Large scale networks security strategy

Ya. Mostovoy[1], V.Berdnikov[1]

[1]*Samara National Research University, 34 Moskovskoe Shosse, 443086, Samara, Russia*

Abstract

The article deals with optimum two-phase planning of secure routs in large scale computer networks. Uncertainty of future needs is covered by extensive statistical modeling, which resulted in identification of statistical dependences and phenomena allowing for optimization of creation. To describe secure paths in random matrices the author uses programmable percolation apparatus. Tolerance of the created secure routes to failures in certain secure paths is demonstrated here.

Keywords: IT security; large scale networks; percolation; programmable percolation; two-phase operations

1. Introduction

Large scale (complex) networks are characterized by the large number of nodes, paths connecting them and mixed topology. There are a number of crucial research tasks pertaining to such networks, for example, analysis of dimensions and number of various-object clusters appearing in the networks; analysis of paths connecting nodes and clusters; analysis of nodes, removal of which may cause disintegration of the network into unlinked parts and etc.

The main task of the security strategy being a generalized long-term activity plan aimed at assuring security of large scale networks is effective use of limited resources.

In this case problem solving done in a responsive planning way basing on minimum aggregate expenditures allows succeeding.

Papers [10, 14, 17] tackle the issue of using the classical percolation theory for the applied research networks. However, the authors never address methods to study large scale networks based on programmable percolation theory explicated in [5, 6, 7, 8].

Habitually the percolation theory describes a grid of vertices and bonds or a square matrix of L lines, where the random number of cells is black allowing liquid, gas, traffic or data through, whereas the rest of cells are white or closed. If the concentration (probability of occurrence) of black cells increases, some of them randomly adjoin with the edges and merge. These black cells with adjoining edges make random open-bond clusters. These clusters appear and grow as the black cells concentration grows [1, 2].

The classical percolation theory describes randomly-filled matrix representing model of environment in direct geometrical interpretation [1, 3, 4, 9, 15, 16]. As it is, such an approach does not suit for large scale networks security analysis, because bonds among network nodes are diverse or even ill-defined, and though there are protective bonds among nodes they do not cover the majority of possible routes. It is necessary to migrate from network topology to that of the secure paths inter nodes matrix (SPNM). Such a matrix might ignite research of network security and availability of through paths using methods of the percolation theory.

2. Statement of the problem, method of analysis and computer experiment

A large scale network is considered here. Paths among certain nodes are secure. Resource scarcity makes it impossible to build all possible secure routes at once. One may make a secure route each time it is necessary, though it is time-consuming and costly, if compared to other routes incorporating available secure paths (or passing through clusters of such secure paths) provided the latter are abundant. In this case, to create the required route it is necessary to introduce few secure paths covering inter-cluster gaps.

Uncertainty in realization of the given secure route may be determined by statistical analysis based on a large number of random routes.

Thus, the network under consideration has a random number of securely connected nodes making up a somewhat stochastic secure basis. Now it is possible to build a completely secure route via any nodes of the network introducing additional secure segments where they are missing or necessary. As these additional secure segments are formed emergently, they require thorough positioning. Besides they are more expensive than secure paths from stochastic basis.

It is required to define probability of a secure path in the stochastic basis (in terms of the percolation theory – concentration of the open black cells) which minimizes overheads of building secure routs in the network.

In the classic percolation theory [1, 2, 12, 15, 16] they define K_{th} – the concentration of the open black cells or stochastic percolation threshold, when a random route passing through black cells from top to bottom of the matrix in the given direction, i.e. stochastic percolation cluster, appears. However, this stochastic percolation cluster has loose structure, considerable number of dead brunches and is obviously redundant for real-world application.

With the programmable percolation [5, 6, 7, 8] at the first stage there is built a basis consisting of randomly distributed secure paths making clusters and having concentration well below the stochastic percolation threshold. At the second stage by inserting additional secure paths into existing inter-cluster gaps there is created a through percolation route. Here concentration of the stochastic basis is chosen to make cumulative cost of the two-phase operation minimal. Solving of this problem shows that programmable percolation allows having concentration of objects (secure paths) more than twice as little as the stochastic percolation threshold. As well the concentration of objects is in the neighborhood of concentration typical to average maximal number of clusters appearing ($K = 0.25$).

As far as targets of research are large scale networks and statistical phenomena of secure-path clusters, and the goal of research is long-term planning of optimal-cost secure routes in large scale networks, our theoretical considerations are verified by a computer experiment - the only possible way of application investigation.

The computer experiment for long-term planning of secure routs in large scale networks consisted of a number of consecutive stages, repeated for each of randomly filled matrices (SPNM) being models of operation environment.

Each time the following steps were made for each value of secure routs concentration:

- the matrix was randomly filled with objects in conformity with the predetermined probability law and concentration value;
- the resultant clusters and objects were identified and analyzed;
- measures of cluster distribution (average values, scatter and etc.), cluster size, inter-cluster gaps and etc. were calculated;
- gaps between stochastically-formed clusters were analyzed; shortest artificial percolation paths were formed; average length of the above mentioned path was measured and average number of additionally inserted secure segments covering inter-cluster gaps was calculated per totality of randomly-filled matrices.

In order to identify clusters and estimate their characteristics we used the Hoshen-Kopelman algorithm [11, 13]. To make paths through clusters we created a Lightning – Closest Point algorithm which is an adaptation of Lighting strike and Dijkstra's algorithms [5, 6, 7, 8].

3. SPNM properties

Classic percolation theory considers a randomly-filled matrix to be a model of environment in direct geometrical interpretation. Such an approach does not suit for analysis of network security, because network topology cannot be rendered by a two-dimensional array. It is necessary to migrate from network topology to that of the secure paths inter nodes matrix (SPNM). Such a matrix might ignite research of network security and availability of through paths using methods of the percolation theory.

Example of such a transmission is demonstrated below in Figures 1 and 2.

Black lines are data connections (paths) between network nodes. Yellow lines are secure connection (secure paths) between the nodes.

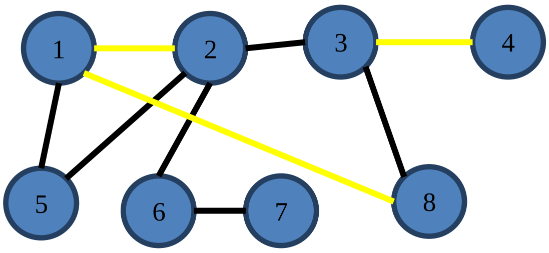


Fig. 1. Random network graph.

	1	3	6
2	■		
8	■		
5			
7			
4		■	

Fig. 2. SPNM for secure sections of the network? Demonstrated in Figure 1.

SPNM is filled according to the following rule: end-node names of interest are recorded in the vertical direction, start-node names of interest are recorded in the horizontal direction (in Figure 2 they are blue). **Note that nodes in the vertical and horizontal directions are not repeated.** The suggested research tool SPNM strict squareness is unimportant. Casual randomization of lines and columns is possible. SPNM filling algorithm is the following:

```
1. Repeat unless all nodes are done:
    1.1. If node $A$ is missing in the table, record the node name in the horizontal direction.
    1.2. Record nodes, which are connected with the node $A$ in the network in vertical direction.
    1.3. Mark SPNM cells correspondingly: black if there is secure connection between the node $A$ and other nodes from the table.
2. End of the loop.
```

Adjoining black cells make up a cluster. The point is that information can be securely transferred via this segment of the network. In the given example (see Fig.2) there are two clusters: the «1-2, 1-8» cluster and the «3-4» cluster.

If certain inter-cluster gaps in the SPNM are filled with secure connections (marked red), then a through non-stochastic but programmed percolation route is created in the SPMN. It means that all the nodes recorded in the vertical direction are available for secure connection with the nodes recorded in the horizontal direction.

Thus, secure interconnection of all the nodes recorded in the vertical direction is rendered on the SPNM as a programmable percolation vertical route (see Fig.3):

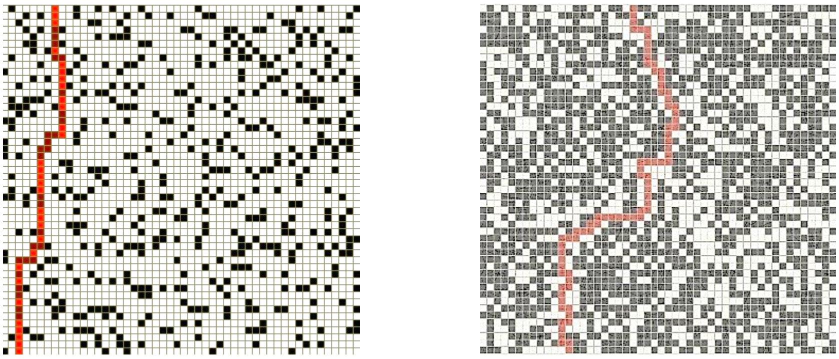
	1	3	6
2	⬛		
8	⬛		
5	🟥	🟥	
7		🟥	
4		⬛	

Fig. 3. Programmable percolation in SPNM.

It is obvious, that the number of such secure paths might be great. They might pass through one or several nodes located on the horizontal axis. There might be other percolation routes generated with directed percolation. To plot the shortest route in the given direction we used an adaptation of Dijkstra's algorithm. All programmable percolation routes are the subject of statistical modeling.

For statistical analysis we used different-size SPNM filled with the help of the random number generator.

Example of an SPNM randomly filled with secure paths (black cells) is given in Figure 4. Concentration of the black cells differs. SPNM size here is 50×50. Possible shortest routes of the programmable percolation in the bottom-top direction across the SPNM are plotted in red. Note greater tortuousness of the programmable percolation route across the matrix with $K = 0.6$ concentration.



a) b)

Fig. 4. Examples of percolation routes across matrices with a) $K = 0.25$, b) $K = 0.6$ population concentration.

4. Some statistical peculiarities of clusters’ formation in large scale networks

Concentration K is a relative fraction of black nodes during random and homogeneous filling of the matrix. It makes black cells [2] likely to appear, when probability of their occurrence in the matrix is uniformly distributed. That is why here and elsewhere we use both: the expression “probability of the predefined object (secure path) in the matrix cell” and its epitomized version - “concentration”.

Statistical modeling using square randomly filled matrices allows detection and analysis of cluster statistical phenomena (peculiarities) being of great practical consequence.

The first peculiarity is presence of the stochastic percolation threshold in the shape of matrix dissection by the open percolation cluster. It is guaranteed at $K = 0.6$.

The second peculiarity is such concentration of objects when average number of clusters is maximum [5, 6, 7, 8, 18]. A ibid is demonstrated that the value is robust, i.e. low responsive to the object presence in the matrix cell probability distribution law. This peculiarity manifests itself at $K = 0.25$ (see Fig. 5).

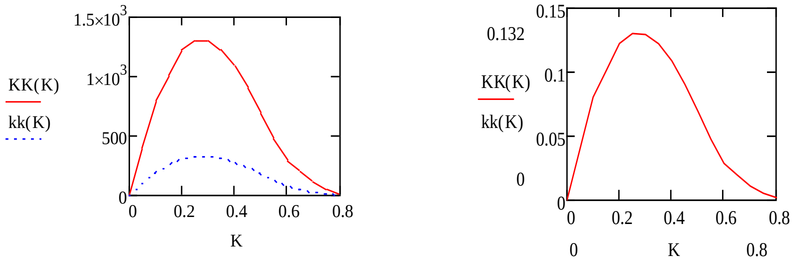
The third statistical peculiarity is maximum average length of the shortest route through the stochastically formed clusters in the percolation direction. This value appears when the population of objects and route tortuousness grows. Average length of the programmable percolation $L(K)$ shortest route grows up to the stochastic percolation threshold, and upon reaching it starts decreasing. The more tortuous is the percolation route (i.e. the longer it is), the more passing clusters it incorporates.

5. Analysis of two-phase operations

During statistical modeling we considered several thousands of different size matrices. The cells of those matrices were randomly filled with provision for equal probability of objects distribution in the cells. In order to identify all the clusters in the received random matrices we used the Hoshen-Kopelman algorithm [5, 6, 7, 13]. Then we estimated their statistical characteristics and plotted curves of average values.

Dependence of the average number of clusters in the matrix from the probability of the object in the cell K is given in Figure 5. When the probability increases up to ~ 0.25, the matrix is being filled with the objects, and the number of cluster grows. Further growth of concentration results in merging of the clusters. Their average number decreases while their size grows.

On several physical grounds we established that the number of clusters appeared in the matrix with the certain concentration depended on the matrix area size L^2, while length of routes depended on the linear dimension of the matrix L. Consequently, it is possible to save numerical results of statistical modeling from influence of the matrix size by dividing them by L or L^2 correspondingly. Numerical computations verify the above said (see Fig.5).



a) b)

Fig. 5. Dependence of the average number of clusters on the object probability in the cell for a 50x50 matrix (in dots) and a 100x100 matrix – a) average number of clusters normalized by the matrix area size – b) for both cases.

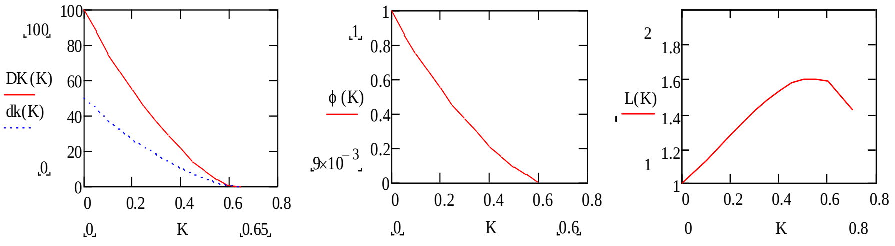
We may decrease the appropriate concentration and consequently number of objects necessary for percolation, if we replace classical stochastic percolation with the suggested programmable percolation and apply the two-phase approach.

Taking into account different value of type I objects randomly distributed to form a stochastic basis (black cells) and type II objects inserted in certain places of the coverage area to get the shortest programmable percolation route (red cells), we are able to come at a such concentration of the stochastic basis when total cost of the created programmable percolation route is minimal.

Having said this it can be believed that each of the objects from the stochastic basis scattered in the operating environment is cheaper than an additional object inserted into a certain place of the same operating environment.

Figure 6 demonstrates processed results of two-phase operations computer experiment: average number of the inserted objects necessary for programmable percolation with various concentrations of objects in the stochastic basis and for different-size matrices. Figure 6a: in vertical direction is given the average number of the objects inserted in 50×50 matrix (dotted line) and 100×100 matrix. Figure 6b: the dependences are normalized according to the matrix size (whereupon the graphs coincided).

Stochastic percolation cluster is formed at concentration $K = 0.6$ and the shortest percolation route passes through it. That is why in this case the average number of the added cells tends to zero. At this concentration tortuousness and length of the percolation route are maximal. Further growth of the concentration makes the shortest percolation route more straight and its length decreases (Fig. 6c) [8].



a) b) c)

Fig. 6. Dependency of the average number of inserted objects $\varphi(K)$ and the average normalized length of the programmable percolation route (K) from the probability of the object in the cell K.

Let us calculate cost of the two-phase operation. The cost of finding (preparation) of each random secure path is designated as α, the cost of a single additional secure path selected (prepared) in a certain place of a large scale network during the second phase is designated as $\theta(K)$.

Then the total cost of the two-phase operation P is:

$$\text{P} = \alpha * K * L^2 + \theta(\text{K}) * \varphi(\text{K}) * L \qquad (1)$$

Where the first term is the cost of preparation of the operating environment stochastic basis, $K * L^2$ – number of basic secure paths in the stochastic basis expressed as concentration function. The addend in (1) is the cost of the secure paths necessary to form the shortest programmable percolation route through stochastically generated clusters. $\varphi(K) * L$ is average number of the added secure paths in SPNM of L size specified by stochastic computer experimental results and demonstrated in the normalized dependence (see Fig. 6c).

$\theta(K)$ function reflects cost of each secure path created and inserted in the large scale network variation versus stochastic basis concentration.

We assume that the cost of each additional secure path created in a certain SPNM cell is proportional to the size and number of inter-cluster gaps covered along the percolation route. In other words it is proportional to the number of the reds in the route $\varphi(K) * L \big/ L(K) * L$ and inversely proportional to relative tortuousness of the route $L(K) \big/ L$ as maximal tortuousness means absence of gaps to be covered (absence of the reds). Therefore, wealth of gaps in the route results in greater cost of each additional secure path, $\theta\,(K) = \theta_0 * \varphi(K) * L \big/ L(K)^2$. With account of this equation, the cumulative costs formula (1) shall be written as:

$$\mathrm{P} = \alpha \times K \times L^2 + \theta_0 * \varphi(K)^2 * L^2 \big/ L(K)^2 \tag{2}$$

Let us analyze relative cost of a two-phase operation. For this we divide the left-hand side and the right-hand side of the obtained equation (2) by $\mathrm{P}_{\text{п}} = \alpha * K_{\text{п}} * L^2$ – cost of a purely stochastic one-phase operation.

Then:

$$\mathrm{P}_{\text{отн}} = \mathrm{P} \big/ \mathrm{P}_{\text{п}} = 1.7K + 1.7 * \left((\theta_0 * \varphi(K)^2) \big/ (\alpha * L(K)) \right) = 1.7\left(K + R * \varphi(K)^2 \big/ L(K)^2 \right), \tag{3}$$

where $R = \theta_0 \big/ \alpha$ is ratio of the additional object cost to the stochastic basis object cost. Figure 7 demonstrates two-phase operation relative cost variation versus stochastic basis K obtained with the above equation taking into consideration $\varphi(K)$ and $L(K)$ variations (see Fig. 5) for $R = 1$.

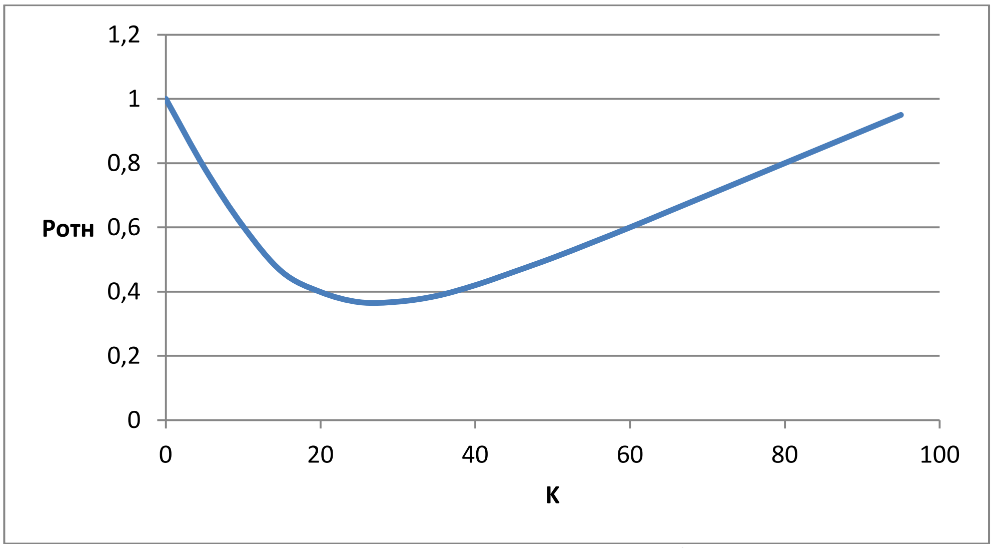


Fig. 7. Two-phase operation relative cost variation versus concentration of objects in the stochastic basis.

The plot in Figure 7 demonstrates that from the perspective of two-phase operations total cost minimization, optimal probability of a secure path in the stochastic basis cell shall be ~ 0.25, which corresponds to the maximal number of clusters in the stochastic basis (see Fig. 5). This remarkable point does not explicitly occur in the equations used for plotting of the graph (see Fig.7). The result may be interpreted as validation of statistic computer experimental data and model of two-phase operation labor consumption.

6. Percolation route stability analysis

Complex technical systems rarely work as expected. But smart security strategy shall consider off-design operation and available redundancy. This is the only way to overcome failures and errors.

For this reason large scale network security model shall support safe operation of the network even in case of node faults. Statistical analyses of SPNM suggested in the article enables pointedly handle the issue.

We studied failures of secure bonds along a percolation route. Here failure means that protection in the SPNM cell is destroyed and it results in interruption of secure information stream along a chosen route. Supposing that, it is impossible to promptly recover the fault point, but to bypass. The question is: what is the cost of such a bypass, how much is the route lengthened? It is evident that the answer depends on the large scale network topology or on the concentration of the blacks if we consider our percolation model.

So, to answer the question we performed statistical computer experiment. Computer-aided experiment was conducted in the following way: first of all we randomly chose a cell on the percolation route and put a veto on the route passing through it.

Then, we let a new percolation route start from the preceding point in the same direction and on the same conditions of optimality bypassing the fault. Probability that the new percolation route reverts to the former one is demonstrated in Figure 8; optimal concentration of the stochastic basis $K = 0.25$ was estimated in advance. The plot obtained in the cause of the experiment did not depend on the matrix size. The plot was normalized according to the following rule: $L_n = {}^{i}/_{L}$, where i is coordinate position of the failure in SPNM in the vertical direction, L is the SPNM height.

It is safe to say that the route is invariable up to $L_n = 0.85$, i.e. a new percolation route is likely to revert to the former one. Al the upsurges seen on the plot are within statistical error.

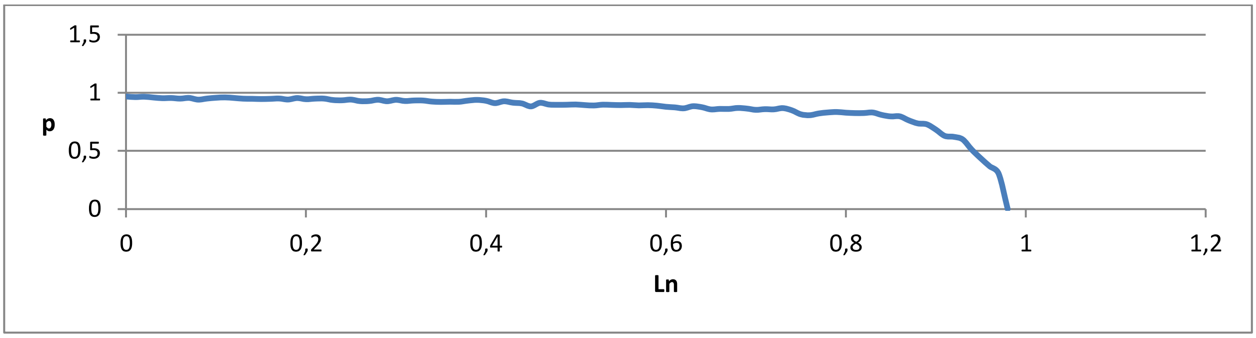


Fig. 8. Probability of a new route to revere to the formed one from the failure point.

As a part of the study were obtained variations of the green-cell number versus concentration for matrices of different L sizes. From now on green cells are the cells newly added to the percolation route with the purpose to bypass a banned cell denoting a failure.

Variation of the green cells number L_G versus concentration is given in Figure 9. Actually, the plot for matrices of different sizes is the same.

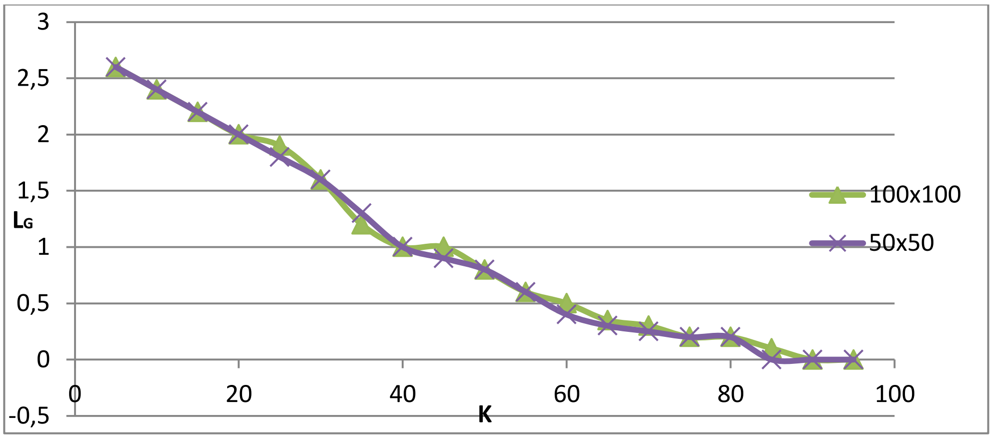


Fig. 9. Variation of the number of green cells added to a bypass route versus concentration for matrices of different sizes.

Note that all the deviations of the plot are within statistical error.

Based on the findings it follows that the number of the green cells independent from the matrix size, but depends on the concentration. Therefore, the fault phenomenon is of purely local nature.

7. Analysis of programmable percolation in SPNM

Besides, we studied programmable percolation, i.e. making routes from the given point to the target point. This problem might be highly topical for information networks outside statistical research, when, for example, it is required to establish secure communication between some given nodes of the network.

Let us locate point A anywhere in the first row of the matrix and point B – anywhere in the last row of the matrix. Percolation route created for points A and B goes at some angle, further on referred to as “angular displacement” relative to the matrix vertical line. In other words, this route goes along some centerline between points A and B. To avoid variation versus matrix size we shall normalize to the length of the guiding axis in the following way:

$$\varphi_T = \frac{L_R}{l}; l = \sqrt{(i_B - i_A)^2 + (j_B - j_A)^2}, \tag{4}$$

where L_R – the number of red cells added to the percolation route, l – geometric distance between points A and B calculated by the Pythagorean theorem, where $(i_A; j_A)$ и $(i_B; j_B)$ are coordinates of points A and B correspondingly.

Upon thorough study of various angular displacements it was found out that value φ_T was independent from the angular displacement. Then we constructed variation of the value φ_T versus concentration according to the following rule: points A and B should be located inside the clusters. We juxtaposed the obtained plot (see Fig. 10, plot b) with the plot in Figure 6b (see Fig.

10, plot a). The results agreed. Hence, the average number of objects (red cells) added to the percolation route would be the same irrespective to the direction of a percolation route.

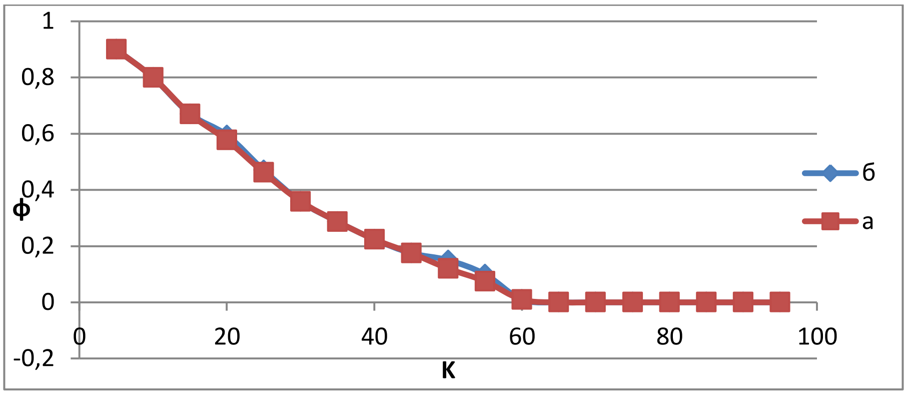


Fig. 10. a – variation of the average number of added objects $\varphi(K)$ and average normalized length of a programmable percolation route versus concentration (Fig. 9); b – variation of the normalized length of a programmable percolation route versus concentration.

Note that graph 10a was plotted by averaging the number of the red cells in the situation of the percolation failure, whereas graph 10b was plotted by averaging the number of the red cells for programmable percolation between target points A and B.

8. Conclusion

1. In case of limited resources cost-effective planning of secure routes shall be two-phased: firstly is created a stochastic basis of secure though rather low-concentrated paths, and secondly are built secure routes via clusters of the stochastic basis with minimal insertion of secure paths in between the gaps of the clusters.

2. Concentration of secure paths in the stochastic basis shall be 0.25. At such concentration of secure nodes the number of the generated clusters is maximal. In this case any secure route built between the given nodes of the network has minimal average total cost.

3. Subsequent to the results of the percolation route stability analysis it was found that the fault phenomenon was of purely local nature and bypass routes were likely to revert to the original percolation route.

4. When the optimal concentration of secure paths is 0.25 , the average number of additionally inserted secure paths to bypass the failed one is not more than 2.

References

[1] Moskalev P, Shitov V. Porous structures computer experiment. Moscow: Fismatlit, 2007; 120 p. (in Russian)

[2] Percolation: theory, application, algorithms: Reference Book. Edited by Tarasevich YuYu. Moscow: Editorial URSS, 2002; 109 p. (in Russian)

[3] Golubev AS, Zvyagin MYu, Milovanov D. Percolation effect in information networks with unstable links. Bulletin of Lobachevsky State University of Nizhni Novgorod 2011; 2(3): 260–263.

[4] Nekrasova AA Sokolov SS. Study of the possibility of percolation theory for flow control in information networks transport. Bulletin of Admiral Makarov State University of Maritime and Inland Shipping 2010; 32(4): 192–198.

[5] Mostovoi YaA. Statistical phenomena in large-scale distributed clusters of nanosatellites. Vestnik of Samara University. Aerospace and Mechanical Engineering 2011; 26(2): 80–89.

[6] Mostovoi YaA. Two-phase operation in large-scale networks of nanosatellites. Computer Optics 2013; 37(1): 120–130.

[7] Mostovoi YaA. Programmable percolation and optimal two-phase operations in large-scale networks of nanosatellites. Infokommunikacionnye Tehnologii 2013; 11(1): 53–62.

[8] Mostovoi YaA. Simulation of optimal two-phase operations in random operating environments. Avtometriya 2015; 51(3): 35–41.

[9] Alexandrowicz Z. Critically branched chains and percolation clusters. Physics Letters A 1980; 80(4): 284–286.

[10] Agrawal P, Redner S, Reynolds PJ, Stanley HE. Site-bond percolation: a low-density series study of the uncorrelated limit. J. Phys. A: Math. Gen. 1979; 12: 2073–2085.

[11] Babalievski F. Cluster counting: the Hoshen-Kopelman algorthm vs. Spanning three approach. International Journal of Modern Physics 1998; 9(1): 43–61.

[12] Galam S, Mauger A. Universal formulas for percolation thresholds. Phys. Rev. E 1996; 53(3): 2177–2181.

[13] Hoshen J, Kopelman R. Phys. Rev. B 1976; 14: 3438–3445.

[14] Sarshar N, Boykin PO, Roychowdhury VP. Scalable Percolation Search in Power Law Networks. Proceedings of the Fourth International Conference on Peer-to-Peer Computing. Zurich, 2004.

[15] Stauffer D. Scaling theory of percolation clusters. Physics Reports 1979; 54: 1–74.

[16] Stauffer D, Aharony A. Introduction to Percolation Theory. London: Taylor & Francis, 1992.

[17] Vakulya G, Simon G. Energy Efficient Percolation-Driven Flood Routing for Large-Scale Sensor Networks. Proceedings of the International Multiconference on Computer Science and Information Technology. Wisla, Poland, 2008; 877–883.

[18] Wilkinson D, Willemsen JF. Invasion percolation: A new form of percolation theory. J. Phys. A 1983; 16: 3365–3376.