Adv Syst Sci Appl 2019; **04**; 66-78
Published online at http://ijassa.ipu.ru/index.php/ijassa/article/view/827

A Planning and Scheduling Method for Large-Scale Innovation Projects

Vladimir Topka*, Anatoliy D. Tsvirkun

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

E-mail: topka3@mail.ru, tsvirkun@ipu.ru

Received May 16, 2019; Revised December 12, 2019; Published December 31, 2019

Abstract. We consider the problem of uniform allocation according to the minimax criterion of a non-renewable resource of an innovative project's activities. A lexicographic method is used for ordering these minimax criteria To solve the problem - one objective function - developed a greedy heuristic for finding the maximum path on an acyclic digraph with double weights on arcs. The greedy heuristic gives approximate solution of the problem mentioned. The obtained solution allows building a project resources plan and schedule with four types of temporary slacks with inaccurate initial data. The proposed approach allows performing an analysis of the project Risk of Performance parameter, for which has not developed the apparatus for its analysis. These include the risks that the project when complete fails to perform as intended or fails to meet the mission or business requirements that generated the justification for the project. Performance risks are estimated on the probability distribution function of a favorable/unfavorable outcome. Proposed lexicographic disintegration is a decomposition procedure and is suitable for planning and scheduling large-scale projects. Practical calculations for the convex problem in the initial statement may be performed with the help of ready-made software, and the obtained greedy solution of the problem-consequence has theoretical significance in graph theory.

Keywords: large-scale innovative project, planning, scheduling, reliability index of project activities, greedy algorithm, the maximum path with double weights on arcs.

1. **INTRODUCTION**

In existing project management software systems, some modules perform Risk of Cost and Risk of Schedule assessment. On the other hand, an approach based on the concepts of the theory of reliability can be proposed for modeling uncertainty and risk of innovative projects.

In the theory of reliability, the indicator of network reliability is determined by the probability of its connectivity. This requires that at least one path is found connecting the beginning and the end of the network. However, in managing projects in the PERT network, in order to ensure reliable execution of the project, it is necessary to perform all activities included in the project. In this case, the task is to assess not only the connectivity of the beginning and the end of the network, but also the completeness of the whole project.

In present project management software, a quantitative assessment of Risk of Cost and Risk of Schedule is made on the basis of Monte Carlo simulation.

The proposed approach allows one the analysis of the Risk of Performance [1] of a project for which the apparatus of its analysis has not been developed. The execution risk, upon completion, accomplish the required mission - to achieve the specified technical characteristics, is based on the probability distribution function of a favorable/unfavorable outcome. For her: the probability of success of the project is the probability that the execution of this project will not fail, and the probability of reliable execution is taken as an index of the reliabil-

* Corresponding author: topka3@mail.ru

ity of implementation. The advantage of the proposed approach is that it provides a tool for working in an unexplored area.

For activities of innovative projects with a high degree of uncertainty, to obtain the characteristics of random variables of their implementation - the likelihood of achieving the goal or probability of technical success, that is, to build the distribution function of such a random variable, you can use empirical data or data obtained as a result of an active experiment in which the Monte Carlo method obtained the probability distribution function of the technical success of the activity, the degree of achievement of the specified performance characteristics of this development, in the form of a monotonously increasing with the saturation of the function of expended resources [2].

The likelihood of performing the activity by a particular time characterizes the possibility of attaining specific technical indicators of the result of the activity, provided that the previous activity, ensuring its beginning, has been completed. This probability depends on the amount of those or other resources spent, which have a value expression as cost.

The approach developed in the article is aimed at developing quantitative methods for assessing and optimizing the project parameters, taking into account the criterion of uniform allocation of a non-renewable (stored) resource with a restriction on the assessment of project reliability index.

2. INDICATORS OF THE RELIABILITY OF ACTIVITIES AND THE PROJECT AS A WHOLE

Previously, integer, linear and some non-linear models were used to describe innovative projects execution [3]. We will model [4] the dependence of the probability of technical success (reliability index) of the project's activities on the homogeneous non-renewable resource spent by 2-parameter power-concave functions of the form

$$\begin{aligned} p_j(u_j) &= \frac{u_j^{\alpha_j}}{u_j^0} \in [\varepsilon,\ 1], \varepsilon \downarrow 0, \\ u_j &\in \left[\varepsilon,\ \left(u_j^0\right)^{1/\alpha_j}\right],\ j = 1,\dots,n, \end{aligned} \tag{2.1}$$

where $0 < \alpha_j < 1$ is the form parameter, $u_j^0 > 0$ is the scale parameter. Which at the late phases of an innovation project, when it is advisable to use quantitative methods to assess the uncertainty of the project, quite visually and precisely approximates the practical [2; 5] functions of the activities reliability of the innovation project.

Let an acyclic directed graph *G* (*E*, *Γ*) be given, where *E* is the set of vertices corresponding to the project's activities (Activity-on-Node representation), and *Γ* is the set arcs defining partial order relations of direct technological activity precedence. The network *G* (*E*, *Γ*) has one final and one initial vertex; *n* - final vertex - the last number among all the vertices of the project and the dummy vertex 0, which means the launch of the project.

Thus, we have a network *G* (*E*, *Γ*), at the vertices of which a development reliability index is given that satisfies the relation (2.1). We will set the reliability of the project by the probability of its technical success (the degree of achievement of the specified performance characteristics of the project) by a certain allowable period, the assessment of the reliability of the project in the most unfavorable case. In the theory of reliability, such an assessment is determined by the weakest link, i.e. the worst of the technological chains from the vertex of the lower level to the final vertex.

$$P_n(u) = \min_{\mu_i} \prod_{(j,k)\in\mu_i} w_{jk} \frac{u_j^{\alpha_j}}{u_j^0} \in [\varepsilon,\ 1],\ \varepsilon \downarrow 0,$$
$$\alpha_j \in (0,1),\ w_{jk} \in [\varepsilon_0,\ 1],\ u_j^0 > 0;\ \forall \mu_i, \tag{2.2}$$

where $w_{jk} \in [\varepsilon_0,\ 1],\ \varepsilon_0 > 0$ is arc (*j, k*) reliability coefficient, (from the matrix of paths along the arcs of the network *G* (*E, Г*)) transfer the result of the *j*-th activity by the arc (*j, k*) to perform the *k*-th activity (like the transfer function of connecting links in theory of automatic control). We denote μ_i the path with the number $i = 1,...,m$ on the network from the initial vertex to the final vertex *n.*

When solving problems of project planning in terms of a power model [6], the time variable is not taken into account. In this article we will consider a deterministic network model of the project with disjunctive input arcs OR, or with conjunctive input arcs AND having the property that for the project reliability indicator is performed assessment of lower bound (2) [6]. Thus, we will speak about p_i from (2.1) as a reliability index of activity, and about P_n from (2.2) - as an assessment of indicator of guaranteed reliability of project performance. In the model for representing network constraints, we will use the matrix of paths (along arcs) of the network *G* (*E, Г*): which is based on the list of arcs of the network as the characteristic function of the constraint system. In the network model of the project *G* (*E, Г*), each path from the initial vertex to the final one is uniquely represented by a line of a specially constructed matrix of paths (along arcs) $\left(l_{j,k}\right)$ $\left(j,k\right) \in \mu_i$, in which its elements – arcs stand in the *i*-th line only if (*j, k*) - arc belongs to *i* -th path. All other cases when $\left(j,k\right) \notin \mu_i$ there are empty cells.

3. THE PROBLEM OF SUCCESSIVE UNIFORM ALLOCATION OF NON-RENEWABLE RESOURCE BY CHEBYSHEV CRITERION

Let u=$\left(u_1,\ u_2,\ ...,\ u_n\right)$ − be the vector of a non-renewable resource allocated for the execution of all activities of the project *G* (*E, Г*). As an objective function for the allocation of resources, we will use uniform, or Chebyshev criterion in the form

$$\max_{j\in E} u_j \to \inf_{\mathrm{u}\in U},\ u_j \geq 0, \quad j = 1,...,n. \tag{3.1}$$

A problem of the form (3.1) is usually called a discrete minimax problem. Therefore, this discrete minimax problem is considered in the form of the following problem of smooth conditional minimization

$$u^0 \to \inf_{(u^0, u_j)\in U},$$

$$U = \begin{cases} u_j \leq u^0, \quad \forall j = 1,...,n, \\ \prod_{(j,k)\in\mu_i} w_{jk} \dfrac{u_j^{\alpha_j}}{u_j^0} \geq p_0 \in [\varepsilon,\ 1], \quad \forall \mu_i,\ i = 1,...,m, \\ u_j \in U_0 = \left[\varepsilon,\ \left(u_j^0\right)^{1/\alpha_j}\right],\ j = 1,...,n. \end{cases} \tag{3.2}$$

By logarithmizing the second group of constraints of the problem (3.2) and taking into account the first one, we get

$$u^0 \to \min_{(u^0, u_j)\in U}, \tag{3.3}$$

$$U = \begin{cases} \ln u^0 \geq \dfrac{\sum\limits_{(j,k)\in\mu_i} \ln u_j^0 - \sum\limits_{(j,k)\in\mu_i} \ln w_{jk} + \ln p_0}{\sum\limits_{(j,k)\in\mu_i} \alpha_j}, \\ u_j \in U_0. \end{cases} \tag{3.4}$$

Let the constraints of the problem be consistent, both here and in the future $U \neq \varnothing$. Due to inequality (3.4), the consequence of the original formulation, taking into account the network specifics of the problem, we will seek its solution in the form of the maximum path with double weights, when the optimal solution of the problem (3.3), (3.4) is realized on a path (possibly a set of paths) with a value λ^0 such that

$$\lambda^0 = \max_{\mu_i} \frac{\sum\limits_{(j,k)\in\mu_i} \ln u_j^0 - \sum\limits_{(j,k)\in\mu_i} \ln w_{jk} + \ln p_0}{\sum\limits_{(j,k)\in\mu_i} \alpha_j} = \max_{\mu_i} \frac{\sum\limits_{(j,k)\in\mu_i} \sigma_{jk}}{\sum\limits_{(j,k)\in\mu_i} \alpha_j}.$$

Wherein, $\min\limits_{(u^0,u_j)\in U} u^0 = \exp\lambda^0$. Therefore, it is necessary to find (possibly, a set) of maximal paths with double weights on arcs in an acyclic directed graph $G\ (E, \Gamma)$.

$$\mu^{0*} = Arg \max_{\mu_i} \lambda(\mu_i),$$
$$u_j^* = u^0 = \exp\lambda^0 = \exp\lambda\left(\mu^{0*}\right),\ j \in \mu^{0*}$$

We assume that the parameters of the problem are such that here and in the future $u \in U_0$ it is satisfied. Received *resource critical path*, with the maximum consumption of a non-renewable resource. Let $\mu^{0*} = Arg \max\limits_{\mu_i} \lambda(\mu_i)$ are found. The optimal solution of (3.3), (3.4) we fix:

$$u_j^* = u^0 \to \text{fixed},\ \ j \in \mu^{0*},$$
$$E^0 = \left\{ j \middle| \, j \in \mu^{0*} \right\} \to \text{fixed}.$$

And we consider the problem of allocating a non-renewable resource according to a successively applied minimax criterion, varying uncommitted variables. In this case, the task of the second stage is:

$$\max_{j\in E\supseteq E^0} u_j \to \inf_{u\in U},\ u_j \geq 0, \quad j = 1,\ldots,n.$$

or

$$u^1 \to \inf_{(u^1,u_j)\in U},$$

$$U = \begin{cases} u_j \leq u^1, \quad \forall j = 1,\ldots,n, \\ \prod\limits_{(j,k)\in\mu_i} w_{jk} \dfrac{\left(u_j^*\right)^{\alpha_j} u_j^{\alpha_j}}{u_j^0} \geq p_0 \in [\varepsilon,\ 1], \\ u_j \in U_0, \quad \forall \mu_i,\ i = 1,\ldots,m. \end{cases}$$

By logarithmizing the second group of constraints of the problem and taking into account the first one, we get

$$\ln u^1 \ge \frac{\sum\limits_{(j,k)\in\mu_i} \ln u_j^0 - \sum\limits_{(j,k)\in\mu_i} \ln w_{jk} + \ln p_0 - \sum\limits_{j\in\mu^{0*}} \alpha_j \ln u_j^*}{\sum\limits_{(j,k)\in\mu_i} \alpha_j}$$

$$\lambda^1 = \max_{\mu_i} \frac{\sum\limits_{(j,k)\in\mu_i} \ln u_j^0 - \sum\limits_{(j,k)\in\mu_i} \ln w_{jk} + \ln p_0 - \sum\limits_{j\in\mu^{0*}} \alpha_j \ln u_j^*}{\sum\limits_{(j,k)\in\mu_i} \alpha_j} =$$

$$= \max_{\mu_i} \frac{\sum\limits_{(j,k)\in\mu_i} \sigma_{jk}^1}{\sum\limits_{(j,k)\in\mu_i} \alpha_j}, \quad \mu^{1*} = Arg \max_{\mu_i} \lambda^1(\mu_i).$$

We also fix the optimal solution to the problem of the second stage:

$$u_j^{1*} = u^1 = \exp \lambda^1 (\mu^{1*}) \to \text{fixed},$$

$$j \in \mu^{1*} \setminus \mu^{0*} = E^1 \to \text{fixed}.$$

At the same time, we gradually expand the viewer set $j \in E \supseteq (E^1 \cup E^0) \supseteq \varnothing$.

Next, we consider the minimax problem of the third stage, the optimal solution of the previous stages is fixed, and the unfixed variables vary.

$$\max_{j\in E\supseteq(E^1\cup E^0)} u_j \to \inf_{u\in U}, u_j \ge 0, j = 1,\dots,n.$$

We continue to allocate resources according to the minimax objective function successively until at some stage q (the set indicator) we get the domain of definition of the problem $E \supseteq (E^q \cup \ldots \cup E^r \cup \ldots \cup E^0)$ and

$$u_j^{q*} = u^q = \exp \lambda^q (\mu^{q*}) \to \text{fixed},$$

$$E^q = \left\{ j \middle| j \in \mu^{q*} \setminus \bigcup_0^{q^*-1} \mu^r \right\} \to \text{fixed},$$

$$E = \bigcup_0^q E^r . \tag{3.5}$$

The overall problem is solved by successive optimization of the remaining sub graph by the minimax objective function at each stage, as in the lexicographic method of organizing the solution of multi-criteria optimization problems. When the whole set of vertices is covered in optimal ways, this will solve the problem of a successive, monotonically expanding the scanned area, uniform allocation of a non-renewable resource according to a minimax criterion with a restriction on the assessment of reliability index of the project in the worst case. This procedure converges in a finite number of stages. Such a lexicographic disintegration, in essence, is a decomposition procedure and is suitable for the planning of large-scale projects.

To find the path of maximum efficiency in [7, p. 13], an algorithm of a search type was proposed (with exponential complexity) that reduces to finding the maximum path in the network. In [8, p. 192], to find in the graph with double weights of the cycle with the minimum value, the detection algorithm in the graph of the negative weight cycle is used (provided that for all cycles the sum of weights in the denominator is positive). The solution of this problem with double weights using the proposed algorithm requires $O\left(|E|^3 \log\frac{1}{\eta}\right)$ operations, where $\eta -$ the magnitude of the error (weakly polynomial algorithm), $|E|$ - is the number of vertices in the network. In [9], a general method was proposed for solving the average problem in linear spaces. From it, as a special case, follows a strong polynomial algorithm for finding the minimal average contour in a weakly connected oriented graph, which complexity is $O\left(|E|^3|\Gamma|\right)$. To solve the problem of finding the maximum path with double weights in the acyclic digraph *G* (*E*, *Γ*), we propose a greedy heuristic, for which, as will be shown below, the estimate $O\left(|\Gamma|^2\right)$ of its computational complexity is valid. The advantage of which, compared with the above, is lower computational complexity.

4. GREEDY ALGORITHM OF FINDING THE MAXIMUM PATH IN DIGRAPH WITH DOUBLE WEIGHTS ON ARCS

Construction of a set of paths $M = \{\mu_s\}$, with weights σ_{ji} and α_j on arcs l_{ji} , $(j,i) \in \Gamma$ of the digraph *G* (*E, Γ*) covering all its vertices *E*.

The initial step. Watched: arcs $L := \varnothing$, vertices $E' := \varnothing$, path number *s*: = 1.

1). Selection of an arc on a set of unvisited arcs $\tilde{\Gamma} = \Gamma \backslash L$. In the acyclic digraph *G* (*E*, *Γ*), we choose an arc l_{ji}, where $(j,i) \in \Gamma$ is optimal by the local criterion

$$l_{ji} = Arg \max_{(j,i)\in\Gamma} \frac{\sigma_{ji}}{\alpha_j}.$$

In the array L we bring the arc $l_{ji} : L := l_{ji}$.

2). Build a path.

2.1). Adding new arcs.

Construct a path μ_s from the initial dummy vertex *j*: $\Gamma_j^{-1} = \varnothing$ to the final vertex *j* = *n*: $\Gamma_j = \varnothing$ by local criterion, i.e. for an arc l_{ji}, adjacent arcs l_{kj} and l_{it} are selected from conditions

$$l_{kj} = Arg \max_{k\in\Gamma_j^{-1}} \frac{\sigma_{kj} + \sum_{(j,i)\in L} \sigma_{ji}}{\alpha_k + \sum_{j\in L} \alpha_j},$$

$$l_{it} = Arg \max_{t\in\Gamma_i} \frac{\sum_{(j,i)\in L} \sigma_{ji} + \sigma_{it}}{\sum_{j\in L} \alpha_j + \alpha_i},$$

which are added to the existing arcs $L := l_{ji}$, forming a path $\mu_s = \{...l_{kj} \cup L \cup l_{it}...\}$ with double weights. Where $\sum_{(j,i)\in L} \sigma_{ji}$ and $\sum_{j\in L} \alpha_j$ - values for the path found in the previous step.

Putting $\Delta_{\mu}^{s} := 1$, for the constructed in such a way the track μ_s, we define the value $\lambda^{s}\left(\mu_s\right)$, s = 1,2, ...

$$\max_{\mu}\frac{\sum_{(j,i)\in\mu}\sigma_{ji}}{\sum_{(j,i)\in\mu}\alpha_{j}}\Delta_{\mu}^{s}=\lambda^{s}\left(\mu_s\right).$$

List of viewed arcs - $L:=\left\{...l_{kj}\cup L\cup l_{it}...\right\}$; for viewed vertices – $E^s = \{j|j \in \mu_s\}$; $E' := E' \bigcup E^s$.

2.2). Avoiding algorithm loops on scanned paths.

In order to prevent the algorithm from looping along the scanned path μ_s, in the future we will calculate the indicator value Δ_{μ}^{s+1}. If for non-viewed vertices $E\setminus E'\neq\varnothing$, then let $k\in E\setminus E'$

We put $\delta_k=\begin{cases}1, \text{ if } k\in E',\\ 0, \text{ if } k\in E\setminus E'.\end{cases}$ Then $\begin{cases}\Pi_{k\in\mu}\delta_k=1, \ \forall k\in E';\\ \Pi_{k\in\mu}\delta_k=0, \text{ if } \exists k_0\in\mu:\ k_0\in E\setminus E'.\end{cases}$

In this case, we set

$$\Delta_{\mu}^{s+1}=\begin{cases}-\infty, \text{ if } \Pi_{k\in\mu}\delta_k=1,\\ 1, \text{ if } \Pi_{k\in\mu}\delta_k=0,\end{cases}$$

and then in the future, we will sequentially determine

$$\max_{\mu}\frac{\sum_{(j,i)\in\mu}\sigma_{ji}}{\sum_{(j,i)\in\mu}\alpha_{j}}\Delta_{\mu}^{s+1}=\lambda^{s+1}\left(\mu_{s+1}\right)$$

$\lambda^{s+1}\left(\mu_{s+1}\right)>-\infty$ only for new paths μ_{s+1} that do not coincide with $\left\{\mu_s\right\}$, s = 1,2, ...

3). Cycle If for unvisited arcs $\tilde{\Gamma}\neq\varnothing$, then cycle through $s:=s+1$ and go to step 1. otherwise: Viewing the constructed set of paths $\left\{\mu_{s+1}\right\}$ and determining the optimal path(s) with double weights on arcs $\mu^{0*}=\operatorname{Arg}\max_{s}\lambda^{s}(\mu_s)$, s = 1,2, ... For the first stage (3.3), (3.4) of the problem: $\lambda^0=\lambda\left(\mu^{0*}\right)$, and for the subsequent ones $\lambda^r=\lambda\left(\mu^{r*}\right)$, r = 1, ..., q.

In the algorithm, the review of arcs for constructing a set of paths is carried out from Γ to $\varnothing$, which allows you to avoid a complete search of the paths, and the introduction of a counter $\Delta_{\mu}^{s}\neq-\infty$ for paths that have at least one unwatched arc eliminates looping the algorithm on the same old path. Described algorithm is, in essence, greedy heuristic and so it gives approximate solution of the above problem.

The complexity of the algorithm described is estimated as follows. The selection of the initial locally optimal arc l_{ji} requires $|\Gamma|$ operations, building a path μ from the initial vertex to the final vertex is also requires the order of $D|\Gamma|$ operations, where D is the maximum degree of the vertices of the graph G (E, Γ). The construction of all such paths for each of the $|\Gamma|$ arcs is repeated about $\mathrm{O}\left[\left(|\Gamma|+D|\Gamma|\right)|\Gamma|\right]$ time plus the successive selection of the λ-optimal path, which will require order $|\Gamma|$ operations. Therefore, the complexity makes up $O\left[\left(|\Gamma|+D|\Gamma|\right)|\Gamma|+|\Gamma|\right]\sim O\left(|\Gamma|^2\right)$ of operations. And for a complete vertex coverage, you will need to repeat such an algorithm one more $|\Gamma|$ time, then the total will be required $O\left(|\Gamma|^3\right)$ operations.

For clarity, we illustrate (Table 1) finding the first maximal path with double weights on arcs by calculating all the available routes and choosing the maximal one.

Table 1. Finding the first maximal values with double weights on the paths for $\ln u^0$

path number	$\ln u^0$
1	**4.0628**
2	2.3828
3	2.7416
4	2.3891
5	2.9082
6	2.1519
7	2.3016
8	2.1389
9	2.8276
10	2.5899
11	2.2212
12	2.2322
13	2.2010
14	2.5093
15	2.3978
16	2.2619
17	2.7578
18	3.1007
19	2.3034
20	2.2053

A test network consisting of 32 vertex-activities, from the well-known library of network models of projects from http://www.om-db.wi.tum.de/psplib/ provide precedence relations of the selected network project as in the Table 2 below.

Table 2. Precedence relations of the test network project.

jobnr. - activity number;

#modes - the number of execution modes (does not matter);

#successors - the number of vertices at which the edges of the relations of precedence go out of the vertex of this work;

successors - numbers of activities that go to the edges of the relations of the preceding from this vertex.

Table 2. Precedence relations of the test network project.

jobnr.	#modes	#successors	successors
1	1	3	2 3 4
2	1	3	6 11 15
3	1	3	7 8 13
4	1	3	5 9 10
5	1	1	20
6	1	1	30
7	1	1	27

8	1	3	12 19 27
9	1	1	14
10	1	2	16 25
11	1	2	20 26
12	1	1	14
13	1	2	17 18
14	1	1	17
15	1	1	25
16	1	2	21 22
17	1	1	22
18	1	2	20 22
19	1	2	24 29
20	1	2	23 25
21	1	1	28
22	1	1	23
23	1	1	24
24	1	1	30
25	1	1	30
26	1	1	31
27	1	1	28
28	1	1	31
29	1	1	32
30	1	1	32
31	1	1	32
32	1	0	

Above information from Table 2 about relations of technological precedence of selected network project will be organized then with the help of matrix "columns - vertices" and "rows - paths by vertices". Which specifies the characteristic function of the constraint system, and the parameters are specified in the last two columns of Table 3 and Table 4.

Table 3. Upper the border ub for $\ln u^0$

№ j=	Upper the border ub for $\ln u^0$ ub =	form parameter α_j =	scale parameter u_j^0 =
0		0.3000	0.5100
1	4.3944	0.2500	0.6000
2	4.1650	0.2200	0.5000
3	0	0.2400	0.5000
4	0	0.2500	0.5000
5	0	0.2800	0.4800
6	4.3565	0.2300	0.4780
7	0	0.2330	0.4750
8	0	0.2350	0.4600
9	0	0.2380	0.4550
10	0	0.2400	0.4500
11	0	0.3000	0.4480

12	0	0.3200	0.4460
13	0	0.3500	0.4450
14	0	0.3800	0.4420
15	0	0.4000	0.4400
16	0	0.4200	0.4300
17	0	0.4300	0.4200
18	0	0.4400	0.4000
19	0	0.4420	0.3800
20	0	0.4450	0.3500
21	0	0.4460	0.3200
22	0	0.4480	0.3000
23	0	0.4500	0.2400
24	0	0.4550	0.2380
25	0	0.4600	0.2350
26	0	0.4750	0.2330
27	0	0.4780	0.2300
28	0	0.4800	0.2800
29	0	0.5000	0.2500
30	4.4742	0.2800	0.7000
31	0	0.5000	0.3200
32	4.1759	0.3000	0.7000

Table 4. Numerical solution to the problem (3.2) as a whole

№ j=	$\ln u_j$ =	Restrictions c =	Upper the border ub =	form parameter α_j =	scale parameter u_j^0 =
0	3.1203	25.000	3,1203	0,300	0,510
1	4.3944	4.1021	4,3944	0,250	0,600
2	4.1650	3.7607	4,1650	0,220	0,500
3	3.8179	3.4393	3,8179	0,240	0,500
4	3.6652	3.2914	3,6652	0,250	0,500
5	**0.2924**	0.0444	3,1267	0,280	0,480
6	**0.2924**	0.0444	4,3565	0,230	0,478
7	**0.2924**	0.0394	3,7124	0,233	0,475
8	**0.2924**	0.0444	3,5443	0,235	0,460
9	**0.2924**	0.1276	3,4537	0,238	0,455
10	**0.2924**	0.0444	3,3789	0,240	0,450
11	**0.2924**	0.0444	2,6883	0,300	0,448
12	**0.2924**	0.0444	2,5063	0,320	0,446
13	1.7583	1.4512	2,2850	0,350	0,445
14	**0.2924**	0.0444	2,0868	0,380	0,442
15	**0.2924**	0.0444	1,9711	0,400	0,440
16	**0.2924**	0.0444	1,8225	0,420	0,430
17	1.2483	0.9752	1,7254	0,430	0,420
18	1.5753	1.2756	1,5753	0,440	0,400

19	**0.2924**	0.0444	1,4522	0,442	0,380
20	1.2576	0.9652	1,2576	0,445	0,350
21	**0.2924**	0.0307	1,0538	0,446	0,320
22	0.9051	0.6127	0,9051	0,448	0,300
23	0.4052	0.1128	0,4052	0,450	0,240
24	0.3823	0.0899	0,3823	0,455	0,238
25	0.3506	0.0582	0,3506	0,460	0,235
26	0.3215	0.0291	0,3215	0,475	0,233
27	**0.2924**	0	0,2924	0,478	0,230
28	0.7010	0.4086	0,7010	0,480	0,280
29	0.4463	0.1539	0,4463	0,500	0,250
30	4.4742	4.1818	4,4742	0,280	0,700
31	0.9400	0.6476	0,9400	0,500	0,320
32	4.1759	3.8835	4,1759	0,300	0,700

$u^0 = 1.3396.$

If we solve the problem in question (3.2) as a whole, as the problem of mathematical programming using ready-made software, we obtain the numerical solution presented in Table 4 above.

For the $w_{jk} \in [\varepsilon_0,\ 1],\ \varepsilon_0 > 0$ – arc reliability coefficient (*j, k*), (from the matrix of paths along the arcs of the network *G* (*E, Γ*):) transfer the result of the *j*-th activity through the arc (*j, k*) to perform the *k*-th activity (like the transfer function of the connection links in the theory of automatic control) we have the following Table 5 values.

Table 5. Arcs reliability coefficient

Γ^0	arcs	value w_{jk}
1	(5,20)	0.95
2	(11,20)	0.95
3	(18,20)	0.95
4	(16,22)	0.95
5	(17,22)	0.95
6	(18,22)	0.95
7	(10,25)	0.95
8	(15,25)	0.95
9	(20,25)	0.95

For all other arcs of the network $w_{jk} = 1\ \ \forall w_{jk} \in \Gamma^1 = \Gamma \setminus \Gamma^0$.

5. DETERMINATION OF THE SCHEDULE WITH INACCURATE SOURCE DATA

Since the values of the scheduling model parameters

$$\alpha_j \in (0,1),\ w_{jk} \in [\varepsilon_0,\ 1], \varepsilon_0 > 0,\ u_j^0 > 0,$$

$$j,k = 1,...,n,\quad (j,k) \in \Gamma \subset G(E,\Gamma)$$

are known with a certain error, the resource allocation algorithm, as well as the physical volume of work v [USD × hour] and the amount of a non-renewable resource, found as a result of solving the considered problem, which has the value expression u [USD], are approximate, the project's time parameters - the duration of the activity $t_j^* = v_j / u_j^{r*}$, *r* = 0,1, ...,

$q\ j=1,...,n$; and the activities slacks are determined by processing the numbers known with an error. The obtained solution of problem (3.1), (3.5) together with reliability constraints, vector

$$\mathrm{u}^*=\left\{u_j^{r*}\right\},\ u_j^{r*}>0,\ j=1,...,n,$$
$$r=0,1,...,q,\ \left\{r\right\}\subset\left\{j\right\},\ q\ll n$$

such that (omitting the index r)

$$u_j^*=\left\{u_j^*(\delta_0)\middle| u_j^*(\delta_0)\in\left[\left[u_j^*\right]+\Delta u_j^*\right]\right\}. \quad (5.1)$$

By denoting inaccurately set time values as $t(\delta)$, using actions on approximate numbers [5], one can obtain approximate parameters of the optimal schedule. To do this, we calculate the deterministic project schedule for one pass in the forward and reverse directions based on the obtained optimal (under given error) duration of the project $t_i^*(\delta_0)$, starting from the initial project activities.

The Critical Path Method, with values, specified inaccurately (5.1), using actions on approximate numbers [5], also allows you to calculate late deadlines for performing project activities in a back pass through the network, starting from the project completion date (calculated by direct passing through the network). And also to calculate slacks of activities [5]: total (full) slack, free (local) slack, safe slack, independent slack.

The total (full) slack SL^{t} of the project - the period of time by which you can postpone the activity without violating the limitations and deadlines for the project

$$SL_i^{\mathrm{t}}(\delta)=t_i^{lb}(\delta)-t_i^{eb}(\delta)=t_i^{lc}(\delta)-t_i^{ec}(\delta).$$

Free (local) slack SL^{f} of the project - the period of time by which you can postpone the activity without violating the deadline for the subsequent work –

$$SL_i^{\mathrm{f}}(\delta)=\min_{j\in\Gamma_i}(t_j^{eb}(\delta)-d_{ij}(\delta))-t_i^{ec}(\delta).$$

The safe slack SL^{s} of the project activity is the period of time for which activity *j* can be prolonged when all its predecessors $i\in\Gamma_j^{-1}$ started working at the latest date so that the shortest possible time to complete the project does not increase –

$$SL_j^{\mathrm{s}}(\delta)=t_j^{lb}(\delta)-\max_{i\in\Gamma_j^{-1}}t_i^{lc}(\delta).$$

An independent slack SL_j^{i} of the project activity *j* is equal to the time for which the duration of activity *j* can be prolonged regardless of the time of completion of its predecessors - $i\in\Gamma_j^{-1}$ and the time of the beginning of its followers $k\in\Gamma_j$ –

$$SL_j^{\mathrm{i}}(\delta)=\max\left\{0;\min_{k\in\Gamma_j}t_k^{eb}(\delta)-\max_{i\in\Gamma_j^{-1}}t_i^{lc}(\delta)-t_j^*(\delta)\right\}.$$

The following relations are valid (under $\delta\to 0$):

$$SL_j^{\mathrm{t}}(\delta)\ge SL_j^{\mathrm{f}}(\delta)\ge SL_j^{\mathrm{i}}(\delta);\qquad SL_j^{\mathrm{t}}(\delta)\ge SL_j^{\mathrm{s}}(\delta)\ge SL_j^{\mathrm{i}}(\delta).$$

CONCLUSION

The article discusses a deterministic network model of a project in which, for a project's overall reliability indicator, it is fair its assessment from below in the form of the weakest link – in the worst technological chain from the initial vertex of the project to the final one.

For the indicator of the reliability of the activity-vertex of the project, a power-concave model is used depending on the spent non-renewable (stored) resource. The peculiarity of the model is weighted network arcs, which set the arc reliability factor (*j, k*), transferring the result of the *j*-th activity by the arc (*j, k*) to perform the *k*-th activity.

Under these conditions, the problem of planning and scheduling projects is considered on the basis of a successively applied minimax criterion according to the type of lexicographic ordering of criteria in multi-criteria optimization problems. Such a lexicographic dissociation, in essence, is a decomposition procedure and is suitable for planning large-scale projects.

To solve the problem of uniform allocation of a non-renewable resource, a greedy heuristic has been developed for finding the maximum path with double weights on arcs of an acyclic digraph, computational complexity that is quadratic in the number of arcs. The greedy heuristic gives approximate solution of the problem under consideration. Which allows you to get a *resource critical path*, with the maximum consumption of a non-renewable resource. After that, the project schedule is determined, including its temporary critical path, with inaccurate data. Practical calculations for the considered convex problem (3.2) can be performed using ready-made software, and the resulting greedy solution of the problem-consequence (3.3), (3.4) has theoretical significance in graph theory.

REFERENCES

1. *The Owner's Role in Project Risk Management.* (2005) Washington, D.C.: The National Academies Press, www.nap.edu.
2. Elkjaer M. (2000). Stochastic Budget Simulation, *Int. J. Project Management.* 18(2), 139-147.
3. Rabbani M., Tavakkoli Moghaddam R., Jolai F. & Ghorbani H.R. (2006). A Comprehensive Model for R&D Project Portfolio Selection with Zero-One Linear Goal-Programming, *IJE Transactions A. Basics.*19(1), 55-66.
4. Tsvirkun A.D., Akinfiyev V.K. & Konovalov Ye.N. (1991). Modelirovaniye i upravleniye innovatsionnymi programmami v krupnomasshtabnykh tekhnicheskikh sistemakh. [Modeling and management of innovative programs in large-scale technical systems] – Moscow, Russia: (Working Paper / ICS). [in Russian].
5. Topka V.V. (2014). Lexicographic Solution of Two-Objective Project Planning Problem under Constrained Reliability Index, *J. Computer & Systems Sci. Int.* 53(6), 877-895.
6. Topka V.V. (2012). Minimization of Project Time and Cost under Constrained Reliability Index in the Disjunctive Project Model, *Automation and Remote Control.* 73(7), 1173-1180.
7. Burkov, V.N., Zalozhnev, A.Yu. & Novikov, D.A. (2001). *Teoriya grafov v upravlenii organizatsionnymi sistemami.* [Graph theory in the management of organizational systems.]. Moscow, Russia: Sinteg, [in Russian].
8. Christofides N. (1975). *Graph theory: An algorithmic approach* (*Computer science and applied mathematics*). N.Y.: Academic Press.
9. Karzanov A.V. (1985). *O minimal'nykh po srednemu vesu razrezakh i tsiklakh oriyentirovannogo grafa* [On minimal cuts and cycles with respect to average weight of an oriented graph] / *In: Kachestvennyye i priblizhennyye metody issledovaniya operatornykh uravneniy*. (pp. 72-83). [Qualitative and approximate methods for the investigation of operator equations.] - Yaroslavl', USSR: Yar SU, [in Russian].

