Distributed corporate information systems optimization based on genetic algorithm and object model

S Y Zemlyanskaya¹, O V Chengar², E O Savkova¹ and O A Shumaieva¹

¹ Donetsk National Technical University, 58, Artyoma str., 83001, Donetsk
² Sevastopol State University, 33, Universitetskaya str., Sevastopol, 299053, Russia

E-mail: ovchegar@sevsu.ru

Abstract. In paper object model of distributed corporate information system is developed in the form of interacting models of its typical components, which takes into account replication and distribution of data between nodes of computer network, characteristics of hardware and makes it possible to simulating DCIS of any structure. A new modification of the genetic algorithm for the DCIS optimization problem has been developed, in which combined multichromosomes have been used to represent the placement of database tables, applications and parameters of servers, network devices and communication channels, genetic operators of recombination, crossover and mutation have been developed. To optimize the DCIS functioning, together with a modified genetic algorithm, the object model is used, that calculates the fitness-function, forming the optimal configuration of DCIS data distribution and hardware parameters in real time for the performance criterion formulated as the minimization of the system response time. A serial scheme of implementation of the genetic algorithm was developed, which includes full or partial optimization of DCIS in accordance with the requirements. The approach proposed allows obtaining suboptimal solution to the DCIS optimization problem with deviation from the global optimum of no more than 5%.

1. Introduction

Distributed Corporate Information System (DCIS) is one of the complex types of computer systems that are created for large industrial enterprises and various corporations to organize information exchange about the results of activities and, most importantly, for rapid decision-making. Large modern corporations bring together several enterprises (organizations) that can be geographically located in different cities or regions of the country, i.e. are distributed. Communication between computers in such systems was provided by local area networks and wide area networks, and data was stored in distributed databases.

The design and implementation (or modernization) of DCIS requires considerable costs, time and labor of teams of qualified specialists. Problems arise, on the one hand, with the limited funding of the DCIS being created (or modified) and, on the other hand, with the need to ensure the effective functioning of the DCIS.

In previous studies, models were mainly created for simple computer systems built on local networks, or a number of assumptions were applied to more complex systems model. Traditional methods of operational research were used to optimize such systems: linear, dynamic and discrete programming, and various modifications of these methods [1-5]. Application of such methods to optimize DCIS is limited by large scale, heterogeneity of DCIS structure, as well as computational complexity of
calculations. Prior art or simulation models cannot be used to model DCIS because of their complexity, cumbersome nature or the existence of a number of assumptions that reduce the type of results obtained from such models.

The selection of the optimal or "close" to the optimal system structure can be carried out using one of the approaches of the scientific direction of Natural Computing, which is based on the principles of natural decision-making mechanisms and includes neural network calculations, greedy algorithms, ant algorithms, tabulated-search, genetic algorithms and others [4].

Ant and greedy algorithms have proved to be good at optimizing systems characterized by a set of uniform parameters (the DCIS is described by a large number of unrelated parameters) or if it is necessary to choose the optimal sequence of actions, while in the DCIS all events are random.

Tabu Search is a variation of the known memory gradient descent method. During the search, a list of tabulated positions from among those already calculated is maintained. The critical parameters of algorithm are the range of prohibitions. During the search, operations are performed to include the list of prohibited states around the current state, which adds an accident factor to the search process. For DCIS optimizing this method is difficult to implement due to the need to determine multiple states around the current state, which is characterized by a large number of unrelated parameters.

Genetic algorithms, using the analogy between natural selection and process of the best solution choice among the many possible, are one of the most common options for implementing evolutionary calculations. By simulating the selection of the best variants as a process of evolution in a population of individuals, one can obtain a solution to the optimization problem by setting the initial conditions of the evolutionary process. Complexity of coding scheme selection, i.e. selection of parameters and their coding type in "chromosomes", possibility of population degeneration, and complexity of scheduling constraints description create problems of genetic algorithms implementation. The advantage of genetic algorithms over others is the simplicity of their implementation, the relatively high speed of operation, the parallel search for a solution by several individuals at once, which allows avoiding falling into the trap of local optimums.

Thus, the task of the DCIS optimal parameters at limited financial cost in order to achieve maximum system productivity requires further research. This task can be achieved by using modern modeling and optimization methods [4-6].

The purpose of paper consists in improving the DCIS performance by optimizing the data and query processing applications distribution between computer information system nodes, as well as optimizing server, link and network device parameters.

2. Formulation of the DCIS effectiveness criterion

The main purpose of the DCIS development and deployment is to enable rapid management decision-making on the basis of data on the state of affairs in the corporation (taking into account the current financial and production situation), access of users to the necessary data and the possibility of rapid and safe exchange of data within a certain corporate organization. Thus, one of the main requirements for the DCIS is to provide the fastest response to a user's request. Therefore, the performance criterion of the DCIS can be represented as the response time of the system to user requests.

Let us consider how the reaction time of the system is determined. Requests are typically made by users on workstations and then transmitted over communication links to processing nodes, which may include a database server and an application server for processing complex requests, after which the response over the communication links is transmitted to the user. Then, the execution time $T_{resp}$ of a user request is determined as follows:

$$T_{resp} = T_t + T_p + T_w + \max_{i=1, N_{sq}} \{T_{sq_i}\}$$

where $T_t$ is the request transmission time, $T_p$ is the request processing time, $T_w$ is the request transmission and processing delay time, $N_{sq}$ is the quantity of communication channels between source and handler of i-th subquery, $T_{sq_i}$ is the i-th subquery execution time.
\[ T_{sqi} = T_{ti} + T_{pi} + T_{wi} \]  \hspace{1cm} (2)

where \( T_{pi} \) is the i-th subquery time, \( T_{wi} \) is the i-th subquery transmission and processing delay time, \( T_{ti} \) is the i-th subquery transmission time:

\[ T_{ti} = \sum_{j=1}^{N_{cq_i}} \frac{(V_{qi} + V_{ai})}{S_j} \]  \hspace{1cm} (3)

where \( V_{qi} \) is the i-th subquery size (bytes), \( V_{ai} \) is the i-th subquery response size (bytes), \( S_j \) is the capacity of j-th communication channel (Kbps);

\[ T_{pi} = T_{pa_i} + T_{pdb_i} \]  \hspace{1cm} (4)

where \( T_{pa_i} \) is the time of execution of i-th subquery by application server, \( T_{pdb_i} \) is the time of i-th subquery execution by database server:

\[ T_{pa_i} = \frac{C_{ai}}{P_{sa_i}} + \max_{j=1,N_{sq_i}} \{ T_{sqj} \} \]  \hspace{1cm} (5)

where \( C_{ai} \) is the complexity of the i-th subquery when processing the subquery by the server (JOPS), \( P_{sa_i} \) is the performance of application server processing i-th subquery (JOPS/sec);

\[ T_{pdb_i} = \frac{C_{dbi}}{P_{sd_{bi}}} + \max_{j=1,N_{sq_i}} \{ T_{sqi} \} \]  \hspace{1cm} (6)

where \( C_{dbi} \) is the complexity of i-th subquery in subquery execution by database server (TPC), \( P_{sd_{bi}} \) is the performance of the database server processing the i-th subquery (TPC/sec), \( N_{sq_i} \) is the quantity of subqueries of the query, \( N_{sq_i} \) is a quantity of subqueries of i-th subquery,

\[ T_{wi} = T_{wti} + T_{wp_i} \]  \hspace{1cm} (7)

where \( T_{wti} \) is the transmission delay time of the i-th subquery communication channels (sec), \( T_{wp_i} \) is the waiting time in the server request queue.

When evaluating the performance of a computer system by reaction rate, two types of criteria are used: weighted average and threshold. The weighted average criterion is the sum of the response times to all user requests of the corporate system that require system response (read requests), that is, the sum of the view

\[ T_{r_A} = \sum_{i=1}^{N_q} \frac{T_{resp_i}}{N_q} \]  \hspace{1cm} (8)

where \( T_{resp_i} \) is the system response time to i-th user request, \( N_q \) is the quantity of different user queries in the system.

Threshold criteria corresponds to the worst response time of the system to a user request:

\[ T_{r_L} = \max_{i=1,N_q} \{ T_{resp_i} \} \]  \hspace{1cm} (9)
The optimization threshold criterion (TrL) ensures that all users have some satisfactory level of response to their requests. The weighted average criterion (TrA) may discriminate against some users for whom the reaction time is too long, while the average result is quite acceptable.

Thus, we select threshold criterion (3) as the criterion for optimality of the DCIS.

3. DCIS optimization problem statement

Let the distributed corporate information system be set with the following parameters:

\[ \{MN_n\} \] is a set of DCIS nodes; \( n = 1, \bar{N}n \), where \( Nn \) is the quantity of DCIS nodes;

\[ \{MSdb_i\} \] is a set of data base servers; \( i = 1, \bar{N}db \), where \( Ndb \) is the quantity of data base servers;

\[ \{MSa_j\} \] is a set of application servers in DCIS nodes; \( j = 1, Nsa \), where \( Nsa \) is the quantity of application servers;

\[ \{MWs_k\} \] is a set of workstations in DCIS nodes; \( i = 1, \bar{N}ws \), where \( Nws \) is the quantity of workstations;

\[ \{MC_l\} \] is a set of channels between DCIS nodes; \( l = 1, \bar{N}c \), where \( Nc \) the quantity of channels;

\[ \{MNd_d\} \] is a set of network devices in DCIS nodes, \( n = 1, \bar{N}nd \), where \( Nnd \) is a quantity of network devices;

\[ \{MDs_d\} \] is a set of datasets; \( m = 1, \bar{N}ds \), where \( Nds \) the quantity of datasets;

\[ \{MA_j\} \] is a set of applications; \( p = 1, \bar{Na} \), where \( Na \) the quantity of application;

\[ \{MQ_q\} \] is a set of queries, \( q = 1, \bar{N}q \), where \( Nq \) is the quantity of queries in DCIS.

The structure of the DCIS (territorial location of servers, workstations in the DCIS nodes and communication channels between them) is given as follows:

a) relation between nodes and their workstations, servers and network devices, i.e. between sets \( \{MSa\}, \{MSdb\}, \{MWs\}, \{MNd\} \subset \{MN\} ; \)

\[ \forall MSa_j \in \{MSa_j \mid j = 1, Nsa\} \exists MN_{nsa} \in \{MN_n \mid n = 1, Nn\} \]

\[ \forall MSa_j \in \{MSa_j \mid j = 1, Nsa\} \exists MN_{ndd} \in \{MN_{nd} \mid d = 1, Nnd\} \]

\[ \forall MSdb_i \in \{MSdb_i \mid i = 1, Ndb\} \exists MN_{ndb} \in \{MN_{n} \mid n = 1, Nn\} \]

\[ \forall MSdb_i \in \{MSdb_i \mid i = 1, Ndb\} \exists MN_{ndd} \in \{MN_{d} \mid d = 1, Nnd\} \]

\[ \forall MWs_k \in \{MWs_k \mid k = 1, Nws\} \exists MN_{nws} \in \{MN_{n} \mid n = 1, Nn\} \]

\[ \forall MWs_k \in \{MWs_k \mid k = 1, Nws\} \exists MN_{nda} \in \{MN_{d} \mid d = 1, Nnd\} \]

\[ \forall MNd_d \in \{MNd_d \mid d = 1, Nnd\} \exists MN_{ndd} \in \{MN_{n} \mid n = 1, Nn\} \]

\[ \forall MNd_d \in \{MNd_d \mid j = 1, Nnd\} \exists MC_{ld} \in \{MC_l \mid l = 1, Nc\} \]

b) relation between workstations of DCIS and queries from workstations, i.e. between sets \( \{MQ\} \subset \{MWs\} ; \)

\[ \forall MQ_q \in \{MQ_q \} \exists MWs_{kq} \in \{MWs_k \mid k = 1, Nws\} \]

Next parameters are in addition set:

\[ \{MTbkd\} \] is a set of DB servers types; \( r = 1, \bar{N}tsd \), where \( Ntsd \) is the quantity of DB servers types;

\[ \{MTA\} \] is a set of application servers types; \( s = 1, \bar{N}tsa \), where \( Ntsa \) is the quantity of application servers types;

\[ \{MCT\} \] is a set of communication channels types; \( t = 1, \bar{N}tc \), where \( Ntc \) is the quantity of communication channels types.

\[ \{MTN\} \] is a set of net devices types; \( u = 1, \bar{Nm} \), where \( Nm \) is the quantity of net devices types.

Configuration of DCIS is determined by ratios (4) and (5):
1) For each node, the types of database servers, application servers, network devices, and communication channels are set such that the correspondence between the sets is ensured:

\[ \{MSdb\} \text{ and } \{MTbd_i\}, \text{ where} \]

\[ \forall MSdb \in \{MSa_i \mid j = 1, Ndb\} \exists MTbd_i \in \{MTdb_i \mid r = 1, Ndb\} \]

\[ \{MSa_i\} \text{ and } \{MTA_j\}, \exists MSa_j \in \{MSa_j \mid j = 1, Nsa\} \exists MTA_j \in \{MTA_j \mid j = 1, Nta\} \]

\[ \{MC_i\} \text{ and } \{MTC_i\}, \exists MC_i \in \{MC_i \mid l = 1, Nc\} \exists MTC_i \in \{MTC_i \mid t = 1, Nt\} \]

\[ \{MNd_a\} \text{ and } \{MTN_a\}, \exists MNd_a \in \{MC_i \mid n = 1, Nnd\} \exists MTN_a \in \{MTC_a \mid u = 1, Nm\} \]

So only one of the elements of the second set of the corresponding pair is matched to each element of the first set.

2) Data sets location on database servers and applications location on application servers is specified, that is, a subset of system database servers \(\{MLD_j \mid d = 1, mi\}\) is defined and a subset of application servers \(\{MLA_a \mid a = 1, pj\}\) is defined, so there is a relation between sets:

\[ \{MDs_m\} \text{ and } \{MSdb\}, \]  
\[ \forall MDs_m \in \{MDs_m \mid m = 1, Nds\} \exists \{MLD_j \mid d = 1, mi\}, \exists \{MSdb_i \mid i = 1, Ndb\}, 1 \leq mi \leq Ndb \]

\[ \{MA_p\} \text{ and } \{MSa_i\}, \]  
\[ \forall MA_p \in \{MA_p \mid p = 1, Na\} \exists \{MLA_a \mid a = 1, pj\}, \exists \{MSa_j \mid j = 1, Nsa\}, 1 \leq pj \leq Nsa \]

So at least one element of the second set is matched to each element of the first set, but less than \(Nsa\) or \(Ndb\).

\(\{DCIS_p \mid p = 1, NP\}\) is multiple options of DCIS configurations, where \(NP\) is the quantity of DCIS configurations. We need to find such a DCIS configuration named \(DCIS\), that has criterion of efficiency (3) with minimum value:

\[ T_{opt} = \min_{\{DCIS_p \mid p = 1, NP\}} \{\max_{i = 1, Nq} \{Tresp_i\}\} \]

where \(Nq\) is the quantity of users queries processed in the system.

When configuring DCIS it is necessary to take into account a number of limitations imposed by the data and applications used in the system, as well as the financial capabilities of the organization within which the DCIS is operated or designed:

1. Limit on data placement. The amount of disk space placed in this DB server node must be sufficient to accommodate the required datasets:

\[ \sum_{j=1}^{Nds} Vds_j \leq Vd_i \quad \forall MSdb_i \in \{MSdb_i \mid i = 1, Ndb\} \]

where \(Ndb\) is the quantity of DB servers, \(Vd_i\) – i-th server disk space, \(Nds_i\) is the quantity of datasets, placed in i-th server, \(Vds_j\) – size of \(j\)-th dataset.

2. Each dataset must be hosted on at least one database server

\[ \forall MDs_j \in \{MDs_j \mid j = 1, Nds\} \exists MSdb_i \in \{MSdb_i \mid i = 1, Ndb\} \]

3. The amount of RAM in the application servers must be sufficient to run the applications on it:
\[ \sum_{j=1}^{Nsa} Va_j + Vos_j \leq Vram_i \quad \forall MSa_j \in \{MSa_j [j = 1, Nsa]\} \]

where \( Nsa \) is the quantity of application servers, \( Vram_i \) is the amount of RAM of the \( i \)-th application server, \( Na_i \) is the quantity of applications, to run on it, \( Va_j \) is the size of \( j \)-th application, \( Vos_i \) is the amount of RAM of the \( i \)-th application server, which is occupied by the operating system.

4. The cost of hardware and hardware support of the system shall not exceed the amount allocated by the organization for the purchase of hardware:

\[ \sum_{i=1}^{Nsa} Ssa_i + \sum_{j=1}^{Ndb} Ssdb_j + \sum_{k=1}^{Nws} Sws_k + \sum_{l=1}^{Nnd} Snd_l + \sum_{m=1}^{Nc} Sc_m \leq Sts, \]

where \( Nsa \) is the quantity of application servers, \( Ndb \) is the quantity of DB servers, \( Nws \) is the quantity of workstations, \( Nnd \) is the quantity of net devices, \( Ssa_i \) is the quantity of communication channels, \( Ssa_i \) – the cost of \( i \)-th application server, \( Ssdb_j \) – the cost of \( i \)-th DB server, \( Sws_k \) – the cost of \( i \)-th workstation, \( Snd_l \) – the cost of \( i \)-th network device, \( Sc_m \) – the cost of \( i \)-th communication channel, \( Sts \) – amount of funds allocated.

4. Proposed approach to the DCIS optimization

The above-mentioned task of the optimal DCIS configuration building refers to NP-complete tasks [2]. This problem cannot be solved by classical analytical methods due to the large number and heterogeneity of input parameters, as well as the inability to accurately determine the value of the objective function due to the randomness of processes taking place in the system.

Therefore, in order to optimize the parameters of the DCIS, it is proposed to use a new approach based on the joint use of the object model of the DCIS [8] and the apparatus of genetic algorithms (GA) [9], the essence of which is as follows. The initial version of the DCIS is formed on the basis of information on its structure. The parameters of the individual components of the DCIS (server types, workstations, communication channels, and the placement of datasets on database and application servers on application servers) are encoded as GA chromosomes. GA population is a set of some search space points.

In the process of optimization chromosomes (various variants of system configuration) are generated using GA operators. The resulting schemas are the source information for the object model. Simulation of the DCIS using this object model allows to obtain estimates of the performance criteria of the DCIS (12). These estimates, in turn, are the values of the GA fitness function for this solution. This process is repeated until the genetic algorithm shutdown criterion is reached.

An object-oriented approach has been chosen to construct the DCIS simulation model because in this model DCIS is formed as a set of objects of pre-created classes, including properties and methods, which implement interaction of the system objects. This approach makes it possible to build and modify DCIS models of any complexity.

The analysis showed that the following groups of standard components, parameters of which affect the performance of the system: technical support components (computer and network equipment), software, dates used in the DCIS. Object classes have been developed for these generic components. The DCIS model is a collection of interacting objects of developed classes. The model implements an event-driven simulation algorithm, that selects the minimum event time from the event list as the next value of the model time [8]. We get the DCIS response times (1-7) for different user requests as a result of simulation. The system performance is evaluated by the maximum query execution time (9). By changing the configuration of the system, this time can be reduced and the optimality condition (12) will be met.
5. Modification of the DCIS optimization genetic algorithm

When using a genetic algorithm to optimize DCIS, it is necessary to encode a large number of heterogeneous DCIS parameters, such as servers, network devices, communication channels, data sets. Therefore, it is not possible to use the classical genetic algorithm to determine the optimal parameters of the DCIS. Let us look at the proposed modifications to the methods of chromosome coding and implementation of genetic algorithm operators, taking into account the specifics of the problem to be solved. Information on the structure of the RCIS is proposed to be encoded as a set of 4 multichromosomes:

1. Parameters of hardware (database servers and applications, network devices) in DCIS nodes - multichromosome Xt;
2. Types of communication channels connecting system nodes - multichromosome Xc;
3. Applications placing on application servers - multichromosome Xa;
4. Datasets placing on DB servers - multichromosome Xd.

Limitations (13-16) are taken into account when forming chromosomes. If at least one of the conditions given is not met for a RCIS whose structure is chromosome-encoded, the chromosome is excluded from the population.

6. Algorithm of DCIS configuration optimization

Each multichromosome in the multichromosome representation of the DCIS structure is responsible for a certain aspect of the solution (system hardware parameters, dataset or applications placement). This allows the multi-chromosomes sets to be combined in a single solution, which is approximation of natural evolution. On the other hand, the presentation of a solution by a set of multi-chromosomes allows organizing the search of solutions in various productions. Fixing individual chromosomes results in narrowing search space, with the possibility of losing optimal solutions. It would therefore be useful to combine individual productions in the search of the optimal solution. In general, three approaches to combination of productions are possible: serial, parallel and parallel-serial [9-12].

Since DCIS devices parameters changing can significantly affect the efficiency of data and application placement on servers and the speed of queries execution, it is advisable to implement a serial approach in genetic algorithm of DCIS optimization, which consists in the sequential implementation of an evolutionary process in four stages. At each stage, one of the multi-chromosomes is modified: Xt (devices), Xc (channels), Xd (datasets) and Xa (applications). Thus, each of the modified set of Ms includes one type of multi-chromosomes:

\[ MS_1 \leftarrow \{Xt\}, \ MS_2 \leftarrow \{Xc\}, \ MS_3 \leftarrow \{Xd\}, \ MS_4 \leftarrow \{Xa\}. \]

At the first stage of the serial genetic algorithm the G0 population is used as the initial one, at the second stage - the G1 population formed after the first stage and so on. The scheme of the serial genetic algorithm implementation for DCIS optimization is shown in Figure 1.

![Figure 1. Scheme of implementation of serial genetic algorithm of DCIS optimization.](image-url)
Diagram of DCIS object model interaction with GA is shown in Figure 2.

The algorithm is terminated after all stages have passed and an individual with a best fitness function value (12) is selected from the best individuals. The parameters of this individual represent a suboptimal solution of the DCIS optimization problem [13].

![Diagram of DCIS object model interaction with GA](image)

**Figure 2.** Diagram of DCIS object model interaction with GA.

7. Conclusion

1. The task of the DCIS optimization is formulated. The maximum time of user request execution is proposed as an optimization criterion.

2. A dynamic object model of the DCIS has been developed as a hierarchy of classes describing the behavior of the main components of the DCIS and a control class implementing interaction between the DCIS objects during modeling.

3. A modified genetic algorithm has been developed for finding a suboptimal solution to the problem of DCIS optimization, in which parameters of hardware and placement of data sets and applications of the system are encoded with four multi-chromosomes, the process of evolution is simulated by genetic operators adapted to the specific problem. A serial approach to implementation of a genetic algorithm is proposed.

4. A new approach to DCIS optimization based on interaction of an object model of DCIS and a genetic algorithm is proposed. Parameters of servers, network devices, communication channels of DCIS as well as a method of arrangement of datasets and applications on servers are determined and the restrictions on data placement, applications and cost of the system are considered. Implementing the approach proposed allows creating DCIS configuration with efficiency criterion nearing the global optimum.

References

[1] Popov S V 2018 Distributed Data Processing Systems *Scientific Review. Technical Sciences* 2 16-20

[2] Kondratieva S D and Semisenko M G 2016 Modeling of complex systems: modern mathematical methods and practical aspects *Fundamental research* 5-1 43-46

[3] Arhipov V, Skobtsov V, Novoselova N, Aliushkevich V and Pavlov A 2016 Development of Event-
Driven Models for Operation Data Automation Control Theory Perspectives in Intelligent Systems. Proceedings of the 5th Computer Science. On-line Conference 2016 (CSOC2016) (Springer International Publishing Switzerland) 3 pp 403-414

[4] Skobtsov Yu A, Sekirin A I, Zemlyanskaya S Y, Skobtsov V Y, Chengar O V and Potryasaev S A 2016 Application of Object-Oriented Simulation in Evolutionary Algorithms Automation Control Theory Perspectives in Intelligent Systems. Proceedings of the 5th Computer Science. On-line Conference 2016 (CSOC2016) (Springer International Publishing Switzerland) 3 pp 453-462

[5] Luger J F 2005 Artificial Intelligence. Strategies and methods of solving complex problems 4th ed (Moscow: Williams) p 864

[6] Glamazdin E S, Novikov D A and Tsvetkov A V 2003 Corporate Program Management: Information Systems and Mathematical Models (Moscow: Company Sputnik+) p 159

[7] Haggarty R 2003 Discrete Mathematics for Programmers (K.: Technosphere) p 320

[8] Agibalov G P and Skutin A A 2001 Mathematical model and technology of development of safe corporate information systems (Tomsk State University) pp 1739-1750

[9] Averill M Low and Kelton D 2004 Imitating modeling (St Petersburg: Piter) p 846

[10] Zemlyanskaya S Y 2017 Formation of optimal configuration of distributed information system of railway transport using evolutionary algorithms Collection of scientific works of the Institute of Railway Transpor (Donetsk: DonIRT) 44 pp 29-41

[11] Bozhich V I and Lebedev V B 2001 Methods of genetic search for decisions, representable multichromosomes Perspective information technologies and intellectual systems (Taganrog: Taganrog State Radio Engineering University) pp 38-44

[12] Lazdyn S V and Zemlyanskaya S Y 2008 Development of dynamic model of computer information system Scientific works of DonNTU. Computer and Automation Series (Donetsk: DonNTU) 14(129) pp 107-116