# Administrating Role-Based Access Control by Genetic Algorithms

Igor Saenko[1,2]
[1] Saint-Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences
14-th Liniya, 39, St.Petersburg, 199178, Russia
[2] St. Petersburg National Research University of Information Technologies, Mechanics and Optics,
49, Kronverkskiy prospekt, St.Petersburg, Russia
ibsaen@comsec.spb.ru

Igor Kotenko[1,2]
[1] Saint-Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences
14-th Liniya, 39, St.Petersburg, 199178, Russia
[2] St. Petersburg National Research University of Information Technologies, Mechanics and Optics,
49, Kronverkskiy prospekt, St.Petersburg, Russia
ivkote@comsec.spb.ru

## ABSTRACT[1]

In the paper we address the problem of administering Role-Based Access Control (RBAC) systems, which consists in substantiation of the choice of solution of the RBAC design or reconfiguration problem based on using genetic algorithms (in conditions of dynamically changing access control policies). The problems of RBAC design and reconfiguration are NP-complete. Therefore, the use of genetic algorithms to solve it seems to be quite appropriate. The paper discusses the mathematical basis for selection of administrative tasks and novelties, implementation of which allows to increase the speed of the developed genetic algorithms. We also consider the structure and possibilities of the developed testbed and the results of its application to estimate the administrative costs to implement changes to an access control policy using the solutions of administration tasks. Additionally, using the testbed we evaluated the gain in speed of the genetic algorithms that was obtained by implementing the proposed contributions, such as creating two chromosomes for each individual, using columns of Boolean matrices as genes of chromosomes, as well as additional procedures that improve the operations of crossover, mutation and selection. The results of the experiments showed high efficiency of the proposed algorithms.

## CCS CONCEPTS

• **Security and privacy → Access control;** *Computing methodologies*; Heuristic function construction

## KEYWORDS

RBAC, role mining problem, access control, genetic algorithm

**ACM Reference format:**
I. Saenko and I. Kotenko, 2017. Administrating Role-Based Access Control by Genetic Algorithms. In GECCO '17: Genetic and Evolutionary Computation Conference Companion Proceedings, 8 pages.
DOI: 10.1145/3067695.3082509


*GECCO '17 Companion*, July 15-19, 2017, Berlin, Germany

ACM ISBN 978-1-4503-4939-0/17/07…$15.00
http://dx.doi.org/10.1145/3067695.3082509

## 1 INTRODUCTION

Currently Role-Based Access Control (RBAC) is a widely used access model, which is applied in various systems. Among RBAC systems there are databases, operating systems, cloud infrastructure and others. The basic idea of the RBAC model is to replace the mapping "users-permissions" to the series-connected mappings "user-role" and "roles-permissions". This idea was first proposed in [1]. RBAC model has found wide popularity because its use requires less administrative costs to design access patterns compared to other known models.

Finding mappings "users-roles" and "roles-permissions" that meet the security policy requirements is a complex task from the field of Data Mining. This task has its own name Role Mining Problem (RMP) [2]. To solve this problem a number of different statements, methods and algorithms were proposed [3, 4]. In our previous papers for this task there were proposed genetic algorithms (GAs) [5-7], which showed a fairly high efficiency. The use of these methods and algorithms allows the administrator of an RBAC system to significantly improve the quality of RBAC design, while avoiding the common pitfalls that are inherent to this work. Such errors are the errors of the "false ban" and the "false permission" of the authority which violate the requirements for confidentiality and availability of information in the RBAC system. The solution to the problem of RBAC design in the variant *Basic RMP* or in the variant *Edge RMP* guarantees the absence of such errors.

A significant feature of the RMP is that in case of changing access control policies specified by a matrix of "users-permissions", it does not take into account the previous state of this policy. In other words, methods and algorithms for solving RMP are sufficient to administer RBAC systems only in the case when the policy of access control is not changed during the entire time the user works with this RBAC system. However, this case rarely occurs in practice. Much more often there are cases when at the request of management of the company or organization it is necessary to change permissions of the users belonging to either role. As a result, the administrator gets the following problem: either to form a new role for these users, or completely redesign RBAC using the RMP solution methods. Unfortunately, in most cases, the administrator decides to make formation of a new role, and the RBAC design can require considerable time. As a result, the number of roles in RBAC systems, in which its access control policy quite often changed without having a serious reason for it, increases rather strongly. As a result, the administrator may completely lose control of the RBAC system, and the errors of the first and second kind can re-emerge.

In order to scientifically justify the actions of the RBAC system administrator in the case of multiple changes of access control policies, in addition to the RMP, the problem of RBAC reconfiguration should be put in consideration. The main difference between the RBAC reconfiguration task from the RMP task is as follows: the previous RBAC scheme is introduced into the composition of the original data; the main criterion is minimization of administration costs. As with RMP, the RBAC reconfiguration task is NP-complete. In [8] for its solution we proposed an approach based on GAs. Other known methods and algorithms of its solution, as shown by the analysis of known works on to the RMP, were not found.

However, the joint use of the developed methods and algorithms to solve the RMP and RBAC reconfiguration tasks puts for the RBAC system administrator a *new problem of selection* of the most effective of these tasks. For this purpose it is necessary to determine how the cost of RBAC administering will be changed in case of implementation by the administrator of one of the tasks mentioned above.

This determines the *main goal* of the paper, which consists in developing tools to justify actions, necessary for the RBAC system administrator in case of repeated changes in access control policies, and capable to perform the following functions: (1) to solve RBAC administrative tasks (design and reconfiguration) of using GAs; (2) to simulate sequential changes in the access control policy; (3) to assess costs, required by the RBAC system administrator in order to implement the solution of particular administrative tasks. As *administrative tasks in RBAC* the problems of RBAC design in versions *Basic RMP* or *Edge RMP* and the task of the RBAC reconfiguration are studied.

Additionally, this toolkit meets the problem of comparative evaluation of the GAs, designed to solve the tasks of RBAC administering, which incorporate a number of novelties that increase the speed of work, with algorithms that do not have these novelties. The purpose of this assessment was to determine the effect of these novelties. In our previous paper these studies have not been conducted.

The main *theoretical contribution* of this paper is the following. *First*, we proposed the mathematical fundamentals to justify the selection by the RBAC administrator of one or another administration task in case of repeated changes in access control policies. *Second*, we clarified and developed a number of novelties implemented in the developed GAs, which allow to significantly increase the speed of their work. These novelties are as follows: (1) each individual in a population has two chromosomes instead of one; (2) as the genes of chromosomes we use the columns of Boolean matrices; (3) we proposed to implement a number of additional operations that improve the operations of crossover, mutation and selection.

Further structure of the paper is as follows. Section 2 overviews the related work. Section 3 considers the mathematical foundations of the problem. Section 4 discusses the issues of development of the GA. Section 5 outlines the experimental results. Conclusions and further research directions are in section 6.

## 2 RELATED WORK

In [2, 4] various versions of the RMP statement (*Basic RMP, Edge RMP* and others), which differ in the criteria for assessment of the access control scheme, were formulated. In these works it is proved that all variants of the RMP are NP-complete problems.

For solving different variants of the RMP the various approaches were proposed. Thus, [9] proposed a cluster approach that requires taking into consideration additional parameters that characterize business processes and user needs. Simple heuristic algorithms for solving the problem in the variant *Basic RMP* were proposed in [10, 11]. They are based on combinatorial decisions. To reduce their complexity in [12, 13] the probabilistic models are proposed. However, the high accuracy of the solution is not guaranteed.

The approach based on Boolean data clustering was proposed in [14]. It is shown that this model is applicable to solve certain variants of the RMP problem. Cost-driven approach is suggested in [15]. It uses a criterion that takes into account the costs of administration. This criterion is used in the present paper for the task of reconfiguring the RBAC. Despite the presence of many different algorithms for solving the RMP, none of them are universal, suitable for any variant. In our previous works [5, 6, 16] we proposed and investigated such GA.

The issues of RBAC reconfiguration are partially addressed in [17-19]. In [17] it was noted that for RMP the reconfiguration the access history logs can be used. [18] shows the possibility of the RMP reconfiguration with negative links. In [19] they proposed individual statements of RMP with different restrictions, which can be considered only as special cases for the RBAC reconfiguration problem.

[20] suggests to use a greedy algorithm and a randomized rounding algorithm. However, these algorithms belong to the class of approximate algorithms.

There are several known works, which investigate the ability of application of GAs for the purposes of access control and computer security. For example, the work [21] can be considered as a good example of applying GAs in computer security problems. However, here these algorithms are used in conjunction with the RBAC, and not for the RBAC creation.

In [22] authors successfully apply GAs for access control of Web services. However, these tasks are less complex than RMP. In [23] a GA is used to solve multi-objective optimization problem to find the network architecture and the Medium Access Control protocol parameters that achieve the Pareto-optima in a computationally efficient manner. However, in this problem the variables have scalar values. Therefore, this task is less complicated than RMP.

Paper [24] examines the architecture of Intrusion Detection System that uses a GA, characterized by the use of both temporal and spatial information of the generated rule set. However, in this system the GA is standard. Each species has one chromosome. Chromosomes consist of binary elements.

In [25], dedicated to the application of GAs for solving the problem of selecting a minimal number of optimally positioned monitors to capture network traffic, also one chromosome with binary elements is used.

Thus, the analysis of related publications shows that, despite the presence of certain algorithms, for solving the RMP, as well as the problem of RBAC reconfiguration, including GAs, the issues of justification of selection by the RBAC system administrator of the most appropriate task (for various conditions of changing the access control policies) in well known works are not considered.

## 3 MATHEMATICAL BACKGROUND

RBAC model is set up by three mappings, defined on the following sets [2, 7, 10, 19]: $U = \{u_i\}, i = 1,..., m, m = |U|$ – set of users; $PRMS = \{p_j\}, j = 1,..., n, n = |PRMS|$ – set of permissions; $ROLES = \{r_l\}, l = 1,..., k, k = |ROLES|$ – set of roles.

The first mapping is established between the set of users *U* and a set of permissions *RPMS*. Let us set it in the form of $m \times n$ the Boolean matrix **A** in which a 1 in cell $\{ij\}$ indicates the assignment of permission *j* to user *i*. Matrix **A** determines initially defined mapping, which is defined by the security policy and which need to be performed using the RBAC model. The second mapping (defined as *UA*) is set between the set of users *U* and the set of roles *ROLES*. Similarly to **A**, let us set it as $m \times k$ Boolean matrix **X**. The third mapping (defined as *PA*) is set between the set of roles *ROLES* and the sets of permissions *PRMS*. Let us represent it as $k \times n$ Boolean matrix **Y**.

The essence of role-based access is the sequential application of the mappings **X** and **Y**, and the result should be the mapping **A**. This requirement can be written as follows:

$$\mathbf{X} \otimes \mathbf{Y} = \mathbf{A}, \quad (1)$$

where symbol $\otimes$ denotes Boolean matrix multiplication.

The main task of the RBAC system administrator is the creation of the mappings **X** and **Y** to satisfy the condition (1). However, the matrix equation (1) has a very large number of solutions. Therefore, when finding **X** and **Y** it is necessary to consider additional criteria. The most known criteria are: (1) the minimum number of roles; (2) the minimum total number of unit elements in the matrices **X** and **Y**. The usage of additional criteria transforms the problem (1) to the problem known as *Role Mining Problem* (RMP) [2, 4]. The variant of the RMP, with considering the first criterion, received the name of the *Basic RMP,* and the RMP option, when considering the second criterion, is called the *Edge RMP*. Formally, the first criterion has the form

$$|ROLES| \Rightarrow \min, \quad (2)$$

and the second criterion has the form

$$|UA| + |PA| \Rightarrow \min. \quad (3)$$

Let us call the task that the RBAC system administrator solves when searching for matrices X and Y from (1) with (2) or (3) *the task of RBAC design.* The initial data for this problem are sets *U* and *PRMS*, as well as the required matrix **A**. The task of RBAC design is usually resolved before users work with the RBAC system.

At the same time, often some time after the beginning of work with the RBAC system there is a need to change the requirements for access control, i.e. to change the matrix **A**. In this case the administrator needs to find new matrices **X** and **Y** that satisfy (1). However, if he (she) will consider (1) or (2), it may lead to a rather large number of changes in the RBAC schema. In other words, it can lead to quite a large administrative overhead that is not always possible or inappropriate.

Another approach to transition to the new matrix **A** is given its previous view $\mathbf{A}_0$. The challenge lies in finding such matrices **X** and **Y**, when the administrative costs of transition to them from the previous matrices $\mathbf{X}_0$ and $\mathbf{Y}_0$ will also be minimal. This task will be called the *task of RBAC reconfiguration*. Its main difference from the RBAC design task is that in this case the original data includes not only the matrix **A** and sets *U* and *PRMS,* but also the previous matrix $\mathbf{A}_0$ and the previous matrices $\mathbf{X}_0$ и $\mathbf{Y}_0$.

We introduce into consideration two new matrices $\Delta\mathbf{X}$ and $\Delta\mathbf{Y}$. Matrix $\Delta\mathbf{X}$ is defined as $\Delta\mathbf{X} = \mathbf{X}_0 \oplus \mathbf{X}$, and matrix $\Delta\mathbf{Y}$ as $\Delta\mathbf{Y} = \mathbf{Y}_0 \oplus \mathbf{Y}$, where symbol "$\oplus$"denotes "exclusive OR". Then, formally, the criteria for the task of RBAC reconfiguration can be written in the following form:

$$(\mathbf{X}_0 \oplus \Delta\mathbf{X}) \otimes (\mathbf{Y}_0 \oplus \Delta\mathbf{Y}) = \mathbf{A}, \quad (4)$$

$$\sum_{i=1}^{m}\sum_{l=1}^{k} \Delta x_{il} + \sum_{l=1}^{k}\sum_{j=1}^{n} \Delta y_{lj} \rightarrow \min. \quad (5)$$

Thus, in the course of work of RBAC system in case of change of access control requirements (i.e. matrix **A**) the RBAC administrator meets this selection problem: (1) either to solve the problem of RBAC design that is presented in the version of the *Basic RMP* (equations 1 and 2) or in variant *Edge RMP* (expressions 1 and 3); (2) or to solve the problem of RBAC reconfiguration (expressions 4 and 5).

In all three cases, the selection criterion is the amount of administrative costs, which are determined by the expression (5).

Two important factors should be noted. First, administrative costs in solving the problem of RBAC design can significantly exceed the same costs for RBAC reconfiguration. However, we cannot exclude the possibility of implementing by an administrator of the design task in practice at the appropriate time. Second, the choice of his possible practical actions (i.e., to implement the solution of the RBAC design task or the RBAC reconfiguration task) the administrator must first directly address these challenges and compare the results using (5). The RBAC design and reconfiguration are the tasks of Boolean matrix factorization and belong to the class *NP*-complete. Therefore, for their effective solution the heuristic methods are necessary. In our previous papers we investigated the possibility of applying GAs for these purposes. To improve the accuracy and speed of GAs there were proposed a number of novelties to implement the algorithms. The composition of these novelties in our studies are continuously changing. The next section considers these novelties in their final form, as well as other features of the development of GAs for solving the tasks of RBAC administration.

## 4 GENETIC ALGORITHMS

### 4.1 Structure of chromosomes

In general, the developed GAs have a well known structure. They contain a block of generating the initial population, the block of the current iteration, including operations of crossover, mutation and selection, and unit of completion. Features of the development include the following: structure of chromosomes; type of fitness functions; the peculiarities of implementation of crossover, mutation and selection.

It is known that chromosomes in GAs encode the possible solutions of the problem. Therefore, in the case of RBAC administrative tasks the chromosomes need to encode the matrices **X** and **Y**. These matrices are asymmetric. Therefore, the initial apparent decision to set up the structure of chromosomes is pulling all rows of the matrices **X** and **Y** into one big string. Each gene of this chromosome will correspond to a particular element

in the matrix **X** or **Y**. However, this solution has several drawbacks. First, you will need to implement several points of crossover, which increases the complexity of the operation. Second, the convergence of the algorithm is not very high, as the new individuals resulting from crossover or mutation will not be much different from the existing ones in the population. In addition, among new individuals very high percentage of defects is possible, when new individuals are not suitable for RBAC in its physical sense and are not considered further.

We propose a new approach to formation of chromosomes, which is characterized by the following two features.

First, it is proposed that each individual in the population had not one, as usual, but two chromosomes. One of these chromosomes $Chr_X$ will encode the matrix **X** and the second chromosome $Chr_X$ – the matrix **Y**. In this approach, at the operation of crossover, each chromosome is subject to division. It is sufficient to have just a single point of crossover on each chromosome to ensure absence of "defective" individuals at crossover and mutation, and quite high convergence.

Secondly, as genes of the chromosome $Chr_X$ and $Chr_Y$ we suggest to use not isolated elements of Boolean matrices **X** and **Y,** but their columns. In this case instead of the matrix **Y** we use its transposed form $\mathbf{Y}^T$. In this case, the columns of the matrices **X** and $\mathbf{Y}^T$ correspond to the elements of the set *ROLES* in the RBAC model, and an isolated column elements will show their connections with the elements of the set *U* (for matrix **X**) and the elements of the set of *PRMS* (for the matrix $\mathbf{Y}^T$).

We illustrate this approach by the following example [4]. Let $n = 4$, $m = 5$ and matrices **A, X** and $\mathbf{Y}^T$ are as follows:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}, \tag{6}$$

$$\mathbf{X} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \mathbf{Y}^T = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{7}$$

From (7) it can be seen that this scheme contains only three RBAC roles, i.e. $k = 3$. It is also easy to check that the condition $\mathbf{X} \otimes \mathbf{Y} = \mathbf{A}$ is true.

A graphical view of these chromosomes is shown in Fig. 1. It should be noted that each column in the chromosomes $Chr_X$ and $Chr_Y$ is a binary number. In order to facilitate software implementation of the GA we shall replace this number with the corresponding decimal number. For example, the first column of the chromosome $Chr_X$ corresponds to a binary number (0101). It corresponds to decimal number 5. The result of the chromosome presented in Fig. 1, can be written in the following form: $Chr_X$ = (5, 2, 14); $Chr_Y$ = (28, 26, 9). The initial population of individuals with two chromosomes with structures shown in Fig.1, is randomly generated with fixed values of *n* and *m*. The number of roles *k* is changed randomly in the range from 1 to *n*. The number of individuals in the population is determined by the algorithm parameter $N_{pop}$. On the basis of existing experience with GAs, value of $N_{pop}$ was set to 200.

$$Chr_X = \left(\begin{bmatrix}0\\1\\0\\1\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}1\\1\\1\\0\end{bmatrix}\right); \; Chr_Y = \left(\begin{bmatrix}1\\1\\1\\0\\0\end{bmatrix}, \begin{bmatrix}1\\1\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\\1\end{bmatrix}\right).$$

**Figure 1: Structure of chromosomes.**

## 4.2 Fitness functions

*4.2.1 Basic RMP.* The fitness function must reflect the search criteria that are studied in the problem being solved. To solve the problem of RBAC design in variant of *Basic RMP* these criteria are set by the expressions (1) and (2). It is therefore proposed to use the fitness function in the following form:

$$F = w_1 k + w_2 \sum_{i=1}^{n} \sum_{j=1}^{m} \left| a_{ij} - \sum_{l=1}^{k} x_{il} y_{lj} \right|, \tag{8}$$

where $w_1$ and $w_2$ are the weight coefficients. Between the weighting factors ratio of $w_1 << w_2$ is set. This ensures that during the GA in the first place it will be finding solutions that meet the condition (1), and then search for solutions with smaller *k* values.

*4.2.2 Edge RMP.* For the problem of designing an RBAC in variant *Edge RMP* the search criteria are specified by expressions (1) and (3). In this case, the next fitness function is proposed:

$$F = w_1 \sum_{i=1}^{n} \sum_{j=1}^{m} \left( \left| x_{ij} \right| + \left| y_{ji} \right| \right) + w_2 \sum_{i=1}^{n} \sum_{j=1}^{m} \left| a_{ij} - \sum_{l=1}^{k} x_{il} y_{lj} \right|. \tag{9}$$

Then at first it will be finding solutions that meet the condition (1), and then solutions satisfying criterion (3).

*4.2.3 Reconfiguration.* For the task of RBAC reconfiguration the search criteria of the solutions are set by expressions (4) and (5). The fitness functions will have following form:

$$F = w_1 \left( \sum_{i=1}^{m} \sum_{l=1}^{K} \left| x0_{il} \oplus x_{il} \right| + \sum_{j=1}^{n} \sum_{l=1}^{K} \left| y0_{jl} \oplus y_{jl} \right| \right) + \\ + w_2 \sum_{i=1}^{n} \sum_{j=1}^{m} \left| a_{ij} - \sum_{l=1}^{k} x_{il} y_{lj} \right| , \tag{10}$$

where $\{x0_{il}\}$ is the matrix elements $\mathbf{X}_0$, $\{y0_{jl}\}$ are the elements of the matrix $\mathbf{Y}_0$. At first it will be finding solutions that meet the condition (4), then solutions that satisfy the criterion (5).

## 4.3 Features of crossover, mutation and selection

*4.3.1 Crossover.* The selection of pairs of parent individuals for the crossover is performed in accordance with probability $W_{cross}$, which is a parameter of the GA. Based on the experience with GAs, this parameter was equal to 0.1. Each chromosome $Chr_X$ and $Chr_Y$ is divided into two parts in own crossover points. Then, from the resulting parts there are formed possible chromosomes for individuals-descendants. As a result of the crossover $4 = 2^2$ new individuals-descendants are formed. In order to make it possible to execute the crossover to the parent chromosomes having different value of *k*, the following two actions are done: (1) the synchronization of the number of roles between parents; (2) synchronization of the number of roles between chromosomes

of the descendant. In the first case, the parent chromosome with the smaller number of elements, is supplemented by the necessary number of zero elements, so the length of each chromosome is the same. In the second case, if during the execution of the crossover in the middle of the chromosome-descendant zero elements appeared, they are automatically transferred to the tail of the chromosome. Further, the number of null elements in the tails of the chromosomes of the descendant is aligned by zeroing the tail of the chromosome of the desired number of nonzero elements. These actions make the crossover more constructive that was confirmed by us further experimentally.

The above-described procedure of crossover is illustrated in the example shown in Fig. 2. The parent individual *Parent 1* is taken from Fig. 1. It has $k = 3$. The second parent individual *Parent 2* has $k = 2$. By synchronizing her chromosomes were increased by one zero element, placed at the end of the row. After performing the crossover was formed by four individuals descendants: *Descendant 1*, *Descendant 2*, *Descendant 3*, and *Descendant 4*, as shown in Fig. 2. To align the number of null elements in the chromosomes in the descendants, after the second syncronization there were done the following changes: for individual *Descendant 2* the chromosome $Chr_Y = (26, 21, 9)$ was replaced by the chromosome $Chr_Y = (26, 21, 0)$, and for the individual *Descendant 3* the chromosome $Chr_X = (12, 2, 14)$ was replaced on the chromosome $Chr_X = (12, 2, 0)$.

*4.3.2 Mutation.* The selection of individuals for mutation is performed in accordance with the probability of $W_{mut}$, which is a parameter of the GA. Based on the experience with GAs, this parameter was equal to 0.01. During mutation of individuals two procedures of selection are done, as shown in Fig. 3. The first procedure selects randomly with a probability $W_{mut1}$ the genes in the chromosomes $Chr_X$ and $Chr_Y$, which correspond to the columns of the matrices $\mathbf{X}$ and $\mathbf{Y}^T$. The second procedure with probability $W_{mut2}$ selects elements which values are then subject to inversion. The probability of selecting a column for mutations $W_{mut1}$ and the probability of selecting a item in the column for mutations $W_{mut2}$ are parameters of the algorithm. These parameters had the following values: $W_{mut1} = W_{mut2} = 0.5$.

*4.3.3 Selection.* New individuals obtained as a result of performing crossover and mutation, were added to the current population of individuals. However, to prevent the accumulation in the population of individuals with identical sets of chromosomes, before adding the new individual in the population, its chromosomes were checked for uniqueness of individuals relative to those already in the population. If this test was unsuccessful, then a new individual was excluded from consideration. After adding new individuals in the population they carried out sorting of all existing individuals of population by the value of the fitness function. In the population was only $N_{pop}$ individuals whose values of fitness function were minimal. The remaining individuals were removed from the population. At this the current iteration was ended, and the algorithm passed to the new iteration to implement crossover, mutation and selection.

*4.3.4 Termination.* The termination of work of the GA was done in two cases. First, when the preset maximum number of iterations was achieved. This number depended on the dimension of the problem. Second, if the population for a sufficiently large number of iterations (e.g., ten) did not changed the minimum value of fitness function. In this case it was considered that we obtained the desired solution of the problem.

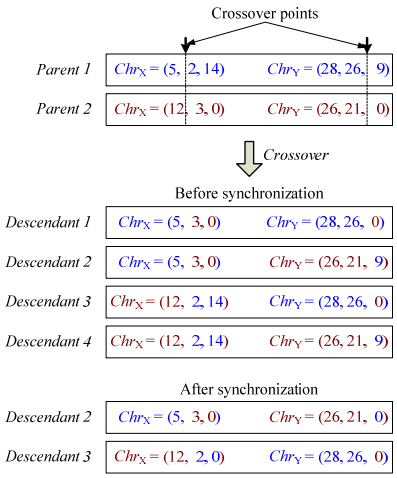


**Figure 2: Crossover.**

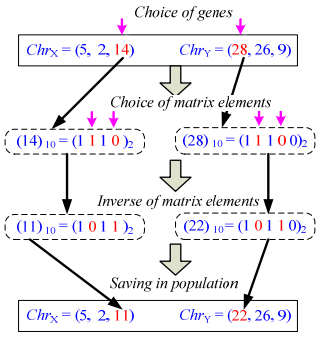


**Figure 3: Mutation of chromosomes.**

# 5 TESTBED AND EXPERIMENTS

## 5.1 Testbed

To estimate the rate of the developed GAs, as well as costs required by the administrator to modify the RBAC schema on the results of solving problems of the RBAC design or reconfiguration, the testbed was developed. Programming language was C#. Testbed has the following functions: (1) Generation of matrices $\mathbf{A}$ and $\Delta\mathbf{A} = \mathbf{A} \oplus \mathbf{A}_0$. (2) Finding the solution of the problem of RBAC design (in versions *Basic RMP* and *Edge RMP*) and the task of RBAC reconfiguration. (3) Evaluation of the operating speed of the developed GAs for different dimensions. (4) Estimation of administrative cost required to solve the problems of RBAC design or reconfiguration. The structure of the testbed is in Fig. 4.

Testbed consists of the following modules: (1) *Initiator*, in which the administrator enters the initial data and parameters of GAs. (2) *Generator*, in which matrices $\mathbf{A}$ and $\Delta\mathbf{A}$ are generated. (3) *GAs*, in which the solutions are found for the problem of RBAC design (in versions *Basic RMP* and *Edge RMP*) and the task of RBAC reconfiguration. These solutions are presented in the form of matrices $\mathbf{X}$ and $\mathbf{Y}$. (4) *Speed Evaluator*, in which the evaluation of the speed of the GAs is performed. (5) *Cost Evaluator,* in which the costs required to solve the problems of RBAC design or reconfiguration are evaluated.

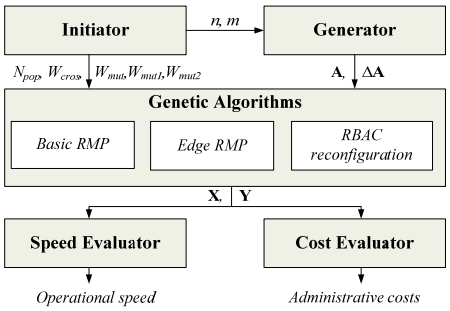


**Figure 4: The structure of the testbed.**

Evaluation of the speed of GAs in the module *Speed Evaluator* was performed by determining the number of iterations of the algorithm required to find the solution of tasks of RBAC design and reconfiguration. Assessment of costs, required to solve the task of RBAC design or reconfiguration, was performed in the *Cost Evaluator* module in accordance with the expression (5).

## 5.2 Experimental results

*5.2.1 Evaluation of the operation speed.* Evaluation of the operation speed of the GAs was carried out for different dimensions of the task of RBAC administration. The dimension of the problem was determined by the pair of values *(m, n)*. We chose three categories of dimension: small (9, 30), middle (21, 100), and large (35, 500). Such approach is applied in many works, for example, in [14, 15, 18]. However, in the known works the assessment of operational speed in case of big dimensions was not carried out. Authors of these works recognized that for big data sets additional optimizers, for example, CPLEX and MIP optimizers using Branch and Cut techniques [18], as a rule, are required. For this reason, we did not carry out a comparative assessment on the operational speed of our results with other operations on the operational speed.

We were not interested in dependence of GA operation speed on crossover and mutation probabilities as well on other categories of dimension though it is very interesting subject. We leave it on future researches.

Besides the developed GAs, there was carried out the same evaluation of operation speed for the GAs without novelties, that are proposed in this paper. The testbed was able to disable these features. This assessment was done to determine the effectiveness of the proposed novelties. Among these novelties were: (1) The use of columns of Boolean matrices as the genes of chromosomes. If you disable this feature, each individual in the population had not two, but one chromosome. Each gene of this chromosome corresponds to a single element in the matrices **X** or **Y**. (2) Check for uniqueness of chromosomes of the new individuals resulting from crossover and mutation. If you disable this feature the number of unique individuals in the population decreased. Therefore the speed of the algorithm had to increase.

The evaluation variant, in which the first novelty was not used, is called V.1, the second is V.2. To refer to the basic variant, where all the novelties were used, we use the label V.0.

The results of the evaluation of operation speed of the GAs are presented in Fig. 5. For each combination of task parameters the experiments were conducted 10 times, and then we calculated the average values. At the same time the dispersion of values in statistical selection did not exceed 10 percent. Operational speed was measured by the number of iterations. This choice allows to remove the dependence of results on a computer configuration. At the same time on the Intel Xenon E5-2620 4x2 GHz Cores processor the operating time in variant V.0 was on average equal: 10 seconds for (9, 30) and 6 minutes for (35, 500) dimensions. The iteration time changed from 50 to 70 ms. The analysis of these data allows to draw the following conclusions.

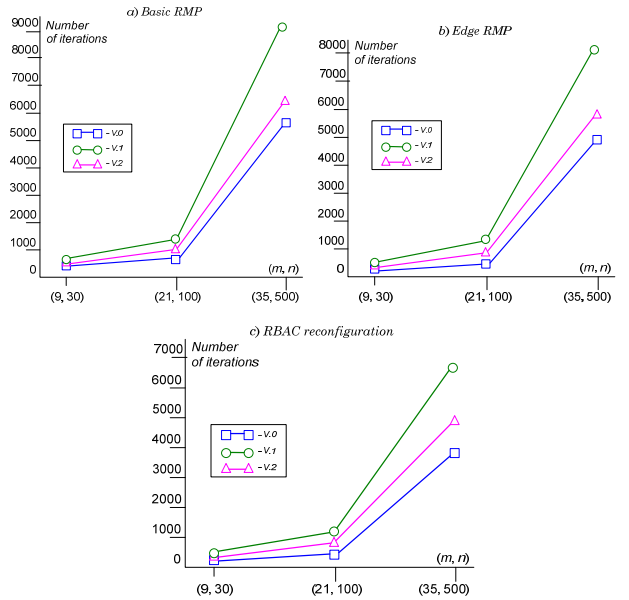


**Figure** 5: **Results estimate the rate of the GAs for different variants (V.0, V.1, and V.2) and different tasks *(a – Basic RMP, b - Edge RMP, c – RBAC reconfiguration).***

First, among the administrative tasks the most difficult is the task of RBAC design in the variant *Basic RMP*. At all dimensions it showed the highest value of number of iterations. Less difficult is the task of RBAC design in the variant *Edge RMP*. The problem of RBAC reconfiguration was solved the fastest. This is because, on the one hand, the criteria for the search of solutions in the form of expressions (1) and (2) are more difficult than the criterion with expressions (1) and (3). On the other hand, the task of reconfiguration is focused on minimal changes in the current RBAC scheme. The search of such solution is done faster than search for any of the tasks of RBAC design.

Second, in all cases the variant V.0, in which there were implemented all the proposed novelties of the GAs, was faster than the other options. The advantage at medium and large dimensions were: for option V.1 – from 63 % to 2.5 times; for option V.2 – from 13 % to 60 %. For small dimension there was no advantage in all cases. Bigger advantage with respect to V.1 indicates that the application for each individual in the developed GAs of two chromosomes $Chr_X$ and $Chr_Y$ (in which the genes are the columns of matrices **X** and $\mathbf{Y}^T$, respectively) is more effective innovation than consideration of uniqueness of new individuals. At that the second novelty is also quite effective.

Third, with increase of dimension of the problem, the running time of the GA increases. This rule is obvious. However, if in this dependence we use as argument not a pair of values *(m, n)*, but the value of $L = m \times n$, we can see that for the developed GAs (i.e. for the variant V.0) this dependence is almost linear. This suggests that the developed algorithms are quite powerful method of solving RBAC administrative tasks, which, as it was mentioned above, belong to the class of *NP*-complete tasks. Thus, the obtained experimental results on the estimation of operation speed of the proposed GAs for RBAC administration confirm our assumptions made in previous papers that implemented novelties significantly increase the operation speed of the proposed GAs compared to the algorithms in which these opportunities do not exist.

*5.2.2 Evaluation of administrative costs.* Evaluation of costs required for administrative tasks in RBAC was performed as follows. Before its beginning the matrix $\mathbf{A}_0$ was generated and using the GA the matrices **X** and **Y** were found. Then we generated the matrix $\Delta\mathbf{A}$, which determines changes in role access control. The matrix $\Delta\mathbf{A}$ allows to obtain new matrix **A** in accordance with the expression $\mathbf{A} = \mathbf{A}_0 \oplus \Delta\mathbf{A}$. When generating the matrix $\Delta\mathbf{A}$ we considered factor $\gamma$ called the *power of reconfiguration*. It is defined as $L_\Delta / L_0$,, where $L_\Delta$ is the number of unit elements in the matrix $\Delta\mathbf{A}$, and $L_0$ is the number of unit elements in the matrix $\mathbf{A}_0$. The coefficient $\gamma$ took the values 0.1 and 0.25. At larger value of $\gamma$ the reconfiguration becomes senseless. It will be necessary to solve the RBAC design problem.

Then for the matrix **A** we were searching for matrices **X** and **Y** in three ways. The first two methods represent the solution to the problem of designing an RBAC options in the *Basic RMP* and *Edge RMP* variants. At that we were taking into account the criteria (1), (2) and (3). The third method is solution of the problem of reconfiguration of RBAC. For it additionally as the initial data there was taken into account matrix $\mathbf{A}_0$ and criteria (4) and (5) were used. The value of administrative costs for all three methods was determined by the formula (5), which shows the number of elements in the matrices **X** and **Y** which need to change to go from the matrix $\mathbf{A}_0$ to the matrix **A**.

Then we regenerated the new matrix $\Delta\mathbf{A}$. Again the matrices **X** and **Y** were found in three ways. For each of these methods the administrative costs were calculated. This iteration in the course of the experiments was repeated 10 times. The results of experiments are shown in Fig. 6. More iterations were not considered, as it was believed that the RBAC system administrator with ten changes of the RBAC schema is guaranteed to at least once solve the problem of RBAC design.

The analysis of data presented in Fig. 6, allows to draw the following conclusions. Administrative costs for the task of RBAC reconfiguration (with increasing number of iterations of experimental evaluation) practically does not change. The observable spread is comparable with the statistical error.

At the same time, for the tasks of RBAC design in the variants *Basic RMP* and *Edge RMP* these costs increase with the number of iterations. This demonstrates that when carrying out a series of access control policy changes the administrative costs, necessary to bring the RBAC scheme in the optimum condition according to the criteria of *Basic RMP* and *Edge RMP,* are accumulated. The results shown in Fig.6, provide the RBAC system administrator with system tools for selecting the strategy to deal with changing access control policies. Knowing how much time it will be required to implement one change in the RBAC scheme, the administrator can estimate the time required for the transition from the matrix $\mathbf{A}_0$ to the matrix **A** in various ways.

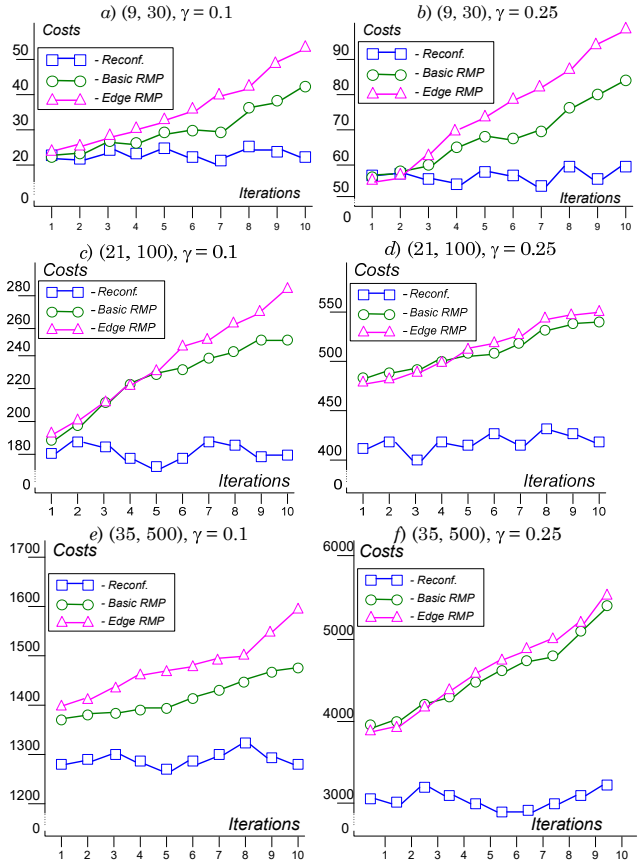


**Figure 6: The results of the assessment of the administrative costs for different tasks and different dimensions of tasks (*a* – (9, 30), γ = 0.1; *b* – (9, 30), γ = 0.25; *c* – (21, 100), γ = 0.1; *d* – (21, 110), γ = 0.25; *e* – (35, 500), γ = 0.1; *f* – (35, 500), γ = 0.25).**

As a result, he (she) can in on-line mode to perform this transition by solving the problem of RBAC reconfiguration, and in more convenient time in the off-line mode to perform this transition by solving the *Basic RMP* or *Edge RMP*. However, the longer he (she) postpones the tasks of *Basic RMP* or *Edge RMP*, the more time he (she) will need to implement their solving.

From Fig. 6 we see that the larger the dimension of the problem is, the more increases the difference between the cost of the design and reconfiguration with increasing number of iterations. This difference also depends on the ratio γ, the greater it is, the greater the difference is. While at low dimension the difference is only 30 units in γ = 0.1 and about 50 units with γ = 0.25, at high dimension this difference has the values 200 and 2500, respectively.

From Fig. 6 we see that the task of RBAC design according to the criterion of *Edge RMP* requires larger number of administrative costs than the task of *Basic RMP*. This is because in the task of *Edge RMP* the optimization criterion requires minimum number of connections in the matrices **X** and **Y**.

Consequently, the divergence between matrices **A** and $\mathbf{A}_0$ in this case will be greater than at the criteria *Basic RMP,* requiring a minimum number of roles. Thus, the obtained experimental results on the assessment of costs to administer RBAC in conditions of multiple changing of access control policy showed that the cost of implementing solution to the task of RBAC reconfiguration in these conditions practically does not change. At the same time, the cost of implementing a solution to the problem of RBAC design continuously increases with increasing number of changes in the access control policy. Consequently, the RBAC administrator when selecting his actions that are necessary for the transition from the old to the new access control policy should take into account the results of this evaluation.

# 6 CONCLUSIONS

The paper presents a new approach to solving the problem of administration of RBAC systems using GAs. This problem is to implement a meaningful choice among the tasks of RBAC design and reconfiguration under conditions of repeatedly changing access control policies.

As metrics to justify the selection we used the value of administrative costs, expressed as a number of changes that need to be done during the transition from the old to the new policy. To solve the problem of RBAC administration we developed the testbed that allows us to estimate the amount of administrative costs required to implement the found solutions in the RBAC scheme, as well as to estimate the operating speed of the proposed GAs.

Basing on the results of our previous papers we formed the final part of the novelties, implementation of which in the GAs allows to significantly increase their speed in solving the task of administration of RBAC systems. Among these novelties are: (1) Forming for each individual of a population of two chromosomes encoding the Boolean matrices **X** and **Y** which are variables in the optimization problems of RBAC administration. (2) Usage as genes of chromosomes of not isolated elements of the matrices **X** and **Y**, but their columns.

The additional procedure for synchronization of descendants was developed to increase the crossover efficiency. Besides, additional procedures were developed for mutation and selection. They select elements for inversion and control of the uniqueness of chromosomes for new individuals, respectively.

Evaluation of the operating speed of the developed GAs showed that the usage of these novelties allows us to get advantages compared to the traditional GAs from 13% to 2.5 times at different configurations of solved tasks.

Further research directions are associated with the transfer of the proposed GAs to new domains of access control and study of the possibility to use other bio-inspired algorithms for design and reconfiguration of access control schemes.

# ACKNOWLEDGMENTS

This work was done by the grant of RSF #15-11-30029 in SPIIRAS.

# REFERENCES

[1] R.S.Sandhu, E.J.Coyne, H.L.Feinstein, and C.E.Youman. 1996. Role-Based Access Control Models. *Computer* 29, 2 (1996), 38–47.

[2] M.Frank, J.M.Buhmann, and D.Basin. 2010. On the definition of role mining. In *Proceedings of the 15th ACM symposium on Access control models and technologies* (SACMAT '10). ACM, New York, NY, 35–44.

[3] G.Verma, V.Verma. 2012. Role and Applications of Genetic Algorithm in Data Mining. *International Journal of Computer Applications*, 48, 17 (2012) 5–8.

[4] J.Vaidya, V.Atluri, and Q.Guo. 2007. The role mining problem: finding a minimal descriptive set of roles. In *Proceedings of the 12th ACM symposium on Access control models and technologies*. ACM, New York, NY, 175-184.

[5] I.Saenko and I.Kotenko. 2011. Genetic Algorithms for Role Mining Problem. In *Proceedings of the 2011 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing* (PDP'11). IEEE Computer Society, Washington, DC, 646–650.

[6] I.Saenko and I.Kotenko. 2012. Design and Performance Evaluation of Improved Genetic Algorithm for Role Mining Problem. In *Proceedings of the 2012 20th Euromicro International Conference on Parallel, Distributed and Network-based Processing* (PDP'12). IEEE Computer Society, Washington, DC, 269–274.

[7] I.Saenko and I.Kotenko. 2015. Improved genetic algorithms for solving the optimisation tasks for design of access control schemes in computer networks. *Int. J. Bio-Inspired Comput.*7, 2 (May 2015), 98–110.

[8] I.Saenko and I.Kotenko. 2017. Reconfiguration of RBAC schemes by genetic algorithms. In Studies in Computational Intelligence, Vol. 678, Springer International Publishing (2017), 89–98.

[9] M.Kuhlmann, D.Shohat, and G.Schimpf. 2003. Role mining - revealing business roles for security administration using data mining technology. In *Proceedings of the eighth ACM symposium on Access* control models and technologies (SACMAT '03). ACM, New York, NY, USA, 179–186.

[10] J.Vaidya, V.Atluri, and J.Warner. 2006. RoleMiner: mining roles using subset enumeration. In *Proceedings of the 13th ACM conference on Computer and communications security* (CCS '06). ACM, New York, NY, 144–153.

[11] C.Blundo and S.Cimato. 2010. A simple role mining algorithm. In *Proceedings of the 2010 ACM Symposium on Applied Computing* (SAC'10). ACM, New York, NY, USA, 1958–1962.

[12] A.Colantonio, R.D.Pietro, A.Ocello, N.V.Verde. 2009. A Probabilistic Bound on the Basic Role Mining Problem and its Applications. In *Emerging Challenges for Security, Privacy and Trust*. IFIP Advances in Information and Communication Technology, Vol. 297, 376–386.

[13] M.Frank, J.M.Buhman, and D.Basin. 2013. Role Mining with Probabilistic Models. *ACM Trans. Inf. Syst. Secur.* 15, 4, Article 15 (April 2013), 28 p.

[14] M.Frank, A.P.Streich, D.Basin, and J.M.Buhmann. 2012. Multi-assignment clustering for boolean data. *J. Mach. Learn. Res.* 13, 1 (2012), 459–489.

[15] A.Colantonio, R.D.Pietro, and A.Ocello. 2008. A cost-driven approach to role engineering. In *Proceedings of the 2008 ACM symposium on Applied computing* (SAC '08). ACM, New York, NY, 2129–2136.

[16] I.Saenko and I.Kotenko. 2016. Using Genetic Algorithms for Design and Reconfiguration of RBAC Schemes. In *Proceedings of the 1st International Workshop on AI for Privacy and Security* (PrAISe '16). ACM, NY, Art.4, 9 p.

[17] M.Jafari, A.H.Chinaei, K.Barker, M.Fathian. 2009. Role Mining in Access History Logs. *International Journal of Computer Information Systems and Industrial Management Applications*, Vol.1, No.1, 258–265.

[18] E.Uzun, V.Atluri, H.Lu, and J.Vaidya. 2011. An optimization model for the extended role mining problem. In *Proceedings of the 25th annual IFIP WG 11.3 conference on Data and applications security and privacy* (DBSec'11), Yingjiu Li (Ed.). Springer-Verlag, Berlin, Heidelberg, 76–89.

[19] C.Blundo and S.Cimato. 2013. Constrained Role Mining. In *Security and Trust Management*, Lecture Notes in Computer Science, Vol. 7783, Springer-Verlag, Berlin, 289–304.

[20] H.Xia, M.Dawande, V.Mookerjee. 2014. Role Refinement in Access Control: Model and Analysis. *INFORMS J. on Computing* 26, 4 (2014), 866-884.

[21] N.Hu, P.G.Bradford, and Jun Liu. 2006. Applying role based access control and genetic algorithms to insider threat detection. In *Proceedings of the 44th annual Southeast regional conference* (ACM-SE 44). ACM, New York, NY, 790–791.

[22] N.Semmanche and S.Selka. 2008. Access control of Web services using genetic algorithms. In *Proceedings of the 2008 High Performance Computing & Simulation Conference* (HPCS'08), ECMS, Nicosia, Cyprus, 249–254.

[23] H.-S.Yang, M.Maier, M.Reisslein, and W.M.Carlyle. 2003. A Genetic Algorithm based Methodology for Optimizing Multi-Service Convergence in a Metro WDM Network. *Journal of Lightwave Technology*, 21, 5, 1114–1146.

[24] N.Rai and K.Rai. 2014. Genetic Algorithm Based Intrusion Detection System. *International Journal of Computer Science and Information Technologies*, 5, 4 (2014), 4952–4957.

[25] R.Mueller-Bady, R.Gad, M.Kappes, and I.Medina-Bulo. 2015. Using Genetic Algorithms for Deadline-Constrained Monitor Selection in Dynamic Computer Networks. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*. ACM, NY, 867–874.