DOI: http://doi.org/10.25728/cs.2021.6.5

VULNERABILITY ANALYSIS OF COMPLEX NETWORK INFRASTRUCTURES USING A GENETIC ALGORITHM

G.G. Grebenyuk,[1] S.M. Nikishov,[2] and L.A. Sereda[3]

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

[1] ✉ grebenuk@lab49.ru, [2] ✉ nikishov@lab49.ru, [3] ✉ sereda@lab49.ru

Abstract. This paper proposes a method for analyzing the vulnerability of network infrastructures. The method uses a genetic algorithm for finding cross-sections that block delivering resources from their sources to consumers. The well-known approaches to solving network problems based on combinatorial and evolutionary approaches are considered. A feature of the proposed method is the fitness function chosen as an algorithm for calculating the number of paths in the graph when isolating the graph vertices that make up the individual. The graph reachability matrix and simple mathematical operations are adopted to optimize the fitness function and calculate the number of paths. The efficiency of the genetic algorithm compared to combinatorial methods is shown: multiple failures are found significantly faster than using exhaustive search algorithms.

Keywords: safety, engineering networks, vulnerability, reliability, combinatorial algorithms, models for damage analysis, crucial elements of an engineering network, genetic algorithms.

INTRODUCTION

The object of this research is a network infrastructure, i.e., a complex technical system (electricity, gas, etc.) [1].

Vulnerability is an internal property that makes the object susceptible to the impact of a risk source that can lead to some consequence [2]. The presence of vulnerability contributes to the realization of a threat. For technical systems under consideration, threats are destructive effects such as natural disasters, terrorist attacks, technical failures of system components, etc. The negative consequences of realizing a threat include, first of all, the lost operability of large consumers (enterprises, research centers, etc.) due to interrupting the supply of any resource. This understanding of technical system vulnerability corresponds to the definition of supply system reliability, which is directly related to the continuous supply of resources to consumers.

As an internal property, this paper considers the object's structure. It includes sets of nodes (vertices, elements), further referred to as key ones, destructive impacts on which disrupt the continuous supply of resources to consumers.

Various methods and models are used to find key nodes. One approach is the mathematical modeling of physical processes in resource supply networks (electricity, gas, heat, water, and others). The disadvantages of this approach include the need to use information about the parameters and modes of the system's operation (which should be introduced into the model), the random occurrence of emergencies, and slow calculations of the system of high-order algebraic and differential equations for complex objects.

Topological analysis methods are preferable for systems represented by a graph [3–6]. This approach is convenient to implement since only the description of the graph structure (vertices and connections between them) is required. The disadvantages of the topological model are redundant solutions due to neglecting the system's modes and parameters. These disadvantages are partially eliminated when passing to weighted graphs, in which the edges and vertices are assigned network characteristics such as capacity, power, etc.

Topological methods yield acceptable results for small-dimension networks (tens of vertices). However, the computation time grows significantly with increasing the network scale. For example, during the exhaustive search of multiple failures, the number of possible alternatives is determined by the number of k-combinations of n elements. (Here, k is the number of damaged elements, and n denotes the total number of system elements.) As a result, the computation time has exponential growth. The well-known methods for constructing minimal cross-sections, such as Petrick's method or the method of disjoint sets [7, 8], do not eliminate this drawback.

The genetic algorithm reduces computational problems when searching key elements. It finds the minimum (or maximum) of a certain function (fitness function) characterizing the infrastructure state. The main advantages of the algorithm include the following:

– The fitness function can be represented by an algorithm.

– The implementation is simple.

– The discontinuities in the fitness function do not affect the solution search.

The main drawback of genetic algorithms is associated with the uncertainty of finding the global optimum. However, when searching key elements using the method proposed below, this disadvantage can be overcome.

The evolutionary approach has various applications in network analysis. For example, a genetic algorithm was used in [9, 10] to assess the vulnerability of a power system (and the system operator's response) when disconnecting its elements.

In the report [11], a genetic algorithm was used to analyze the vulnerability of electrical networks for two optimization levels. When optimizing the upper level, the maximum damage to the power system was determined in terms of the load-off. When optimizing the lower level, this damage was minimized by choosing the optimal operating mode of the power system. A peculiarity of the model [11] is that the system operator can change the network topology among various corrections available.

The paper [12] considered optimal solutions for the maintenance of infrastructure objects. Optimal solutions were those minimizing the network life cycle cost under the reliability and functionality requirements. A Markov chain model was used to predict the efficiency of infrastructure objects.

The report [13] presented an approach to finding the best ways of protecting infrastructure assets (adding or changing infrastructure in response to an emergency, etc.) for complex national and international network structures such as transport, telecommunications, finance, energy, etc. Also, their interconnections were investigated.

The genetic algorithm proposed in the paper [14] allows organizing the joint work of the system operator and the power system. The algorithm calculates control corrections to minimize the power system load-off. As a result, the network elements with the most severe consequences in case of failures were identified.

Potential vulnerabilities in a power system [15] can be identified by determining the power transmission lines causing maximum network disruption in case of failures. The AC power equations were adopted in the network infrastructure model. The failures were initiated by increasing the resistance of the transmission lines. As a result, the authors identified those transmission lines for which minor conductivity disturbances lead to serious network disruptions, voltage drops, and disconnection of consumers.

The genetic algorithm was applied to find vulnerable sections of power transmission lines [16]. Vulnerability assessment of lines allows identifying problematic areas of such infrastructure by modeling and, second, assessing the possibility of cascade failures.

The paper [17] proposed two approaches based on genetic algorithms to improve the system voltage stability under various operating conditions. Within the first approach, a correction is used to optimize the voltage stability index during abnormal control. The second approach involves finding an optimal arrangement of compensators and generator control to minimize the voltage stability index.

An optimal location of energy storage systems to reduce the power system's vulnerability was considered in [18]. The authors analyzed the impacts on busbars and searched the most vulnerable ones. An optimal location was chosen using a genetic algorithm.

The paper [19] was devoted to genetic algorithm-based optimal solutions for protecting and restoring infrastructure in case of accidents or disasters and identifying the assets necessary to maintain the network's operating mode.

Genetic algorithms are often used to tune neural networks [20–22].

The publications cited differ in the field of research, particular problems solved, and the fitness function chosen. Below, we consider the problem of finding key elements in engineering networks based on a specially constructed fitness function.

1. PROBLEM STATEMENT

The technical systems under consideration have a network organization that can be represented by a nondirected or partially directed graph $G = \{V, R\}$, where V and R denote the sets of vertices and edges, respectively [23].

The set of vertices V is described by a triple $\{S, C, U\}$, where S and C are the sets of power sources and consumers, respectively, and U denotes the set of network vertices in which power transformation (transformative stations), power distribution (power distribution plants, taps), and power transmission (power lines) are implemented:

$$V = S \bigcup C \bigcup U.$$

Also, we denote by C_z a given subset of the most important consumer-vertices and by V_α the subset of key vertices to be found.

The problem statement is as follows: for the graph $G = \{V, R\}$, it is required to determine the minimum subset of key vertices $V_\alpha \in U$ such that their removal from the graph G will violate the reachability of all vertices C_z from the vertices S.

Genetic algorithms involve such concepts as individuals, populations, chromosomes, and fitness functions. For the problem under consideration, these concepts are defined below.

An individual $\theta_d = \{V_\alpha\}$ is a set of key vertices V_α whose failure will disconnect important consumers. The subscript d is the individual's number in a population, and α is the number of graph vertices of an individual. The number of graph vertices contained in one individual, α, is advisable to choose according to the number of simultaneously occurring failures (single, double, triple, etc.) assessed in terms of their impact on the system: $V_\alpha \in U$.

The population is $P = (\theta_1, \theta_{2,\ldots,}\theta_k)$, where k denotes the size of a subset from the set of individuals used to select the best ones. The value k is fixed and related to the graph dimension n (the total number of graph vertices). Usually, the number k is chosen in the percentage of the graph dimension n, ranging from 5% to 15% (k/n = 0.05–0.15). For high-dimensional problems, the range of k can be smaller, e.g., k/n = 0.03–0.10.

A chromosome is a numerical vector (or string) representing a particular individual as a binary string of bits (genomes). For instance, the chromosome of the individual $\theta_5 = (v_3, v_2, v_{15}, v_{50})$, containing four vertices (a quadruple failure), is shown in Fig. 1.

θ_5			
000011	000010	001111	110010
v_3	v_2	v_{15}	v_{50}

Fig. 1. An individual represented by a chromosome.

The fitness function $F(\theta_d)$ allows finding an individual with the greatest effect on the total number of routes for supplying resources to consumers. We estimate this value algorithmically [24], counting the number of limited-length paths (3–5 steps) between the graph vertices after isolating the elements that make up the individual. We calculate the number of paths by raising the adjacency matrix to a power. When raising an adjacency matrix E to the power m by ordinary arithmetic operations (instead of Boolean algebra rules), its element $e_{i,j}$ will equal the number of paths of length m from vertex i to vertex j; see [25].

A small number of steps m (hence, a small exponent) is dictated by the need to reduce the algorithm's running time when raising the matrix to a power. We make the following assumption: if isolating the vertices that make up an individual reduces the total number of paths in the graph for small exponents m, it will also decrease the number of paths for supplying resources from sources to consumers. The availability of resource supply paths is tested in three stages:

1. A genetic algorithm finds an individual (a set of vertices) whose isolation will reduce to the greatest extent the total number of paths of length m in the graph.
2. The availability of paths for supplying resources from sources to consumers is tested.
3. If there are no paths from sources to consumers, the vertices are key, and the calculations finish; otherwise, the calculations are repeated.

The flow chart of the genetic algorithm is presented in Fig. 2.

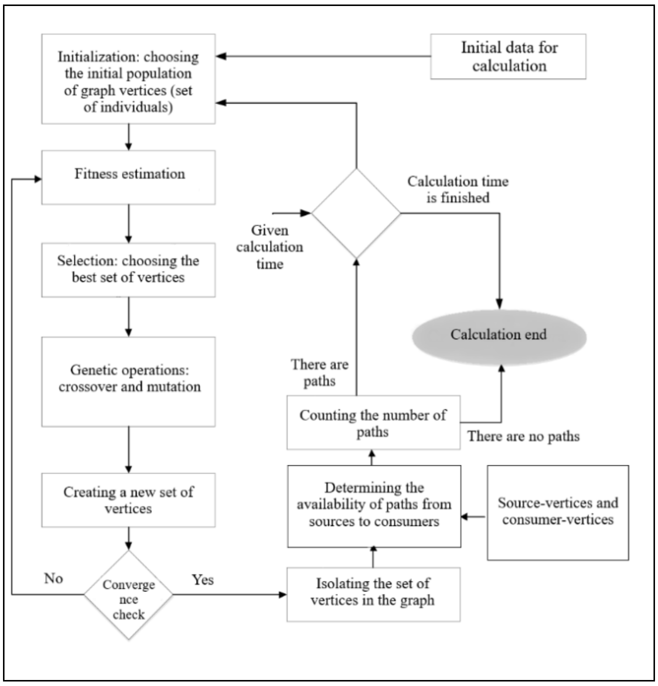

Fig. 2. Block-scheme of a genetic algorithm for finding network key vertices.

2. APPLICATION OF A GENETIC ALGORITHM

2.1. Preparing initial data for calculation

The initial information on the graphs is usually given by tables of paired relations or adjacency lists. The initial information is arranged by assigning serial numbers from 1 to n to all network vertices, $V = (v_1, v_2, \ldots, v_n)$. Then the adjacency matrix E of the graph is formed:

$$E = \left(e_{i,j}\right),$$

$$e_{i,j} = 1 \text{ if } \left(v_i, v_j\right) \in V,$$

$$e_{i,j} = 0 \text{ if } \left(v_i, v_j\right) \in \varnothing,$$

where the subscripts i and j correspond to the rows and columns of the adjacency matrix, respectively.

2.2. Initialization: choosing initial population

Populations have the following main features:

- The initial population is formed as a set of individuals with randomly chosen vertices.
- The individual's size θ is fixed. (The number of graph vertices α in an individual is constant.)
- The population size P remains invariable during the algorithm. (The number of individuals in the population $P = (\theta_1, \theta_{2,\ldots,}\theta_k)$ is constant.)
- Each individual $\theta_d = \{V_\alpha\}$ is initialized by the vertex serial numbers randomly chosen using a uniform distribution on the set of nodes V.

2.3. Fitness estimation

The network is designed to supply resources to consumers. When choosing an individual and assessing its impact on the infrastructure, it is therefore reasonable to calculate the number of graph paths after isolating certain individuals.

The total number of paths between the vertices $\left(v_i, v_j\right) \in V$ with a length not exceeding m is calculated from the reachability matrix $E^* = \left(e^*_{i,j}\right)$. (Below such paths will be called paths of length m.) This matrix is the sum of adjacency matrices E raised to powers from 1 to m:

$$E^* = E + E^2 + E^3 + \ldots + E^m.$$

We denote by S the total number of paths of length m in the graph:

$$S = \sum_{i=1}^{n}\sum_{j=1}^{n} e^*\left(i,\ j\right),$$

where n is the total number of graph vertices.

We test the susceptibility of the infrastructure to the impact of a particular individual θ_d by isolating its vertices. For this purpose, the elements corresponding to these vertices in the adjacency matrix are set to 0. Then a new reachability matrix is calculated, and a new number of graph paths is calculated:

$$\tilde{S} = \sum_{i=1}^{n}\sum_{j=1}^{n} e^*\left(i,\ j\right),$$

where $\tilde{S}$ denotes the number of paths of length m in the graph after isolating all vertices of the individual θ_d.

Dividing the number of all graph paths by the number of all graph vertices, we obtain the average number of paths per one vertex:

$$S_{\text{avg}} = \frac{\tilde{S}}{n}.$$

Then the fitness function can be written as

$$F\left(\theta_d\right) = \min\left(S_{\text{avg}}\right).$$

Therefore, using the fitness function, we determine an individual θ_d with the following property: the average number of paths S_{avg} achieves minimum after isolating its vertices.

The operation of this algorithm can be demonstrated by an example. Consider the graph G of a resource supply network containing ten vertices, two sources $\left(v_1\ v_2\right) \in G$, and one consumer v_{10} (Fig. 3). For testing the individual's impact on the number of paths, we isolated the vertices $\left(v_5\ v_8\right) \in G$. For this purpose, we set to 0 the corresponding rows and columns of the adjacency matrix (Table 1) and calculated the number of paths in the reachability matrix.

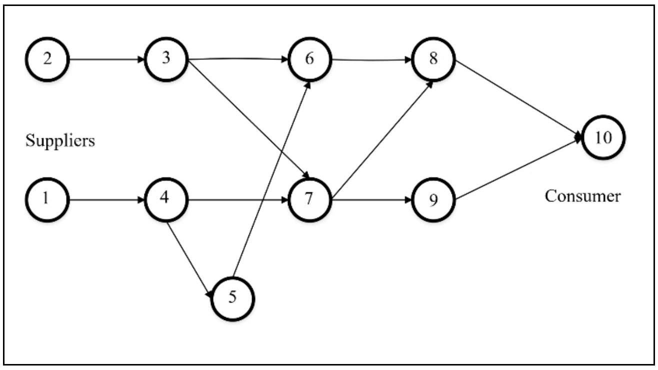

Fig. 3. Graph of a resource supply network.

The path length m was set to 3. Table 2 presents the total and average numbers of such paths (S and S_{avg}) in the original graph G and the modified graph after removing different pairs of vertices.

Clearly, the minimum average number of paths per vertex is obtained by isolating vertices v_6 and v_7. In this case, all paths between sources and the consumer are interrupted.

Table 1

Adjacency matrix after isolating vertices v_5 and v_8

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
v_1	0	0	0	1	0	0	0	0	0	0
v_2	0	0	1	0	0	0	0	0	0	0
v_3	0	0	0	0	0	1	1	0	0	0
v_4	0	0	0	0	0	0	1	0	0	0
v_5	0	0	0	0	0	0	0	0	0	0
v_6	0	0	0	0	0	1	0	0	0	0
v_7	0	0	0	0	0	0	0	0	1	0
v_8	0	0	0	0	0	0	0	0	0	0
v_9	0	0	0	0	0	0	0	0	0	1
v_{10}	0	0	0	0	0	0	0	0	0	0

Table 2

Number of paths of length 3 in graph G

Numbers of removed vertices	S	S_{avg}	Number of paths from sources to a consumer	Paths from sources to a consumer
–	45	4.5	6	1,4,7,9,10 1,4,7,8,10 1,4,5,6,8,10 2,3,6,8,10 2,3,7,8,10 2,3,7,9,10
5, 8	23	2.3	2	2,3,7,9,10 1,4,7,9,10
6, 7	11	1.1	0	–
3, 9	26	2.6	2	1,4,7,8,10 1,4,5,6,8,10

2.4. Selection, crossover, and mutation

The algorithm involves rank selection. With this method, after calculating the fitness values for crossover, $(l \times k)$ best individuals are selected, where l denotes the relative number of the best individuals in the population, and k is the population size. The parameter l describes the influence of selection on the survival of individuals in the population. In this paper, the value l ranges from 0.3 to 0.5.

Individuals selected with a given probability undergo single-point crossover (shuffling of binary strings). As a result, the offsprings receive half of the randomly determined characters from each parent. The offsprings form a new population of a given size k.

Mutation is necessary to prevent convergence to a local optimum. Since individuals are binary strings, mutation consists in inverting a randomly chosen gene for one randomly chosen individual. (Inverting means replacing 1 for 0 and vice versa.) The mutation frequency is set at the beginning of calculations, without any changes at subsequent stages.

For one generation, the search procedure stops according to the following criterion: the fitness function has the same value after selection and mutation. The resulting individual should be checked for the isolation of important consumer vertices from sources. For this purpose, the paths from sources to consumers are calculated using the Floyd–Warshall algorithm. If such paths exist, the genetic algorithm should be restarted to find a new solution for a new generation.

2.5. Application of genetic algorithm: example

The algorithm was tested on a graph of a real network segment; see Fig. 4.

The power system graph under consideration consists of 47 elements, including 15 sources and 5 consumers. We studied sextuple failures of elements.

Table 3 shows the cross-sections (sets of disconnected vertices) and the number of disconnected consumers. Clearly, the global optimum (disconnection of all consumers from the network) was achieved only in one case out of six. The time required to obtain one solution was approximately 3 s.

The sources and consumers were not analyzed (disconnected in the software implementation): such solutions are trivial and can be seen directly on the graph.

This approach can be extended to high-dimensional networks after representing in the form of graphs.

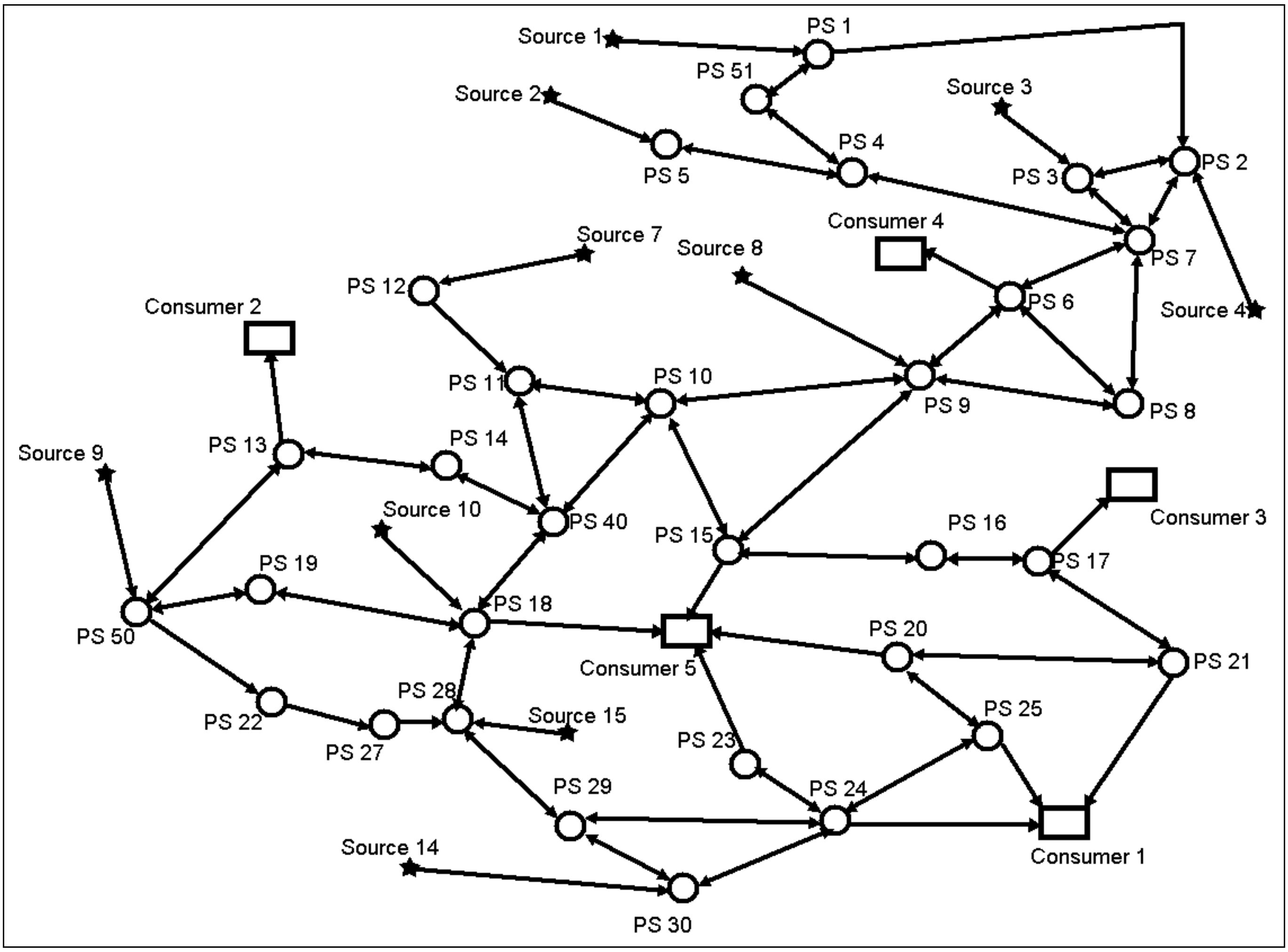

Fig. 4. Graph of a power system segment.

Table 3

Cross-sections in the graph of a power system segment: Calculation results

Disconnected vertices	PS 7, PS 10, PS 24, PS 18, PS 13, PS 9	PS 7, PS 15, PS 10, PS 13, PS 24, PS 18	PS 51, PS 22, PS 18, PS 10, PS 9, PS 24	PS 40, PS 15, PS 9, PS 7, PS 24, PS 13	PS 7, PS 9, PS 10, PS 18, PS 17, PS 24	PS 29, PS 9, PS 10, PS 18, PS 7, PS 24
Number of disconnected consumers	5	4	4	4	4	4

CONCLUSIONS

According to an analysis of publications on the subject, genetic algorithms are not applied to block the supply of resources to important consumers, despite their widespread use. At the same time, the disadvantages of traditional methods make it relevant to apply genetic algorithms for solving such problems.

The genetic algorithm proposed in this paper calculates the fitness function by estimating the average number of paths per one vertex after isolating individuals. Implementing the genetic algorithm to find cross-sections in infrastructures is not very difficult.

The proposed genetic algorithm needs significantly less computation time than the exhaustive search to determine multiple failures in high-dimensional networks.

REFERENCES

1. Bhamidipati, S., Van Der Lei, T.E., and Herder, P.M., A Layered Approach to Model Interconnected Infrastructure and Its Significance for Asset Management, *European Journal of*

Transport and Infrastructure Research, 2016, vol. 16, no. 1, pp. 254–272. DOI: 10.18757/Ejtir.2016.16.1.3130/
2. *GOST* (State Standard) R 53114-2008. *Information Protection. Ensuring Information Security in the Organization. Basic Terms and Definitions*, 2018.
3. Cagno, E., Grande, O., and Trucco, P., Towards an Integrated Vulnerability and Resilience Analysis for Underground Infrastructures, *Reliability Engineering & System Safety*, 2011, vol. 96, no. 1, pp. 139–148.
4. Valencia, V.V. and Maj, P.E., Network Interdependency Modeling for Risk Assessment on Built Infrastructure Systems, 2013. https://pdfs.semanticscholar.org/95ba/f36ae65157638a83f82084c39884b3f0fcb2.pdf?_ga=2.74081425.2003714931.1570034586-1895369083.1570034586.
5. Dorri, M.Kh., Roschin, A.A., and Sereda, L.A., Application of RDS Software Suite for Calculation and Visualization of Engineering Structures' Failure Consequences, *Automation in Industry*, 2017, no. 11, pp. 11–14. (In Russian.)
6. Grebenyuk, G.G. and Nikishov, S.M., Blocking of Energy and Resource Supply of Target Objects in Network Infrastructures, *Automation and Remote Control*, 2018, vol. 79, no. 3, pp. 535–544.
7. Ryabinin, I.A., *Nadezhnost' i bezopasnost' strukturno-slozhnykh sistem* (Reliability and Security of Structurally Complex Systems), St. Petersburg: St. Petersburg State University, 2007.
8. Haenni, R., Generating Diagnoses from Conflict Sets, *Proceedings of the 11th International FLAIRS Conference*, Sanibel Island, Florida, USA,1998. URL: www.aaai.org/Papers/FLAIRS/1998/FLAIRS98-081.pdf.
9. Arroyo, J.M. and Fernández, F.J., A Genetic Algorithm for Power System Vulnerability Analysis under Multiple Contingencies, in *Metaheuristics for Bi-level Optimization*, *Studies in Computational Intelligence*, 2013, vol. 482, pp. 41–68.
10. Arroyo, J.M. and Fernández, F.J., Application of a Genetic Algorithm to n-K Power System Security Assessment, *Electrical Power and Energy Systems*, 2013, vol. 49, pp. 114–121.
11. Genesi C., Granelli G., Marannino P., et al., A Genetic Continuation Power Flow Approach To Assess Power System Vulnerability, *Proceedings of the 43rd International Universities Power Engineering Conference*, Padua, Italy, 2008, pp. 1–5. https://ieeexplore.ieee.org/document/4651440.
12. Morcous, G. and Lounis, Z., Maintenance Optimization of Infrastructure Networks Using Genetic Algorithms, *Automation in Construction*, 2005, vol. 14, no. 1, pp. 129–142.
13. Permann, M.R., Toward Developing Genetic Algorithms to Aid in Critical Infrastructure Modeling, *Proceedings of 2007 IEEE Conference on Technologies for Homeland Security*, Woburn, MA, USA, 2007, pp. 192–197.
14. Agudelo, L., Munoz, N., and López-Lezama, J.M., Vulnerability Assessment of Power Systems to Intentional Attacks Using a Specialized Genetic Algorithm, *DYNA*, 2015, vol. 82, no. 192, pp. 78–84.
15. Kim, T., Wright, S.J., Bienstock, D., and Harnett, S., Vulnerability Analysis of Power Systems, *IEEE Transactions on Network Science and Engineering*, 2016, vol. 3, no. 3, pp. 132–146.
16. Karimi, E., Madani, S.M., and Ebrahimi, A., Power Transmission System Vulnerability Assessment Using Genetic Algorithm, *Intelligent Systems in Electrical Engineering*, 2012, vol. 3, no. 3, pp. 1–10.
17. Roselyna, J.P., Devarajb, D., and Dash, S.S. Multi-Objective Genetic Algorithm for Voltage Stability Enhancement Using Rescheduling and FACTS Devices, *Ain Shams Engineering Journal*, 2014, vol. 5, no. 3, pp. 789–801.
18. Teng, J.-H., Chen, C.-Y., and Martinez, I.C., Utilising Energy Storage Systems to Mitigate Power System Vulnerability, *IET Generation, Transmission and Distribution*, 2013, vol. 7, no. 7, pp. 790–798.
19. Permann, M.R., Genetic Algorithms for Agent-Based Infrastructure Interdependency Modeling and Analysis, *Proceedings of the 2007 Spring Simulation Multiconference* (*SpringSim* 2007), Norfolk, Virginia, USA, 2007, vol. 2, pp. 169–177.
20. URL: https://medium.com/swlh/genetic-algorithm-in-artificial-neural-network-5f5b9c9467d0.
21. URL: https://towardsdatascience.com/using-genetic-algorithms-to-train-neural-networks-b5ffe0d51321.
22. Anand, H. and Darayi, M., Power Network Component Vulnerability Analysis: A Machine Learning Approach, *Procedia Computer Science*, 2021, vol. 185, pp. 73–80.
23. Grebenuk, G.G., Dorri, M.Kh., Roshchin, A.A., and Nikishov, S.M., A Study of Influence of the Structure of Heterogeneous Engineering Networks on the Reliable Operation of Large Consumers, *Proceedings of the 11th IEEE International Conference on Application of Information and Communication Technologies* (*AICT*2017), Moscow: IEEE, 2017, vol. 2, pp. 57–61.
24. URL:https://progr-system.ru/wp-content/uploads/Math/МАПКС-15-ГрафыАтак.pdf.
25. URL:http://poivs.tsput.ru/ru/Math/DiscreteMath/GraphTheory/AccessibilityMatrix.

This paper was recommended for publication by I.B. Yadykin, a member of the Editorial Board.

Received May 17,2021, and revised September 1,2021. Accepted September 16,2021

Author information

Grebenyuk, Georgy Grigor'evich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ grebenuk@lab49.ru

Nikishov, Sergey Maksimovich. Cand. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ nikishov@lab49.ru

Sereda, Leonid Anatol'evich. Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ sereda@lab49.ru

Cite this article

Grebenyuk, G.G., Nikishov, S.M., Sereda, L.A. Vulnerability Analysis of Complex Network Infrastructures Using a Genetic Algorithm, *Control Sciences* **6,** 44–50 (2021).
http://doi.org/10.25728/cs.2021.6.5

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com