starting analysis})} + p_c^{(\text{starting analysis})}} \otimes \\ \otimes \left(\tilde{t}_c \oplus \left(1 - p_c^{(\text{useful conclusion})}\right) \otimes \tilde{t}_b\right) \end{array} \right) \end{array} \right) \oplus \\ \oplus \frac{p_b^{(\text{starting analysis})}}{p_a^{(\text{starting analysis})} + p_b^{(\text{starting analysis})} + p_c^{(\text{starting analysis})}} \otimes \\ \otimes \left(\begin{array}{l} \tilde{t}_b \oplus \left(1 - p_b^{(\text{useful conclusion})}\right) \otimes \\ \otimes \left(\begin{array}{l} \frac{p_a^{(\text{starting analysis})}}{p_a^{(\text{starting analysis})} + p_c^{(\text{starting analysis})}} \otimes \\ \otimes \left(\tilde{t}_a \oplus \left(1 - p_a^{(\text{useful conclusion})}\right) \otimes \tilde{t}_c\right) \oplus \\ \frac{p_c^{(\text{starting analysis})}}{p_a^{(\text{starting analysis})} + p_c^{(\text{starting analysis})}} \otimes \\ \otimes \left(\tilde{t}_c \oplus \left(1 - p_c^{(\text{useful conclusion})}\right) \otimes \tilde{t}_a\right) \end{array} \right) \end{array} \right) \oplus \\ \oplus \frac{p_c^{(\text{starting analysis})}}{p_a^{(\text{starting analysis})} + p_b^{(\text{starting analysis})} + p_c^{(\text{starting analysis})}} \otimes \\ \otimes \left(\begin{array}{l} \tilde{t}_c \oplus \left(1 - p_c^{(\text{useful conclusion})}\right) \otimes \\ \otimes \left(\begin{array}{l} \frac{p_a^{(\text{starting analysis})}}{p_a^{(\text{starting analysis})} + p_b^{(\text{starting analysis})}} \otimes \\ \otimes \left(\tilde{t}_a \oplus \left(1 - p_a^{(\text{useful conclusion})}\right) \otimes \tilde{t}_b\right) \oplus \\ \frac{p_b^{(\text{starting analysis})}}{p_a^{(\text{starting analysis})} + p_b^{(\text{starting analysis})}} \otimes \\ \otimes \left(\tilde{t}_b \oplus \left(1 - p_b^{(\text{useful conclusion})}\right) \otimes \tilde{t}_a\right) \end{array} \right) \end{array} \right) \end{array} \right) = \quad (9)$$

Since parallel branches are independent of each other, one can take the following statements: The probability of the beginning of the analysis is equal to the sum of the probabilities of the beginning of the analysis of all branches. To calculate the probability of a successful investigation, a known averaging trend is used, similar to the calculation of a fuzzy investigation time. Then

$$p_{e_r}^{(\text{starting analysis})} = p_a^{(\text{starting analysis})} + p_b^{(\text{starting analysis})} + p_c^{(\text{starting analysis})}, \quad (9)$$

$$p_{e_r}^{(\text{useful conclusion})} = p_a^{(\text{useful conclusion})} + \left(1 - p_a^{(\text{useful conclusion})}\right) \left(\begin{array}{l} p_b^{(\text{useful conclusion})} + \\ + \left(1 - p_b^{(\text{useful conclusion})}\right) \times \\ \times p_c^{(\text{useful conclusion})} \end{array} \right). \quad (10)$$

The second example in Table 1 gives parallel branches of the network with a deterministic output and the input "Inclusive-Or". All processes on the network run simultaneously and end as the fastest of them is finished. When determining the equivalent time of passage of the network section, it is necessary to take into consideration the uncertainty factor, and, accordingly, perform defuzzification operations.

$$d\tilde{t}_a = \frac{\left(a_1 + 2a_2 + 2a_3 + a_4\right)}{6}.$$

After that, one can perform procedures for sorting the results based on the lowest oddity value: $Sor\tilde{t}_{\min}\left(d\tilde{t}_a, d\tilde{t}_b, d\tilde{t}_c\right)$.

Based on the assumptions made, we specify the time characteristic.

$$\tilde{t}_{e_r} = p_a^{(\text{useful conclusion})} \otimes \tilde{t}_a \oplus \left(1 - p_a^{(\text{useful conclusion})}\right) \otimes \otimes \left(p_b^{(\text{useful conclusion})} \otimes \tilde{t}_b \oplus \left(1 - p_b^{(\text{useful conclusion})}\right) \otimes \tilde{t}_c\right). \quad (11)$$

The remaining probabilistic characteristics are formalized as follows

$$p_{e_r}^{(\text{starting analysis})} = = \sum p_a^{(\text{starting analysis})}, p_b^{(\text{starting analysis})}, p_c^{(\text{starting analysis})}, \quad (12)$$

$$p_{e_r}^{(\text{useful conclusion})} = p_a^{(\text{useful conclusion})} + \left(1 - p_a^{(\text{useful conclusion})}\right) \times \times \left(p_b^{(\text{useful conclusion})} + \left(1 - p_b^{(\text{useful conclusion})}\right) \cdot p_c^{(\text{useful conclusion})}\right). \quad (13)$$

The third example in Table 1 is the case of parallel transitions between two nodes with a "deterministic" output and the input "And". Since all these branches must be performed in full, equivalent transformations can be carried out by taking into consideration the maximum of the fuzzy indicator of the investigation.

$$\tilde{t}_{e_r} = \max\left(\tilde{t}_a, \tilde{t}_b, \tilde{t}_c\right) = \left(a_1 \vee b_1 \vee c_1\right), \left(a_2 \vee b_2 \vee c_2\right), \left(a_3 \vee b_3 \vee c_3\right), \left(a_4 \vee b_4 \vee c_4\right). \quad (14)$$

Probabilistic characteristics can be calculated as follows

$$p_{e_r}^{(\text{starting analysis})} = \prod_{,p_b^{(\text{starting analysis})},p_c^{(\text{starting analysis})}} p_a^{(\text{starting analysis})}, \quad (15)$$

$$p_{e_r}^{(\text{useful conclusion})} = \sum p_a^{(\text{useful conclusion})}, p_b^{(\text{useful conclusion})}, p_c^{(\text{useful conclusion})}. \quad (16)$$

Continuing to consider the process of equivalent transformations in accordance with Fig. 4, one can see the relevance of the problem of transforming feedback or loops.

To simplify these structures and achieve the desired result, one must perform the following actions:

1. Carry out simplifying equivalent transformations in accordance with the rules given in Table 1 and expressions (9) to (16).

2. Formalize the corresponding loop in the form of the structure shown in Fig. 5.

3. Mathematically formalize equivalent parameters in the form of the following expressions

$$\tilde{t}_{\ln_{(x-y)}} = \tilde{t}_{(x-y)} \oplus \tilde{t}_{(y-x)}, \quad (17)$$

$$p_{\ln_{(x-y)}}^{(\text{starting analysis})} = p_{(x-y)}^{(\text{starting analysis})} \times p_{(y-x)}^{(\text{starting analysis})}, \quad (18)$$

$$p_{\ln_{(x-y)}}^{(\text{useful conclusion})} = p_{(x-y)}^{(\text{useful conclusion})} \times p_{(y-x)}^{(\text{useful conclusion})}. \quad (19)$$

$$\tilde{t}'_p = \tilde{t}_p \oplus \left(\left(\frac{p_{\ln_{(x-y)}}^{(\text{starting analysis})}}{1 - p_{\ln_{(x-y)}}^{(\text{starting analysis})}} \right) \otimes \tilde{t}_{\ln_{(x-y)}} \right), \quad (20)$$

$$p'^{(\text{useful conclusion})}_p = \\ = p_{(p)}^{(\text{useful conclusion})} \cdot \left(1 - p_{(x-y)}^{(\text{useful conclusion})}\right) \times \\ \times \left(1 + \left(\frac{p_{(x-y)}^{(\text{useful conclusion})} \times p_{(y-x)}^{(\text{useful conclusion})}}{1 - p_{(x-y)}^{(\text{useful conclusion})} \times p_{(y-x)}^{(\text{useful conclusion})}} \right)\right); \ \forall p' \in Z_{e_x}, \quad (21)$$

$$p'^{(\text{starting analysis})}_{p'} = \\ = p_{p'}^{(\text{starting analysis})} \times \left(1 + \frac{p_{(y-x)}^{(\text{starting analysis})}}{1 - p_{(y-x)}^{(\text{starting analysis})}}\right); \ \forall p' \in Z_{e_x}, \quad (22)$$

provided the branch has the shape of a simple loop.

Perform the following operations if the branch takes the shape of a complex loop:

– identify and categorize the existing loop by the levels of formation;

– prioritize the loops by execution time;

– select a loop based on the presented priority for all existing network contours;

– transform complex contours into simple ones and proceed to equivalent transformations in accordance with expressions (17) to (22).

5. 4. The improved fuzzy GERT model for investigating software vulnerability

In accordance with the scheme in Fig. 3, the following priorities of equivalent transformations can be identified and highlighted.

$$\left[\begin{cases} \ln_{(1-3)}^{0} & \ln_{(2-3)}^{1} \\ \text{Zero level} & \text{Simple loop} \\ \text{Complex loop} \end{cases} \right],$$

$$\left[\begin{cases} \ln_{(1-8)}^{0} & \ln_{(5-6)}^{1} \\ \text{Zero level} & \text{Simple loop} \\ \text{Complex loop} \end{cases} \right],$$

$$\left[\begin{cases} \ln_{(7-10)}^{0} & \ln_{(7-9)}^{1} \\ \text{Zero level} & \text{Complex loop} \\ \text{Complex loop} & \\ \ln_{(8-9)}^{2} & \\ \text{Simple loop} & \end{cases} \right].$$

In the first step of the transformation, one must remove the loops and calculate the updated input values. Then represent the improved fuzzy GERT-model for investigating software vulnerability without loops in the form of a diagram in Fig. 5. In this case, the input parameters of the equivalent GERT network are the values given in Table 5.

After the final calculation of probabilistic-temporal characteristics, we obtain the following values of indicators

$$t_{i,j} = (35.75, 52.58, 76.47, 81.18);$$

$$p_{i,j}^{(\text{starting analysis})} \approx 1; \quad p_{i,j}^{(\text{useful conclusion})} \approx 0.8.$$

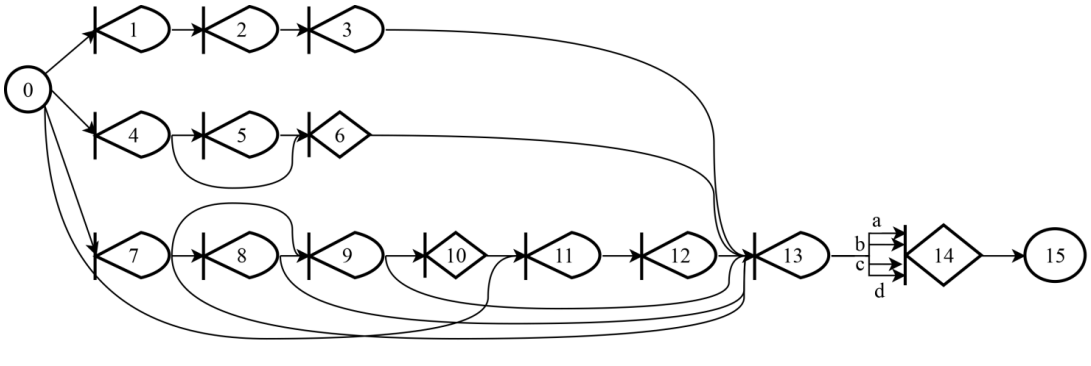


Fig. 5. Simplified equivalent fuzzy GERT network for the process of investigating software vulnerability

Table 5

Input values of parameters of equivalent GERT-network

Transition ID	Time to transformation (conditional units)	Time after transformation (conditional units)	Probability of successful completion of investigation /loop	Probability of starting analysis/loop
0–1	(6,7,8,9)	(6,7,8,9)	0.9	0.9
1–2	(1,2,3,4)	(3.53,5.33,8.03,9.89)	0.9	0.9
2–3	(1,2,3,4)	(3,4.57,5.67,7.33)	0.6	0.8
3–13	(0,0,0,0)	(0,0,0,0)	0.3	0.8
0–4	(6,7,8,9)	(6,7,8,9)	0.9	0.9
4–5	(1,2,3,4)	(3.47,5.24,7.83,9.54)	0.8	0.8
4–6	(0,1,2,3)	(2.45,4.13,6.77,8.41)	0.5	0.9
5–6	(0,1,2,3)	(2.1,3.5,3.66,6.33)	0.4	0.8
6–13	(0,0,0,0)	(0,0,0,0)	0.4	0.9
0–7	(6,7,8,9)	(6,7,8,9)	0.9	0.9
7–8	(1,2,3,4)	(10.51,14.31,17.01,20.87)	0.9	0.8
7–9	(1,2,3,4)	(3.72,5.56,8.19,9.98)	0.9	0.8
7–13	(0,0,0,0)	(0,0,0,0)	0.3	0.8
8–9	(1,2,3,4)	(2.04,4.53,5.87,7.12)	0.9	0.8
8–13	(0,0,0,0)	(0,0,0,0)	0.3	0.8
9–10	(1,1,1,1)	(3.1,3.1,3.1,3.1)	0.5	0.7
9–13	(0,0,0,0)	(0,0,0,0)	0.3	0.8
10–11	(0,0,0,0)	(0,0,0,0)	0.9	0.9
0–11	(6,7,8,9)	(6,7,8,9)	0.9	0.9
11–12	(1,2,3,4)	(1,2,3,4)	0.9	0.9
12–13	(0,0,0,0)	(0,0,0,0)	0.5	0.8
13–14 (a)	(0,1,2,3)	(0,1,2,3)	0.2	0.7
13–14 (b)	(1,2,3,4)	(1,2,3,4)	0.2	0.4
13–14 (c)	(1,2,3,4)	(1,2,3,4)	0.3	0.5
13–14 (d)	(0,1,2,3)	(0,1,2,3)	0.5	0.8
14–15	(0,0,0,0)	(0,0,0,0)	1	1

Our results can be used in the study of a fuzzy GERT model. At the same time, to compare the results of the proposed algorithm with the reference algorithms, the time after the transformation and the probability of the beginning of the analysis are computed.

5. 5. Studying the improved fuzzy GERT model

When conducting comparative studies, the following data were chosen as standards. The results of mathematical modeling of software testing presented in work [11]. Results of fuzzy GERT-modeling based on Critical Path Method (CPM) [16]. Data from practical experiments, using the model built. The values of the testing time are given in Table 6.

Table 6 demonstrates the use of the improved algorithm of equivalent transformations reduced the fuzziness of the output characteristics of the time for investigating software vulnerability by up to 1.12 times compared to the fuzzy GERT model based on CMP [16]. If we take as a basis the reference value of the deviation equal to 28.3, indicated in works [11, 16], it can be noted that the accuracy of the simulation results increased to 13 % compared to the results of mathematical modeling of software testing [11]. At the same time, it approached the results of a practical experiment.

One of the distinctive features of the developed mathematical model for investigating software vulnerability is the consideration of probabilistic characteristics of the process along with the time characteristics.

Table 6

Results of a comparative study on the criterion of minimum average time and its deviation

Model name	Fuzzy time	Time average value	Deviation
Software testing mathematical model	(53,53,53,53)	53	–
CPM-based GERT-model	(32,51,77,83)	60.75	22.25
Improved fuzzy GERT-model	(35.75, 52.58,76.47,81.18)	61.5	19.7
Practical experiment	(64,64,64,64)	64	–

To prove the reliability of the results obtained using the improved equivalent transformation algorithm, comparative studies were conducted. The results of the experiment are given in Table 7.

The results in Table 7 showed the commensurability of probabilistic and temporal indicators obtained using the improved algorithm of equivalent transformation with the values obtained from implementing known Gavareshki and Hashemin reference algorithms [10, 15]. At the same time, the improved algorithm, unlike the reference algorithms, covers a wider range of logical operations and equivalent transformations.

Table 7

Results of the comparative experiment of the improved algorithm of equivalent transformation with the reference Gavareshki and Hashemin algorithms

ID	Time after transformation (proposed algorithm)	Time after transformation (Gavareshki)	Time after transformation (Hashemin)
4–5	(3.47,5.24,7.83,9.54)	(4.5,6.5,9.1,10.6)	(5.67,10.04,15.52,19.31)
7–8	(10.51,14.31,17.01,20.87)	(11.2,15.9,21.11,25.9)	(13.3,17.81,23.42,28.02)
ID	Analysis start probability (proposed algorithm)	Analysis start probability (Gavareshki)	Analysis start probability (Hashemin)
4–5	0.8	0.85	0.9
7–8	0.8	0.85	0.9

6. Discussion of results of studying the improved fuzzy GERT-model

A fuzzy GERT model for investigating software vulnerabilities has been constructed. The developed model has made it possible to estimate the time of successful completion of investigating software vulnerability under the conditions of uncertainty, as well as the probability of successful investigation completion. The results of mathematical modeling have made it possible to draw a conclusion about the increased accuracy in the assessment of the time for investigating software vulnerability. The results of the modeling are given in Tables 6, 7. Such an increase in the accuracy of modeling results became possible due to the synthesis of the mathematical apparatus of fuzzy logic into the GERT modeling technique. In addition, the use of the developed algorithm for simplifying equivalent transformations has also made it possible to reduce the "deviation" indicator and bring it closer to the results of a practical experiment.

A structural model for conducting research into software vulnerabilities has been built. A given structural model made it possible to include in the research process a wide range of analysis techniques and expert data on software vulnerabilities in accordance with the MITRE requirements.

The use of modeling methods with a preliminary prediction of the probabilistic distribution in problems has certain disadvantages and limitations. That reduces the accuracy of the simulation. This paper has paid attention to fuzzy methods that significantly expanded the capabilities of network modeling approaches. The combination of fuzzy and probabilistic methods has made it possible to report a new approach to solve the modeling problem in projects with networks with parallel, serial, and reversible branches of the cycle.

It should be noted that a given modeling approach has prospects for further improvement. This is due to such an unresolved disadvantage of probabilistic modeling as a significant increase in the complexity of the model with a slight complication of the network.

7. Conclusions

1. A structural model for conducting research into software vulnerabilities has been built. A feature of the structural model is the synthesis of expert, static, dynamic, and manual analysis of software, which could reveal its main vulnerabilities recommended by MITRE. On its basis, a clear GERT network for the process of investigating software vulnerability has been developed. The shortcomings of this network associated with neglect of fuzziness of input data and transient characteristics and processes have been revealed.

2. Based on the mathematical apparatus of fuzzy network modeling, a fuzzy GERT model for investigating software vulnerability has been constructed. A distinctive feature of this model is to take into consideration the probabilistic characteristics of transitions from state to state along with time characteristics. This has made it possible to increase the accuracy of modeling up to 13 %.

3. The algorithm for simplifying equivalent transformations has been improved, which differs from known ones by considering the capabilities of the extended range of typical structures of parallel branches between neighboring nodes. This has made it possible to reduce the fuzziness in the output characteristics of the time for investigating software vulnerability (a deviation from the average value) by 1.12 times.

4. Based on the algorithm, a fuzzy GERT model for investigating software vulnerability has been improved, which differs from known ones by the absence of loops in the network structure.

5. Comparative studies were conducted to confirm the reliability of our results. The results of the experiment showed the commensurability of probabilistic and temporal indicators obtained when using the improved algorithm of equivalent transformation with the values obtained from implementing known Gavareshki and Hashemin reference algorithms.

References

1. CWE Version 4.1. Available at: https://cwe.mitre.org/data/published/cwe_v4.1.pdf
2. Semenov, S., Liqiang, Z., Weiling, C., Davydov, V. (2021). Development a mathematical model for the software security testing first stage. Eastern-European Journal of Enterprise Technologies, 3 (2 (111)), 24–34. doi: https://doi.org/10.15587/1729-4061.2021.233417
3. Pritsker, A. A. B. (1977). Modeling and Analysis Using Q-GERT Networks. Wiley: distributed by Halsted Press Division of John Wiley & Sons, 420.
4. Semenova, A., Dubrovskyi, M., Savitskyi, V. (2017). A GERT model of an algorithm for analyzing security of a web application. Advanced Information Systems, 1 (1), 61–64. doi: https://doi.org/10.20998/2522-9052.2017.1.11

5. Semenov, S., Davydov, V., Lipchanska, O., Lipchanskyi, M. (2020). Development of unified mathematical model of programming modules obfuscation process based on graphic evaluation and review method. Eastern-European Journal of Enterprise Technologies, 3 (2 (105)), 6–16. doi: https://doi.org/10.15587/1729-4061.2020.206232
6. Gavrylenko, S., Chelak, V., Hornostal, O., Vassilev, V. (2020). Development of a method for identifying the state of a computer system using fuzzy cluster analysis. Advanced Information Systems, 4 (2), 8–11. doi: https://doi.org/10.20998/2522-9052.2020.2.02
7. Lin, K.-P., Wen, W., Chou, C.-C., Jen, C.-H., Hung, K.-C. (2011). Applying fuzzy GERT with approximate fuzzy arithmetic based on the weakest t-norm operations to evaluate repairable reliability. Applied Mathematical Modelling, 35 (11), 5314–5325. doi: https://doi.org/10.1016/j.apm.2011.04.022
8. Zhang, N., Yan, S., Fang, Z., Yang, B. (2021). Fuzzy GERT model based on z-tag and its application in weapon equipment management. Journal of Intelligent & Fuzzy Systems, 40 (6), 12503–12519. doi: https://doi.org/10.3233/jifs-201731
9. Lachmayer, R., Afsari, M., Hassani, R. (2015). C# method for all Types of Nodes in Fuzzy GERT. International Journal of Artificial Intelligence and Neural Networks – IJAINN, 5 (1), 57–62. Available at: https://www.researchgate.net/publication/304247081_C_method_for_all_Types_of_Nodes_in_Fuzzy_GERT
10. Radziszewska-Zielina, E., Śladowski, G. (2017). Proposal of the Use of a Fuzzy Stochastic Network for the Preliminary Evaluation of the Feasibility of the Process of the Adaptation of a Historical Building to a Particular Form of Use. IOP Conference Series: Materials Science and Engineering, 245, 072029. doi: https://doi.org/10.1088/1757-899x/245/7/072029
11. Tousheh Asl, S., Hashemin, S. S. (2018). Completion Time of Special Kind of GERT-Type Networks with Fuzzy Times for Activities. International Journal of Industrial Engineering, 5 (1), 1–8. doi: https://doi.org/10.14445/23499362/ijie-v5i1p101
12. Wang, H.-H., Zhu, J.-J., Yao, Y.-C. (2019). GERT network optimization with consideration of "time-resource" on large aircraft collaborative development. Kongzhi yu Juece/Control and Decision, 34 (2), 309–316. doi: https://doi.org/10.13195/j.kzyjc.2018.0121
13. Liu, X., Fang, Z., Zhang, N. (2017). A value transfer GERT network model for carbon fiber industry chain based on input–output table. Cluster Computing, 20 (4), 2993–3001. doi: https://doi.org/10.1007/s10586-017-0960-y
14. Semenov, S., Liqiang, Z., Weiling, C. (2020). Penetration Testing Process Mathematical Model. 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T). doi: https://doi.org/10.1109/picst51311.2020.9468039
15. Norouzi, G., Heydari, M., Noori, S., Bagherpour, M. (2015). Developing a Mathematical Model for Scheduling and Determining Success Probability of Research Projects Considering Complex-Fuzzy Networks. Journal of Applied Mathematics, 2015, 1–15. doi: https://doi.org/10.1155/2015/809216
16. Gavareshki, M. H. K. (2004). New fuzzy GERT method for research projects scheduling. 2004 IEEE International Engineering Management Conference (IEEE Cat. No.04CH37574). doi: https://doi.org/10.1109/iemc.2004.1407495