A Genetic Approach for Virtual Computer Network Design

Igor Saenko and Igor Kotenko

Abstract. One of possible levels of computer protection may consist in splitting computer networks into logical chunks that are known as virtual computer networks or virtual subnets. The paper considers a novel approach to determine virtual subnets that is based on the given matrix of logic connectivity of computers. The paper shows that the problem considered is related to one of the forms of Boolean Matrix Factorization. It formulates the virtual subnet design task and proposes genetic algorithms as a means to solve it. Basic improvements proposed in the paper are using trivial solutions to generate an initial population, taking into account in the fitness function the criterion of minimum number of virtual subnets, and using columns of the connectivity matrix as genes of chromosomes. Experimental results show the proposed genetic algorithm has high effectiveness.

Keywords: data mining, genetic algorithms, VLAN, Boolean Matrix Factorization.

1 Introduction

Ensuring computer security is a very important task. The specific security threat may come from internal users (insiders). In many cases, the insiders' behavior prediction is a complex task. At the same time, a damage to computer security, which may be caused by unauthorized insiders' actions (malicious or deliberate) can be huge. Therefore, implementation of an additional level of computer security,

Igor Saenko
St.Petersburg Institute for Information and Automation of the Russian Academy of Sciences,
14-th line, 39, St.Petersburg, 199178, Russia
e-mail: ibsaen@comsec.spb.ru

Igor Kotenko
St.Petersburg Institute for Information and Automation of the Russian Academy of Sciences,
14-th line, 39, St.Petersburg, 199178, Russia
St. Petersburg National Research University of Information Technologies, Mechanics and Optics,
49, Kronverkskiy prospekt, St.Petersburg, Russia
e-mail: ivkote@comsec.spb.ru

© Springer International Publishing Switzerland 2015

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_11

designed to prevent unauthorized access of insiders to nodes of computer networks, is very desirable in all cases.

One of possible levels of computer protection against insiders is splitting computer networks into logical fragments that are known as *virtual computer networks* or, for simplicity, *virtual subnets*. This splitting is carried out through various means of VLAN (Virtual Local Area Network) technology [1]. These tools ensure that if two computers do not belong to the same virtual subnet, then the exchange of information between them is impossible. Using VLAN technology to differentiate information flows in the network should be very convenient for security administrators, because all actions necessary for that purpose can be localized in one place, for example, on an Ethernet switch or a router. An alternative way to differentiate information flows, which is associated with the use of access lists in operating systems for workstations in the computer network, is also possible, but it is less flexible. If it is needed to re-configure the information flow schema in this case, security administrators must access the remote workstations that may be not possible.

The task to create an additional level of computer protection against insider attacks based on virtual subnets is insufficiently discussed in the scientific literature. Usually, design of virtual subnets according to the VLAN technology is based on factors, not related to security. It is now largely accepted to form virtual subnets based on functional grounds. Virtual subnets are usually formed around servers giving access only permitted sets of computers. However, according to this approach the exchange of legitimate information between computers belonging to different subnets can be impossible. The paper investigates the possibility of building an additional level of security to ensure the required differentiation of information flows between computers based on the VLAN technology. The mathematical foundations of this problem are examined, showing that it is one of the varieties of Boolean Matrix Factorization (BMF). As the BMF methods can be considered as Data Mining methods and are applied to solve NP-complete problems, we believe this problem has the same computational complexity. The paper proposes to use genetic algorithms for solving this problem. Genetic algorithms have proved to be successful in solving optimization problems of different complexity. Some examples of their successful application in optimization problems, which deal with Boolean Matrixes decomposition, are known. However, the paper shows that it is not possible to apply directly known implementations of genetic algorithms to solve this problem.

The *main theoretical contribution* of the paper consists in the following: the first time we formulate the problem of virtual subnets design based on logical connectivity matrix; we show that this problem is one of the BMF forms; for solving the problem we propose an improved genetic algorithm. *The basic algorithm improvements* are as follows: using trivial solutions to generate an initial population; taking into account in the fitness function the criterion of minimum number of virtual subnets; using columns of the logical connectivity matrix as the genes of chromosomes.

The rest of the paper is structured as follows. Section 2 provides an overview of related work. Section 3 deals with mathematical foundations. The genetic algorithm to solve the task is described in section 4. Section 5 discusses the experimental results. Section 6 is a conclusion.

2 Related Work

In order to determine the scope of the problem and to form the mathematical basis for its decision, the works related to the decomposition of Boolean matrices were primarily analyzed. P. Miettinen et al. [2, 3, 4] considered some tasks of Matrix Factorization, which include Non-Negative Matrix Factorization (NMF) and BMF. They proved that NMF and BMF problems are NP-complete. Mathematical methods to solve them were suggested for these tasks. H. Lu et al. [5, 6] showed that some of the problems in the field of information security can be summarized by BMF. In particular, they demonstrated that the task of determining roles for role-based access model, known as Role Mining Problem (RMP), is one of these tasks.

These ideas were further elaborated in [7, 8], where the genetic algorithms to solve the RMP were proposed. In [7] a multi-chromosomal approach was suggested for coding solutions. According to this approach, each individual of genetic algorithm population has three chromosomes. One of these chromosomes is for controlling. It helps to perform crossover, when parent chromosomes have different length. In [8] a multi-chromosomal coding was denied and a stage of pre-processing was proposed for crossover. However, both these innovations could not be applied for designing virtual subnets, as the problem, which is discussed in the paper, requires determining only one Boolean matrix instead of two matrices. For this reason, the pre-processing stage, which is proposed in [8], should be considered as redundant. As it will be shown below, the problem does not require chromosomes with different lengths. Moreover, we propose an approach to calculate the length of chromosomes and to use so-called trivial solutions for generating initial population.

High complexity of the NMF and BMF problems and the ability to find effective mathematical methods to solve them only for special cases identified the need to find special heuristics. A. Janecek et al. [9] suggested using bio-inspired algorithms based on populations to meet the challenges of the NMF. They discussed several possible algorithms which can be used: genetic algorithms, particle swarm optimization, differential evolution, fish school search and fireworks algorithms between. Comparative evaluation of these algorithms showed that genetic algorithms have the greatest efficiency in solving NMF problems.

V. Snasel et al. [10, 11] proposed using genetic algorithms for solving BMF problem. These algorithms were designed to find the Boolean matrices $\mathbf{W}$ and $\mathbf{H}$, on which the given matrix $\mathbf{A}$ is decomposed. In this algorithm, rows of the matrix $\mathbf{H}$ are used as the genes of chromosomes and the crossover operator is applied to the columns of the matrix $\mathbf{W}$. The fitness function was formed on the base of Euclidean distance between the source and resulting matrices. In addition, this algorithm cannot strictly provide the equality between $\mathbf{W}$ and $\mathbf{H}^{\mathrm{T}}$, where $\mathbf{H}^{\mathrm{T}}$ is a transposed matrix $\mathbf{H}$. Due these features, the algorithm that was proposed in these works cannot be applied to our problem.

There is a few works on application of Data Mining to create virtual subnets. Ch.-F.Tai et al. [12] proposed to form a schema of virtual subnets by cluster analysis. This approach is oriented to use in mobile ad hoc networks. However, this approach is less efficient than genetic algorithms for large networks. The genetic algorithm for

optimization of virtual network schemes was proposed in [13]. But this algorithm allows forming only matrix $\mathbf{A}$ and does not allow finding matrices $\mathbf{X}$ and $\mathbf{X}^{\mathrm{T}}$, on which matrix $\mathbf{A}$ should be decomposed. In addition, this approach does not take into account the criterion of minimizing the total number of virtual subnets.

Thus, the analysis of relevant work has shown that genetic algorithms should be considered as sufficiently effective to solve the problem discussed. At the same time, genetic algorithms, which are proposed in known works, cannot be directly used for this purpose.

3 Mathematical Background

Suppose that there are n computers in a network. The diagram of allowed flows between these computers is defined by Boolean matrix $\mathbf{A}[n, n]$. If $a_{ij} = 1(i, j = 1, \ldots, n)$ then the exchange between computers i and j is allowed. Otherwise, this exchange is not possible.

Suppose that we have formed k virtual subnets in the network. Each of these subnets combines two or more computers. Let us set the distribution of computers on subnets by using the matrix $\mathbf{X}[n, k]$. If $x_{ij} = 1(j = 1, \ldots, k)$ then the computer i belongs to the subnet j. Otherwise subnet j does not include computer i.

The matrix $\mathbf{A}$ plays a role of initial data. The matrix $\mathbf{X}$ is the result of solving the problem. We will show that between Boolean matrices $\mathbf{A}$ and $\mathbf{X}$ there is the following relationship:

$$\mathbf{A} = \mathbf{X} \otimes \mathbf{X}^{\mathrm{T}} \tag{1}$$

where $\mathbf{X}^{\mathrm{T}}$ is a transposed matrix $\mathbf{X}$, symbol $\otimes$ stands for Boolean matrix multiplication, which is a form of matrix multiplication based on the rules of Boolean algebra. Boolean matrix multiplication allows getting the elements of matrix $\mathbf{A}$ by the following expression: $a_{ij} = \vee_{j=1}^{n}(x_{ij} \wedge x_{ji})$.

Consider the following example. Let $n = 5, k = 3$. There are virtual subnets in the computer network as shown in Fig. 1. Subnet 1 includes workstations WS1, WS2 and WS4, subnet 2 — WS2, WS3 and WS5, subnet 3 — WS1, WS4 and WS5.

In this case, the relationships between the matrices $\mathbf{A}$ and $\mathbf{X}$ are as follows:

$$\mathbf{X} = \begin{pmatrix} 1\,0\,1 \\ 1\,1\,0 \\ 0\,1\,0 \\ 1\,0\,1 \\ 0\,1\,1 \end{pmatrix}, \mathbf{X}^{\mathrm{T}} = \begin{pmatrix} 1\,1\,0\,1\,0 \\ 0\,1\,1\,0\,1 \\ 1\,0\,0\,1\,1 \end{pmatrix}, \mathbf{A} = \begin{pmatrix} 1\,1\,0\,1\,1 \\ 1\,1\,1\,1\,1 \\ 0\,1\,1\,0\,1 \\ 1\,1\,0\,1\,1 \\ 1\,1\,1\,1\,1 \end{pmatrix}.$$

It is easy to see that the matrix $\mathbf{A}$ is symmetric.

Now we will show that this problem is a kind of BMF problems. The BMF problem is to find Boolean matrices $\mathbf{W}$ and $\mathbf{H}$ that are related with given Boolean matrix $\mathbf{A}$ by the following equation:

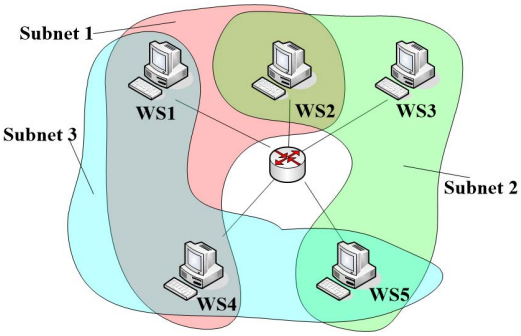

Fig. 1 Distribution of workstations over virtual subnets (example)

$$\mathbf{A} = \mathbf{W} \otimes \mathbf{H}, \tag{2}$$

where $\mathbf{A} = \mathbf{A}[n, m]$, $\mathbf{W} = \mathbf{W}[n, k]$ and $\mathbf{H} = \mathbf{H}[k, m]$.

Comparing (1) and (2), we can notice that the equation (1) can be seen as a special case of the equation (2) when the next two conditions are met. The first condition is the equality $m = n$. The second condition takes the following form:

$$w_{ij} = h_{ji} \text{ for any } i = 1, \ldots, n \text{ and } j = 1, \ldots, k. \tag{3}$$

From the fact that the problem discussed is a variant of BMF problems, it follows that the problem is NP-complete. However, because of conditions (3), the direct application of the BMF methods for solving the problem is difficult. Hence our proposal is to solve this problem by heuristic methods, namely, genetic algorithms.

However, for this purpose, the optimization criterion should be defined. Let us pay attention to the fact that in the task (1) the value k can be arbitrary. Now we show how we can restrict it from above. For this purpose we will split the network on subnets, where each subnet includes only two computers i and j when $a_{ij} = 1$.

For the example shown in Fig. 1 the matrix $\mathbf{X}$ has five rows and eight columns:

$$\mathbf{X} = \begin{pmatrix} 1\,1\,1\,0\,0\,0\,0\,0 \\ 1\,0\,0\,1\,1\,1\,0\,0 \\ 0\,0\,0\,1\,0\,0\,1\,0 \\ 0\,1\,0\,0\,1\,0\,0\,1 \\ 0\,0\,1\,0\,0\,1\,1\,1 \end{pmatrix}.$$

The matrix $\mathbf{X}^{\mathrm{T}}$, respectively, has eight rows and five columns. However, Boolean multiplication of these matrices still results in a matrix $\mathbf{A}$ (it is not hard to check). The order of the columns can be arbitrary.

We will call such a matrix $\mathbf{X}$ as *the trivial solution*. A trivial solution has the following properties. First, the number of columns in the matrix $\mathbf{X}$ is equal to the number of '1'-values in the matrix $\mathbf{A}$ located above the main diagonal.

We denote this value as M. Secondly, in each column of the matrix $\mathbf{X}$ there is just two '1'-values. Finally, a number of trivial solutions is equal to the number P of permutations of columns in the matrix $\mathbf{X}$. It is obvious that $P = 2^M$. For this reason the trivial solutions cannot be considered as good. At the same time, we see that for the example shown in Fig. 1 there is a good solution, in which $k = 3$. Therefore, we propose the *optimization criterion* of the problem as the *minimum value of k under full coincidence* of matrices $\mathbf{A}$ and $\mathbf{X} \otimes \mathbf{X}^{\mathrm{T}}$. The smaller k, the less the complexity of administrative work and higher the network security.

4 Genetic Algorithm

Genetic algorithms are well known method of bio-inspired optimization. However, there are different views on the sequence and content of the steps of genetic algorithms [14, 15, 16]. The paper proposes to follow the following sequence of steps:

1. *Defining the fitness function,* which shows why one solution better than the other solution.

2. *Encoding possible solutions of the problem.* The encoded solution in the terminology of genetic algorithms is called *individual*. Usually solutions are encoded using character or numeric strings. A single character of this code is called *gene*. A set of genes in a string is called *chromosome*.

3. *Creating the initial set of individuals* that called *population*. Typically, this process takes place at random, but the number of individuals in the population N is permanent. This step includes also evaluating all individuals in a population through fitness function and sorting them in descending order of its value.

4. Choosing pairs of individuals, called *parents*, to form new individuals, called *descendants*, by *crossing*. Parents' chromosomes are broken into fragments. Then the fragments of parental chromosomes are crossed to form descendant chromosomes. New individuals are assessed using fitness function and are added to the current population.

5. Choice of individuals for *mutations*, the mutation and evaluation of these individuals. Under mutation, the individuals modify their genes.

6. *Selection of the population*, which consists in reservation of the N individuals possessing the highest values of fitness functions. Other individuals (the "bad") are removed from the current population.

7. If the algorithm termination criteria are fulfilled, then an individual with a maximum value of the fitness function is a solution. Otherwise, return to step 3.

In order to use the genetic algorithm to solve the problem discussed, it is necessary to determine the fitness function and coding decisions.

In the known papers on the application of genetic algorithms for solving the BMF problem, the fitness function was based on the Euclidean distance between the matrices $\mathbf{A}$ and $\mathbf{W} \otimes \mathbf{H}$. Under full matching of these matrices the fitness function has the maximum value. In our case only one Euclidean distance between $\mathbf{A}$ and $\mathbf{X} \otimes \mathbf{X}^{\mathrm{T}}$ is not sufficient. The trivial solutions may be obtained from this one

criterion. We must also take into account the requirement that in the matrix $\mathbf{X}$ the number of columns k should be minimal. So we propose the following type of fitness functions:

$$F = \left(\alpha k + \beta \sqrt{\sum_i \sum_j (a_{ij} - x_{ij} x_{ji})^2} \right)^{-1}, \tag{4}$$

where α and β are the weights that determine the direction of the solution search. The condition $\alpha \ll \beta$ between these coefficients guarantees that first of all we want to look for solutions which matrices $\mathbf{A}$ and $\mathbf{X} \otimes \mathbf{X}^{\mathrm{T}}$ are the same, then we will look for solutions with smaller values of k.

Now, let us consider the order to create chromosomes. To do this, we propose to use as genes of chromosomes the vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})$. The vector $\mathbf{x}_i$ plays the role of a column of the matrix $\mathbf{X}$. Let us set the number of genes in the chromosome equal M. In fact, the number M specifies the number of columns in a trivial solution. More columns make no sense.

Finally, in order to increase the convergence of the genetic algorithm we will make another proposal, which deals with the formation of the initial population. We will form it from possible trivial decisions. If we get $P \leq N$, then the initial population will consist of trivial solutions. Otherwise P individuals are trivial solutions, and the other $(N - P)$ individuals will be randomly generated.

Operation of crossing and mutation will be performed in a standard way.

5 Experimental Results

The proposed approach to the formation of an additional level of protection for the computer network was implemented in a computer network that was physically deployed on three floors of a building. It used Cisco Catalyst 5000 Series switches. The dynamic VLAN technology was implemented based on MAC addresses. For each computer the access lists to the neighboring computers both at the level of one floor of the building and between floors was made by the system administrator. Access lists have served as the basis for the formation of the original matrix $\mathbf{A}$. Each access list forms one row in $\mathbf{A}$. The total number of computers covered under the VLAN and, accordingly, specified by the matrix $\mathbf{A}$, is equal to 100. The larger number of computers was considered inappropriate because of difficulties in forming the matrix $\mathbf{A}$. Using the genetic algorithm, a solution was found, that consists in splitting the network to about 50 of large and small virtual subnets, which were formed by the security administrator. Users have not identified the decrease of the network performance. However, the users were completely prevented from making unauthorized access to neighboring computers.

For evaluation of speed and accuracy of the genetic algorithm, the testbed was developed, the structure of which is shown in Fig. 2.

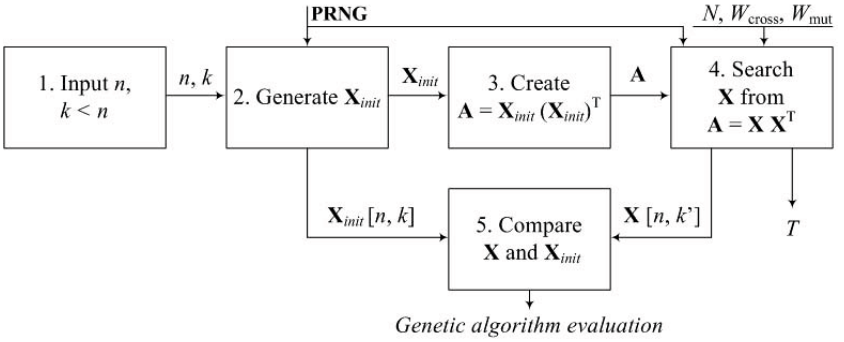

Fig. 2 Testbed for the genetic algorithm assessment

The parameters n and $k(k < n)$ are formed in the module 1 as the *initial data of the testbed*. Using a pseudorandom number generator (PRNG) [17], the initial matrix $\mathbf{X}_{\text{init}}$ of computer distribution over virtual subnets is randomly generated in the module 2. In the module 3, the matrix $\mathbf{A}$ is calculated as the initial data of the problem based on the formula $\mathbf{A} = \mathbf{X}_{\text{init}} \otimes (\mathbf{X}_{\text{init}})^{\text{T}}$. Then, in the module 4, taking into account the matrix $\mathbf{A}$ and using the genetic algorithm, the matrix $\mathbf{X}[n, k']$ is determined, which is the solution of the problem. In general case, the number of columns k' in the matrix $\mathbf{X}$ is not equal to the value of k. If $k' \leq k$, then we consider that the algorithm has found the exact solution of the problem, otherwise — approximate. The algorithm parameters are: number of individuals in the population (N), the probability of choosing the individuals for crossover operation (W_{cross}) and the probability of choosing the individuals for the mutation (W_{mut}). These values were taken from [10]: $N = 200, W_{\text{cross}} = 0.1, W_{\text{mut}} = 0.01$. The outputs of the module 4 are matrix $\mathbf{X}$ and the number of iterations of the algorithm (T). The maximum number of iterations was set to $T_{\max} = 1000$. The exceeding of this value leads to the algorithm completion. Finally, the module 5 fulfills the final evaluation of the algorithm by comparing matrices $\mathbf{X}$ and $\mathbf{X}_{\text{init}}$.

The speed assessment for the algorithm was evaluated, while the number of iterations did not reach the limit (i.e. $T < T_{\max}$). The measure of speed was the average of the number of iterations $(\bar{T})$, determined for the pair of values (n, k). Five tests were carried out for each pair (n, k). The n value were changed in the range from 10 to 100. The k value had the following values: $0.2n, 0.3n$ and $0.5n$.

The accuracy δ of the algorithm was evaluated when the iteration count reached the limit (i.e. $T = T_{\max}$) according to the following formula:

$$\delta = [(n - \max(0; k' - k))/n] \cdot 100\%. \tag{5}$$

In this case, if the condition $k' = k$ hold true, then regardless of the n value the accuracy is 100%. If $k' > k$, then the accuracy depends on the relationship between the $(k' - k)$ and the n values. So, if $n = 50, k = 10$ and $k' = 11$, then $\delta = 98$.

The results of the speed and accuracy evaluation for the genetic algorithm proposed in the paper are outlined in Tab. 1.

Table 1 Experimental results

n	k	$\bar{T}$	$\delta, \%$	n	k	$\bar{T}$	$\delta, \%$
10	2	48.3	100	50	25	534.6	100
10	3	35.6	100	75	15	1000.0	91
10	5	25.8	100	75	23	1000.0	95
25	5	255.8	100	75	38	1000.0	97
25	8	183.7	100	100	20	1000.0	78
25	13	127.5	100	100	30	1000.0	85
50	10	1000.0	98	100	50	1000.0	90
50	15	811.4	100				

Analyzing the data presented in Tab. 1, it is possible to draw the following conclusions. First, under the small dimensions of the problem ($n = 10$ and $n = 25$) the proposed algorithm allowed to generate the solution with maximum accuracy δ = 100% in real or near-real-time. Secondly, for large dimensions ($n = 75$ and $n = 100$) it was impossible to obtain the maximum accuracy in the allotted time. However, the accuracy is quite large and ranges from 78 to 97 percent. Finally, when n is constant, the complexity of solving the problem increases when k decreases. It is well illustrated by the $n = 50$. When $k = 15$ and $k = 25$, the most accurate solutions were received; and when $k = 10$, the accuracy of the solution was 98 percent.

The analysis of experimental results leads to the conclusion that the proposed algorithm has sufficiently high performance for designing virtual computer networks.

6 Conclusion

The paper proposed the approach to the construction of an additional level of protection for computer network based on the formation of virtual subnets according to the specified requirements to logical connectivity between computers. As the main tool to create virtual subnets, VLAN technology is considered.

Analysis of the mathematical statement of the problem showed that it is a type of Boolean Matrix Factorization. However, as distinct from the BMF which looks for two different Boolean matrices, the considered problem searches only one Boolean matrix, such that the multiplication of this matrix and its transposed image gives the desired result. For this reason, it was concluded that using known mathematical

methods of the BMF directly is inappropriate. To solve the problem, the paper suggests to use genetic algorithms. It was shown that known genetic algorithms used for solving the problems of BMF or virtual network schema optimizations cannot be directly used and require some improvements. Key improvements proposed in the paper are: the use of trivial solutions to generate an initial population; taking into account in the fitness function the criterion of minimum number of virtual subnets; the use columns of the connectivity matrix as genes of the chromosomes encoding the solutions. Experimental evaluation of speed and accuracy for the proposed genetic algorithm has shown its high efficiency. Future studies are directed to the application of the proposed approach to the reconfiguration of virtual subnets.

Acknowledgements. This research is being supported by the grants of the Russian Foundation of Basic Research (13-01-00843, 13-07-13159, 14-07-00697, 14-07- 00417), the Program of fundamental research of the Department for Nanotechnologies and Informational Technologies of the RAS (contract #2.2), and by Government of the Russian Federation, Grant 074-U01, and State contract #14.604.21.0033.

References

1. Catalyst 2900 Series XL and Catalyst 3500 Series XL Software Configuration Guide. Cisco IOS Release 12.0(5) WC(1). Cisco Systems, San Jose (2001)
2. Miettinen, P., Vreeken, J.: Model Order Selection for Boolean Matrix Factorization. In: 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, New York (2011)
3. Miettinen, P.: Dynamic Boolean Matrix Factorizations. In: 2012 IEEE 12th International Conference on Data Mining. ACM, New York (2012)
4. Cergani, E., Miettinen, P.: Discovering Relations using Matrix Factorization Methods. In: 22nd ACM International Conference on Information & Knowledge Management. ACM, New York (2013)
5. Lu, H., Vaidya, J., Atluri, V., Hong, Y.: Extended Boolean Matrix Decomposition. In: Ninth IEEE International Conference on Data Mining. IEEE Press, New York (2009)
6. Lu, H., Vaidya, J., Atluri, V.: Optimal Boolean Matrix Decomposition: Application to Role Engineering. In: 24th IEEE International Conference on Data Engineering. IEEE Press, New York (2008)
7. Saenko, I., Kotenko, I.: Genetic Algorithms for Role Mining Problem. In: 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing. IEEE Press, New York (2011)
8. Saenko, I., Kotenko, I.: Design and Performance Evaluation of Improved Genetic Algorithm for Role Mining Problem. In: 20th International Euromicro Conference on Parallel, Distributed and Network-based Processing. IEEE Press, New York (2012)
9. Janecek, A., Tan, Y.: Using Population Based Algorithms for Initializing Nonnegative Matrix Factorization. In: Tan, Y., Shi, Y., Chai, Y., Wang, G. (eds.) ICSI 2011, Part II. LNCS, vol. 6729, pp. 307–316. Springer, Heidelberg (2011)
10. Snasel, V., Platos, J., Kromer, P.: On Genetic Algorithms for Boolean Matrix Factorization. In: Eighth International Conference on Intelligent Systems Design and Applications, vol. 2, pp. 170–175. IEEE Press, New York (2008)
11. Snasel, V., Platos, J., Kromer, P., Husek, D., Neruda, R., Frolov, A.A.: Investigating Boolean Matrix Factorization. In: Workshop on Data Mining using Matrices and Tensors (2008)
12. Tai, C.-F., Chiang, T.-C., Hou, T.-W.: A Virtual Subnet Scheme on Clustering Algorithms for Mobile Ad Hoc Networks. Expert Systems with Applications 38(3), 2099–2109 (2011)

13. Saenko, I., Kotenko, I.: Genetic Optimization of Access Control Schemes in Virtual Local Area Networks. In: Kotenko, I., Skormin, V. (eds.) MMM-ACNS 2010. LNCS, vol. 6258, pp. 209–216. Springer, Heidelberg (2010)
14. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Longman Publishing, Boston (1989)
15. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Massachusetts (1998)
16. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer (2007)
17. Barker, E., Kelsey, J.: Recommendation for Random Number Generation Using Deterministic Random Bit Generators. NIST Special Publication. NIST (2012)