THE GENERAL SCHEME OF THE GENETIC ALGORITHM FOR SOLVING THE TASK SCHEDULING PROBLEM FOR A MULTISTAGE SYSTEM AND ASSIGNING TIME FOR JOBS

S. A. Oleinikova[1], O. Ja. Kravets[1], I. A. Aksenov[2], O. Yu. Frantsisko[3], P. A. Rahman[4], I. V. Atlasov[5]*

[1]Voronezh state technical university, Voronezh; [2]Vladimir State University named after Alexander and Nikolay Stoletovs, Vladimir; [3]Kuban State Agrarian University named after I.T. Trubilin, Krasnodar; [4]Ufa State Petroleum Technological University; [5]Moscow University of Ministry of Internal Affairs of Russian Federation named by V.Ja. Kikot, Moscow
Russian Federation

[*] Corresponding Author, e-mail: csit@bk.ru

Abstract: The object of the research is a project containing a set of mutually dependent jobs with a random duration of execution. A special feature of the task is the different duration of jobs by different performers. The goal is to build a maintenance schedule and assign specialists to each job, minimizing the duration of the project. Genetic algorithm was chosen as the method. The article describes the general scheme of solving the problem by a genetic algorithm and details some of its stages.

Key words: mutual dependence of jobs, random duration of service, genetic algorithm, the chromosome.

1. INTRODUCTION

The problem of forming a schedule for a system designed to perform a variety of sequential-parallel jobs is considered. The duration of each individual operation is a random variable, which depends, in addition to various random factors, also on the performer. Besides, the execution of job may require one or more "types" of resources (specialists, equipment, etc.). Depending on the choice of the resource (performer, materials, etc.), the job can be performed with varying degrees of efficiency and for different times. Completion of service of the application brings a certain amount of profit, which depends on the quality of the job, the presence or absence of penalties, etc. It is necessary for each incoming request to select a service schedule and resources for each job such away as to maximize profit.

Graphically, an example of project representing a variety of jobs can be represented as follows (Figure 1).

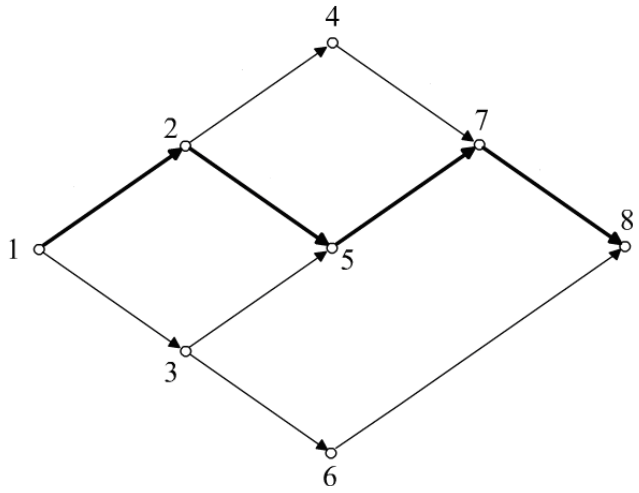

Fig. 1. System design example

2. THE PROBLEM FEATURES

Let's analyze the features of the existing problem. Its fundamental difference from analogs is the need to simultaneously solve both the task of forming a service schedule, which belongs to the class of project management tasks, and the allocation of resources by job, which is actually a task of assignments. Each of these types of problems has been separately studied in depth, and the corresponding methods of solution have been obtained.

In particular, the problem of forming a service schedule belongs to the class of project management tasks [1]. A detailed analysis of existing approaches and methods in this area is presented in [2]. Most of the algorithms for solving this problem are based on the critical path method, which allows forming a schedule of job with constant durations without taking into account resources [3]. If the duration of the job is random variable, the PERT method is used, which makes it possible to estimate these values based on the smallest, largest and most probable values [4]. There are also algorithms that make it possible to generate a schedule in conditions of resource constraints [5-9].

However, the problem under the study has a number of specific features that are not taken into account when solving this class of problems. The specificity of interchangeable, but heterogeneous resources lies in the fact that for each job, in addition to the start time, it is necessary to select the resource required for its execution.

Assigning resources to jobs or establishing a correspondence between performers and jobs is the subject of the assignment task. The classical formulation of this problem does not cause difficulties in solving. It requires determining the correspondence between n jobs and n performers. In particular, the Hungarian method is widely known and is best suited to solve it. It can be solved by any methods used to find solutions to the transport problem, since it is a special case of it. More detailed methods of its solution are presented in [10]. A generalization of the classic assignment problem is that there is no equality between the number of performers and jobs. If at the same time all sorts of restrictions are added, then the problem becomes NP-complete and requires the development of approximate methods for its solution. In particular, one of such approximation algorithms is presented in [11].

Nevertheless, the problem under consideration has specific features that are not taken into account in the above methods and algorithms. In particular, it is required to solve the problem of forming a jobs schedule together with a generalized assignment problem. It should be noted that the choice of a specialist who will be assigned to a particular job will determine not only its duration, but also the possible start time of subsequent job, because the duration of the job will depend on the choice of a specialist.

Let's give an example for a project, the graph of which is shown in Fig. 1. Let it be 5 specialists, the approximate (average) duration of each of the job of which is presented in Table 1.

Table 1. Execution time of jobs by performers

Job		per1	per2	per3	per4	per5
1	1-2	5	7	4	6	4
2	1-3	9	8	10	7	8
3	2-4	6	5	7	7	6
4	2-5	8	11	10	9	10
5	3-5	6	7	5	6	4
6	3-6	7	7	9	6	9
7	4-7	8	4	7	5	8
8	5-7	4	6	5	5	6
9	7-8	3	3	6	5	4
10	6-8	7	9	5	6	8

Each cell of this table contains the approximate duration of this job by this performer. Let us denote the corresponding matrix by T (i.e., T_{ij} is an element that shows the duration of job i performed by performer j).

The main approaches used in classical problems were as follows:

- for the preliminary job of the algorithms, it is necessary to evaluate the early and late start of all the jobs, as well as their time reserve, for which it is necessary to first find the critical path of the project and the time spent on the execution of all the jobs on the critical path;

- jobs were prioritized in accordance with the time reserve;

- at each time interval, the job with the smallest time reserve (the highest priority) was selected, which was assigned the current start time.

This process continued as long as the existing amount of resources allowed to carry out the current job. All other jobs should be postponed to a later period.

In this case, the use of such an approach is fundamentally impossible due to the absence at each planning step of data on temporary jobs reserves and, as a consequence, on their priorities.

At the same time, it should be noted that it is with such features of the problem, as a rule, arise in real production and service systems. In this regard, there is a need for a mathematical and algorithmic apparatus that allows describing and solving the problem presented above, which indicates the relevance of this study. The mathematical formulation of the problem is given in [15]. The goal is to get the maximum profit from the completion of all tasks of the project. Limitations are imposed on the project execution time.

$$\sum_{i}\left(c_i - p_i(t)\right) \rightarrow \max, \quad (1)$$

$$T_{fin\_i} \leq T_i, \quad (2)$$

where c_i – profit that will be received when the application i is executed;

- $p_i(t)$ – these are penalties for delaying an application for time t;
- T_{fin_i} – actual time of completion of application i;
- T_i – standard deadline for completion of the application i.

Let's define some of these indicators. The profit from the execution of the application i can be defined as the sum of the profits for the performance of individual jobs.

$$c_i = \sum_{j \in W_i} c_{ij}\,. \quad (3)$$

where c_{ij} – the profit received from the performance of job i by performer j.

The actual time of completion of the application is determined as follows [1]:

$$T_{fin} = \max_{j}\left(t_j^0 + t_j\right) \quad (4)$$

3. GENERAL SCHEME OF THE GENETIC ALGORITHM FOR SOLVING THE PROBLEM

We will solve this problem using genetic algorithms [12-14]. The general scheme of the genetic algorithm is identical for most of the problems that will use it for their solution. The general scheme with details of the input data and the result of job for the problem under consideration is shown in Figure 2.

We detail each of its steps. In this case, one chromosome will be an array, each element i of which describes the i-th job and stores three values: ✓ start time of job; ✓ it's duration; ✓ the specialist who will be assigned to this job.

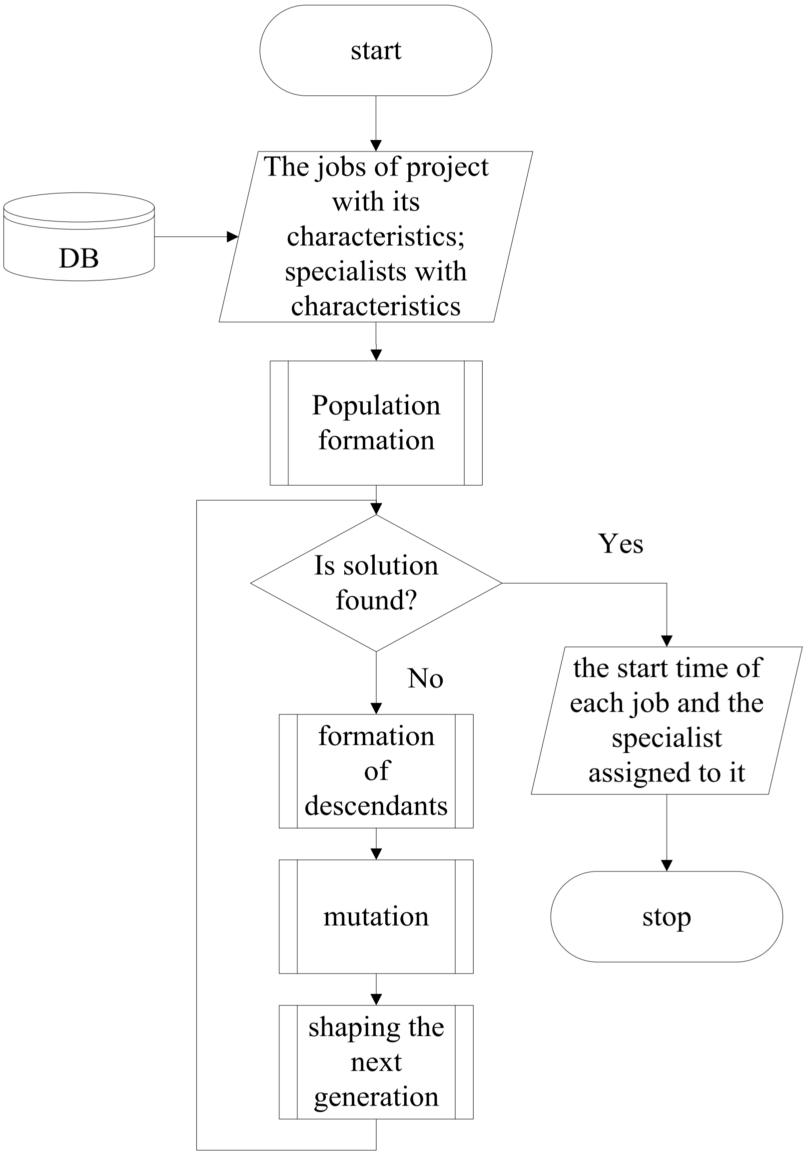

Fig. 2. General scheme of a genetic algorithm

Consider the formation of an initial population, as a result of which many chromosomes with certain genes should be obtained. As noted above, each chromosome must carry information about the beginning of the execution of each of the project's work and the corresponding performers.

The general idea of the algorithm is as follows. For each chromosome, we sequentially form genes. To do this, you must first sort the jobs in such a way that when considering the current job, all previous jobs that affect the time of its start are identified. Further, for each of the job a performer is determined. Since it is necessary to form many chromosomes (a whole generation), this choice, on the one hand, should be random. On the other hand, the probability for each chromosome of choosing a performer who will do the job faster and / or more efficiently should be higher than the probability of choosing the other performers. After assigning the performer, it is necessary to estimate the duration of this job. All these data must be entered into the chromosome gene responsible for this job, after which the transition to the next step is carried out. After the performers for all the jobs are assigned and their time characteristics are determined, the project execution time is estimated as the time of completion of the last job. If this estimate satisfies the constraints (2), then the chromosome is valid, otherwise it is discarded.

The initial population is considered to be formed if the required number of chromosomes is generated that satisfies the constraints (2).

The algorithm for the formation of the initial population is shown in Figure 3.

4. THE EXAMPLE OF THE ALGORITHM

Let's demonstrate how this algorithm works for the formation of one individual. We will use the graph shown in Fig. 1 with the time values given in Table 1.

Pre-sorting of jobs means the order in which they are selected for probabilistic assignment of start time and performer. In particular, Table 1 shows already sorted works.

We fix the start time - 0. Choose a job - this is job 1-2. We choose a performer at random. Let it be the performer 3. We write the following values into the first gene of the chromosome (Table 2).

Table 2. Assigning jobs to performers and fixing the duration at step 1

job	performer	start time	duration	end time
1-2	3	0	4	4

For convenience, we will use one more column, which will indicate the end time of the job.

To perform the job 1-3, you can choose any performer, excluding specialist number 3. Let it be the performer number 4. Then the job 1-3 will last 7 units.

No more job can be done at time 0. Consider the next moment in time - the moment of completion of job 1-2. At the moment, performer 3 is free. You can do jobs 2-4 and 2-5.

Let the job 2-4 be assigned performer 1 (then its duration will be 6 units), and the job 2-5 - performer 5 (in this case it will last 10 units).

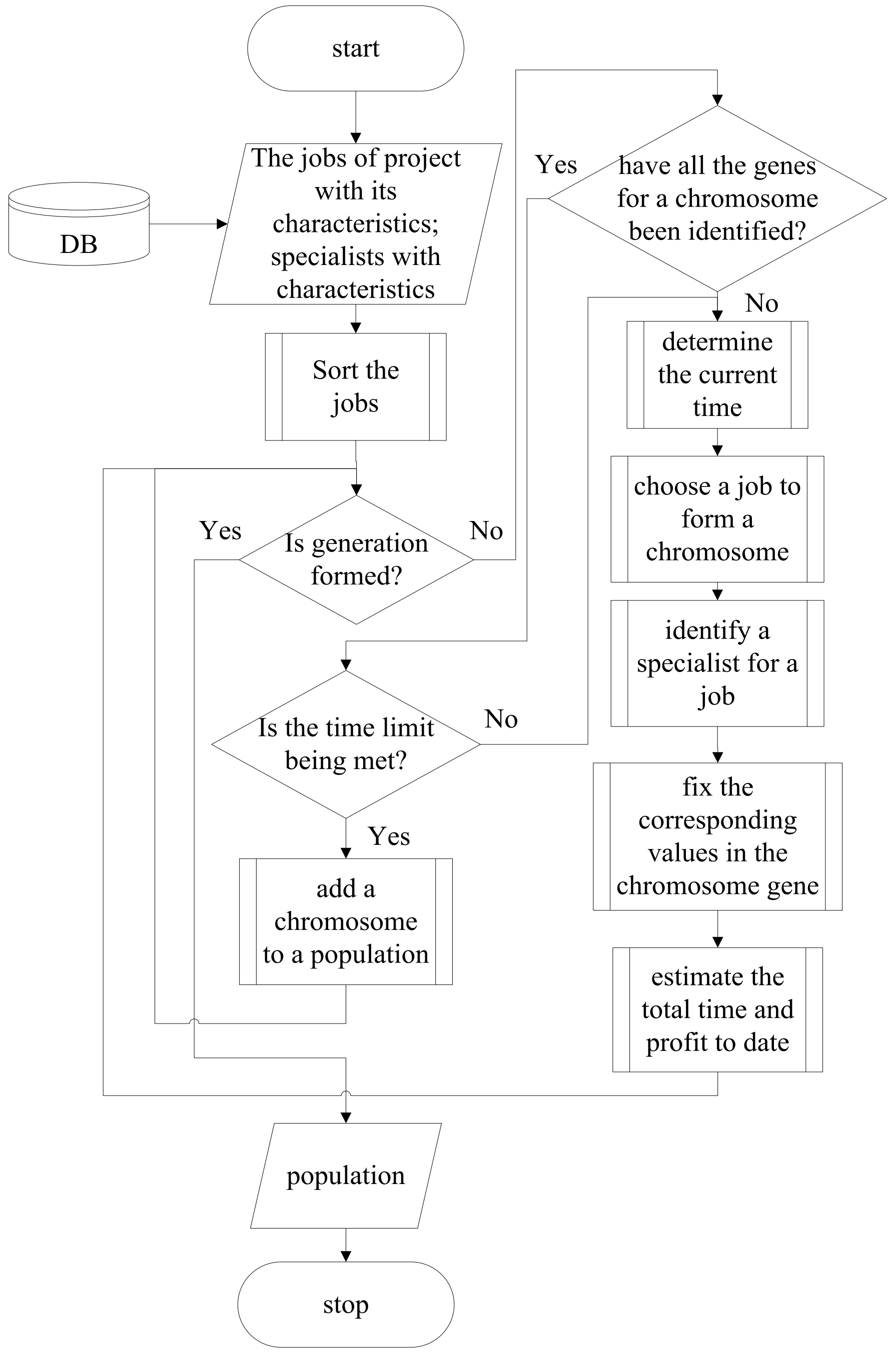

Fig. 3. Initial population formation algorithm

Let's enter the corresponding data into the Table 3.

Table 3. Assigning jobs to performers and fixing the duration at step 2

job	performer	start time	duration	end time
1-2	3	0	4	4
1-3	4	0	7	7
2-4	1	4	6	10
2-5	5	4	10	14

At time 7, the performer 4 will be free (performers 1 and 5 are still busy). It is necessary to complete the jobs 3-5 and 3-6. Let the performer 3 be assigned to the job 3-5; the performer 4 - to the job 3-6. In accordance with these values, fill in the corresponding genes of the chromosome (Table 4).

Table 4. Assigning jobs to performers and fixing the duration at step 3

job	performer	start time	duration	end time
1-2	3	0	4	4
1-3	4	0	7	7
2-4	1	4	6	10
2-5	5	4	10	14
3-5	3	7	5	12
3-6	4	7	6	13

At time 10, the job 2-4 will end and the performer 1 will be released (performers 3, 4 and 5 are still busy). Job 4-7 can be done. Let the performer 2 be chosen at random. The duration of this job will be 4. Add the corresponding gene (Table 5).

At time 12, the job 3-5 ends, and the specialist 3 is released. However, the next job 5-7 cannot be started, since the job 2-5 has not yet been completed.

Time 13. Completed the job 3-6. The specialist number 4 is released. The job 6-8 can be performed. Let it be assigned to the performer 3. Add the gene to the chromosome (Table 6).

At time 14, two jobs are finished at once: 2-5 and 4-7. Two performers are released: number 2 and number 5. You can perform the job 5-7. Let the executor 1 be assigned to it (Table 7).

Table 5. Assigning jobs to performers and fixing the duration at step 4

job	performer	start time	duration	end time
1-2	3	0	4	4
1-3	4	0	7	7
2-4	1	4	6	10
2-5	5	4	10	14
3-5	3	7	5	12
3-6	4	7	6	13
4-7	2	10	4	14

Table 6. Assigning jobs to performers and fixing the duration at step 5

job	performer	start time	duration	end time
1-2	3	0	4	4
1-3	4	0	7	7
2-4	1	4	6	10
2-5	5	4	10	14
3-5	3	7	5	12
3-6	4	7	6	13
4-7	2	10	4	14
6-8	3	13	5	18

Table 7. Assigning jobs to performers and fixing the duration at step 6

job	performer	start time	duration	end time
1-2	3	0	4	4
1-3	4	0	7	7
2-4	1	4	6	10
2-5	5	4	10	14
3-5	3	7	5	12
3-6	4	7	6	13
4-7	2	10	4	14
6-8	3	13	5	18
5-7	1	14	4	18

At time 18, two jobs are completed: 6-8 and 5-8, and all performers are released. Therefore, you can start the job 7-8 and assign any performer to it. Let the executor 2 be appointed. Then the final appearance of the chromosome will be as follows (table 8).

Table 8. Assigning works to performers and fixing the duration at step 6

job	performer	start time	duration	end time
1-2	3	0	4	4
1-3	4	0	7	7
2-4	1	4	6	10
2-5	5	4	10	14
3-5	3	7	5	12
3-6	4	7	6	13
4-7	2	10	4	14
6-8	3	13	5	18
5-7	1	14	4	18
7-8	2	18	3	21

After the formation of the chromosome, the critical path of the project and its duration are known. It is equal to 21 units.

The rest of the chromosomes are formed in a similar way.

We detail in this algorithm the probabilistic choice of performers to be assigned to job. The initial data is the job and its start time, as well as the list of free performers in this period; at the output - the identifier of the performer to whom this job will be assigned.

On the one hand, for the genetic algorithm there must be a factor of randomness in the choice of the performer. On the other hand, it is necessary to take into account the time taken to complete this job by each performer.

We propose the following scheme for identifying performers. Previously, for each row of the matrix describing the time taken to complete the job by one or another performer, we apply the formula.

$$T_{ij}^{*} = \max_{1 \le k \le M} T_{ik} - T_{ij} + 1 \qquad (5)$$

Here M is the number of performers. Then the probability of choosing job i by performer j should be calculated by the formula:

$$p_{ij}^{*} = \frac{T_{ij}^{*}}{\sum_{j=1}^{M} T_{ij}^{*}} \qquad (6)$$

Obviously, the faster a given executor performs a given job, the more likely he will be chosen for this job. Moreover, formula (6) can be applied to any other criteria for assessing the performance of the job by the performer (for example, the quality of performance, etc.).

After that, the algorithm for choosing an executor is as follows: ✓ we generate a random number α in the interval (0,1); ✓ if $\alpha < p_{i1}$, then choose executor 1; ✓ otherwise, if $p_{i1} < \alpha < p_{i1} + p_{i2}$, then choose executor 2; ✓ otherwise, if $p_{i1} + p_{i2} < \alpha < p_{i1} + p_{i2} + p_{i3}$, then choose executor 3; . . . ; ✓ otherwise, if $p_{i1} + p_{i2} + \ldots + p_{iM-1} < \alpha < 1$, then choose the executor M.

Thus, an algorithm has been developed for choosing an initial population for an approach to solving the problem of scheduling the start time of jobs and assigning performers to them.

5. CONCLUSION

The subject of research in this paper was the task of planning the start time of each of the project's jobs, a distinctive feature of which is the presence of mutual dependence between the jobs and the execution time depending on the selected performer. The following results were obtained:

1. An approach to solving the problem based on the use of genetic algorithms has been chosen.

2. A general scheme of the genetic algorithm has been developed, the specificity of chromosomes has been detailed.

3. An algorithm for the formation of the initial population is proposed, which differs in the probabilistic choice of performers depending on the time of the given job.

The proposed algorithm makes it possible to solve the optimization problem described by formulas (1) and (2) under conditions of uncertainty in the execution time of individual jobs.

REFERENCES

[1] Demeulemeester E.L., Herroelen W.S. *Project Scheduling. A research Handbook*, New York, Boston, Dordrecht, London, Moscow: Kluwer Academic Publishers, 2002, 675 p.

[2] *Project Scheduling-Recent Models, Algorithms and Applications* (J. Weglarz, ed.), Boston: Kluwer Academic Publishers, 1999, 535 p.

[3] Ahuja H.N., Dozzi S.P., Abourizk S.M. *Project Management: Technics in planning and controlling construction project* (2nd ed.), New-York: John Willey&Sons, 1994, 509p.

[4] Moder J.J, Phillips C.R., Davis E.W. *Project management with CPM, PERT and precedence diagramming*, New-York: Van-Nostrand Reinhold, 1983, 389 p.

[5] Doneva R., Gaftandzhieva S., Totkov G. FETCH Project: The Maturity of Quality Management through Dynamic Evaluation of the Project Progress. *International Journal on Information Technologies and Security*, ISSN 1313-8251, No. 3 (vol. 8), 2016, pp. 29-38.

[6] Neumann K., Schwindt C., Zimmermann J. *Project scheduling with time windows and scarce resources* (2nd ed.), Berlin: Springer, 2003, 389 p.

[7] Blanco L., Caramia M. A new approach for the Project Scheduling problem with generalized precedence relations. *Proceedings of the 11th International Workshop on Project Management and Scheduling*, Istanbul, 2008, pp. 35-39.

[8] *Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Applications* (Ed. by C. Artigues, S. Demassey, E. Neron), Wiley Online Library, 2010, 308 p.

[9] Oleynikova S.A., Kravets O.Ya., Sekerin V.D., Gorokhova A.E., Gayduk V.I. Heuristic algorithm for forming the service schedule for the multistage systems with a generalized resource criterion. *International Journal of Engineering and Technology*, Vol 7, No 4.38, 2018, pp. 370-374.

[10] Adams W.P., Johnson T.A. Improved linear programming-based lower bounds for the quadratic assignment problem. *Quadratic Assignment and Related Problems, volume 16 of DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, 1994, pp 43-75.

[11] Burkard R., Dell'Amico M., Martello S. *Assignment Problems.* Revised reprint. Philadelphia, SIAM, 2012, 386 p.

[12] Fleischer L., Goemans M.X., Mirrokni V.S., Sviridenko M. Tight approximation algorithms for maximum general assignment problems. *Mathematics of Operations Research*, Vol. 36, No 3, 2011, pp. 416-431.

[13] Mitchell M. *An Introduction to Genetic Algorithms*, MIT Press, 1996, 209 p.

[14] Umbarkar A., Sheth P.D. Crossover Operators in Genetic Algorithms: A Review. *ICTACT Journal On Soft Computing*, Vol. 6, No.1, 2015, pp. 1083-1092.

[15] Selishchev I.A., Oleinikova S.A. Mathematical Model And Algorithm For Solving The Problem Of Planning The Operation Of Multiphase Systems With Heterogeneous Resources And Time Limits. *Software Systems and Computational Methods*, No 1, 2021, pp. 35-45.

Information about the authors:

Svetlana Alexandrovna Oleinikova –Professor at Voronezh State Technical University, Faculty of Information Technologies and Computer Security. Areas of Research are Network planning and management in stochastic systems;

Oleg Jacovlevich Kravets – professor of Voronezh state technical university, areas of scientific research – system analysis, optimization, simulation of complex objects

Ilia Antonovich Aksenov – associate professor of Vladimir State University named after Alexander and Nikolay Stoletovs, areas of scientific research – application of information technologies in economic systems

Olga Yurievna Frantsisko - associate professor of Kuban State Agrarian University named after I.T. Trubilin, areas of scientific research – economic and mathematical modeling, scenario approach, information technology

Pavel Azizurovich Rahman, associate professor of Ufa State Petroleum Technological University, areas of scientific research - information technology, computer networks, reliability modeling and engineering

Igor Viktorovich Atlasov – professor of Moscow University of Ministry of Internal Affairs of Russian Federation named by V.Ja. Kikot’, areas of scientific research – system analysis, optimization, simulation of complex objects

Manuscript received on 11 August 2021