ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ. КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

УДК 519.161

A.V. Morozov, Ph.D., Ass.Prof.
N.O. Kushnir, Ph.D., student
T.M. Loktikova, Seniour Lecturer
Zhytomyr State Technological University

ON THE TASK OF BUILDING THE ROUTES OF PASSENGER BUSES OF TWO AUTOMOBILE COMPANIES

The article formulates a mathematical model search of n bus routes between the two points, which carry out cruises corresponding to the specified schedule, and with specified duration. The duration of each route consists of two cruises and idle hours, which are determined by the moment of the completion of the first cruise and the moment of the beginning of the second. Totally, 2n cruises are performed and provide the passenger transportation by n cruises. In this specific version of the problem is the additional condition, that the execution time of each route shall not exceed the established limit of a standard d.

Offered framework for the solution of the problem of procedure to the task assignment and modification of Kuhn-Munkres algorithm, which is looking for a solution of the problem of assignment to the maximum. The proposed numerical scheme is an iterative process, each step of which provides the topmost layout. To adapt the task to form, which allows to apply the modification of Kuhn-Munkres algorithm, to consider the bichromatic graph, which builds perfect matching with a maximum weight of the ribs.

Keywords: *assignment problem; bichromatic graph; Kuhn-Munkres algorithm.*

Problem statement. The problem of increase of efficiency of functioning of transport systems constantly needs development and improvement of methods and models, aimed at building routes of movement of vehicles. A wide range of tasks that simulate the processes of management and planning in transport networks, formally reduced to problems of finding the closed routes [1-5]. One of these tasks is examined in this article.

Suppose two automobile companies which are located in points 1 and 2 perform passenger transportations between these points by n buses. Dispatch time for all $2n$ cruises from point 1 to point 2 and vice versa is known from the schedule. Every bus of an automobile company k, $k=\overline{1,2}$, according with a schedule starts and finishes the route consisted of two cruises in the same point.

Cruise i from the point 1 to the point 2 starts at the time moment $t_{1,i}$, $i=\overline{1,n}$, and its duration equals τ_{1i}. Cruise j from the point 2 to the point 1, which starts at the time moment t_{2j}, $j=\overline{1,n}$, is performed in a time τ_{2j}. For a bus which leaves as a cruise i from the point 1 to the point 2 and comes back as a cruise j from the point 2 to the point 1, way time is calculated:

$$d_{ij}^1 = t_{2j} - t_{1i} + \tau_{2j};\ t_{2j} - t_{1i} \ge \tau_{1i}.$$

The duration of pendulous cruise of a bus performing firstly a cruise j, and then a cruise i, equals to

$$d_{ij}^2 = t_{1i} - t_{2j} + \tau_{1i},\ t_{1i} - t_{2j} \ge \tau_{2j}.$$

Assigning a task to find n cruises which according to the specified route to bus stations with specified duration of routes τ_{1i} і τ_{2j}, $i, j=\overline{1,n}$, could be performed at the minimal total time.

Presentation of basic material of the research. We sum two tables $\left[d_{ij}^1\right]_n$ and $\left[d_{ij}^2\right]_n$. Every element d_{ij}^1 of the first table equals to the time of performance of probable pendulous cruise which includes the cruise i from the point 1 to the point 2 and the following cruise j from the point 2 to the point 1. In other words, a table $\left[d_{ij}^1\right]_n$ obtained assuming that each bus starts its route in the point 1. Tables $\left[d_{ij}^2\right]_n$ includes the duration of all possible routes starting with a cruise j from point 2 in point 1 and finish with a cruise i from point 1 to point 2. In this case, we assume that every bus goes from point 2 to point 1 and returns to the same point 2.

Overlaying matrix $\left[d_{ij}^2\right]_n$ on matrix $\left[d_{ij}^2\right]_n$ we obtained a table $\left[\left(d_{ij}^1, d_{ij}^2\right)\right]_n$ of ordered formed the matrix $\left[d_{ij}\right]_n$, where $d_{ij} = min\left(d_{ij}^1, d_{ij}^2\right)$. Now it is clear that the solution of this problem is solving the assignment problem (AP) for outgoing data presented in matrix $\left[d_{ij}\right]_n$.

When passing from the real situation to the mathematical model, we often have to ignore some very significant details. Particularly, in practice the optimal solution of AP in the given interpretation can become unsuitable. It refers to a case where between the largest and smallest values of its components unacceptably large discrepancy is revealed. Most real circumstances require solutions that provide equal load on performing sites within the prescribed limits. Indeed, in the situation, the duration of each route has two cruises and idle hours, determined the time of completion of the first cruise and the time of the start of the second. Clearly, the need for planning such idle hours as simple inter-shift break gives no rise to doubt. But it can not be both too short and too long.

Of course, this kind of detail account is the «complications» of a mathematical model and usually accompanies by certain difficulties in the development of algorithms for solving the following task. In the version under consideration, this complication is an additional condition, according to which the performance of each route must not exceed the limit set by regulation d.

In the AP terminology in this case it is necessary to allocate performance i to machines j, $i, j = \overline{1,n}$, so, to minimize

$$B(\pi) = \sum_{j=1}^{n} d_{\pi[i]} \,, \tag{1}$$

where $\pi = (\pi[1], \pi[2], ..., \pi[i], ..., \pi[n])$ − permutation of matrix columns $\left[d_{ij}\right]_n$, $d_{ij} \in R_0^+$,

$$d_{\pi[i]} = \begin{cases} d_{ij,} \text{ if the job } i \text{ assingned to the machine } j = \pi[i]; \\ \infty \text{ otherwise,} \end{cases}$$

on condition that for the specified value $d \in R_0^+$

$$d_{\pi[i]} \le d \,, \; i = \overline{1,n} \,. \tag{2}$$

It turns out that the task (1) is easily reduced to a minimization task $B(\pi)$ without restrictions (2). It is necessary in a parent matrix$\left[d_{ij}\right]_n$ put $d_{ij} = \infty$, if $d_{ij} > d$. It is obvious that the received matrix implicitly takes the condition into account (2). Therefore, to solve the task 1) we require a AP decision algorithm which works with matrix $\left[d_{ij}\right]_n$ and which contains infinitudes of big numbers. This matrix admits the situation when AP has no solution. For this reason, the algorithm should contain means to set the case of undecidable problem.

The above mentioned requirements can be satisfied with the modification of Kuhn-Munkres algorithm for solving AP at maximum: to find the set of all permutations $\pi = \pi\ (\pi[1], \pi[2], ..., \pi[i], ...,$ $\pi[n])$ matrix columns $\left[d_{ij}\right]_n$ permutation $\pi^* = \left(\pi^*[1], \pi^*[2], ..., \pi^*[n]\right)$, which provides the maximum of functional

$$C(\pi) = \sum_{i=1}^{n} d_{\pi[i]} \,.$$

To minimize $B(\pi)$ with the help of Kuhn-Munkres algorithm it is necessary to preliminary modify the matrix$\left[d_{ij}\right]_n$, set

$$d_{ij} = \begin{cases} d_{ij}, \text{ if } d_{ij} \le d, \\ \infty \text{otherwise,} \end{cases}$$

where d – the maximum in a parent matrix $\left[d_{ij}\right]_n$, which is not equal to ∞, or integral sum number in a limit (2).

These considerations allow to formulate AP to maximum as follows.

Let in a permutation $\pi = (\pi[1], \pi[2], ..., \pi[i], ..., \pi[n])$ numbers of matrix columns $\left[d_{ij}\right]_n$, where $d_{ij} \in R_0^+$ or $d_{ij} = -\infty$, work prescription i on an automobile j is characterized by a reciprocal coefficient

$$d_{\pi[i]} = \begin{cases} d_{ij,} \text{ if the job i assingned to the machine } j = \pi[i]; \\ -\infty \text{ otherwise.} \end{cases}$$

Diagonal $\Pi = \left(d_{\pi[1]}, d_{\pi[2]}, ..., d_{\pi[i]}, ..., d_{\pi[n]}\right)$, which corresponds to π, assigned mass equals to $C(\pi) = \sum_{i=1}^{n} d_{\pi[i]}$.

The following permutation should be found $\pi^* = \left(\pi^*[1], \pi^*[2], ..., \pi^*[n]\right)$, which

$$C\left(\pi^*\right) = \max_{\pi} C(\pi). \quad (3)$$

Matrix $\left[d_{ij}\right]_n$ corresponds to bichromatic graph $G(X, Y, U)$, $|X| = |Y| = n$, where a node $x_i \in X$ connects to a node $y_j \in Y$ by a link $\left(x_i, y_j\right)$ with a mass $d\left(x_i, y_j\right) = d_{ij} \neq -\infty$. In a graph G it is required to find the ideal matching π^* with a maximal sum of link weight $C\left(\pi^*\right)$ [6-7].

Developed π^* by the Kuhn-Munkres' method it appears to be iteration process connected with calculation of function f, which is called an accepted topmost layout.

An accepted topmost layout with function f with a value of a set of real numbers which is defined on a multitude $X \cup Y$ so that

$$f(x) + f(y) \geq d(x, y), \quad x \in X, \quad y \in Y.$$

Then $f(x)$ is called a layout of a node x.

Kuhn-Munkres algorithm starts working on admissible topmost layout

$$f(x_i) = \max_{1 \leq j \leq n} d\left(x_i, y_j\right), \quad i = \overline{1,n}, \quad f\left(y_j\right) = 0, \quad j = \overline{1,n}.$$

Function f is put in accordance with a set U_f of links (x, y) to graph G, for which $f(x) + f(y) = d(x, y)$. Subgraph G_f of graph G with a set of links U_f is called a partial graph of equalities that corresponds to f.

The next theorem connects a subgraph of equalities with an optimal matching and creates the basis for Kuhn-Munkres application.

Theorem. Let f is allowable vertex layout of a graph $G = (X, Y, U)$, $|X| = |Y|$. If G_f contains an ideal matching π^*, то π^* – an ideal matching with a maximum weight $C\left(\pi^*\right)$ in a graph G.

Proof. As a partial graph of equalities G_f, which contains an ideal matching π^*, is a subgraph of a graph G, so π^* is an ideal matching in G. Due to the fact that every link u in π^* belongs to the partial graph of equalities G_f and every node of a graph G is incident to one link with π^*, so for weight $C\left(\pi^*\right)$ matching π^* formula is justified:

$$C\left(\pi^*\right) = \sum_{u \in \pi^*} d(u) = \sum_{x \in X} f(x) + \sum_{y \in Y} f(y) = \sum_{v \in X \cup Y} f(v).$$

From the other hand, if π is a random ideal matching in a graph G, then

$$C(\pi) = \sum_{u \in \pi} d(u) \leq \sum_{v \in X \cup Y} f(v).$$

So as $C\left(\pi^*\right) \geq C(\pi)$ and then π^* is an optimal matching in a graph G. □

The modification of Kuhn-Munkres algorithm for solution the task in the statement of (3) is a sequence of steps.

S0. Algorithm of solution AP for maximum. $\left[d_{ij}\right]_n$ – assigning matrix which accepts $d_{ij} = -\infty$; G – bichromatic graph with a node set $X \cup Y$, $|X| = |Y| = n$, where a node $x_i \in X$ is connected to a node $y_j \in Y$ by a link $\left(x_i, y_j\right)$ with weight $d\left(x_i, y_i\right) = d_{ij} \neq -\infty$. For every node $x_i \in Y$, which corresponds to a line i, and for every node $y_j \in Y$, which corresponds to a column j of a matrix $\left[d_{ij}\right]_n$, it is stated $f\left(x_i\right) = \max\limits_{1 \leq j \leq n} d_{ij}$, $i = \overline{1,n}$, $f\left(y_j\right) = 0$, $j = \overline{1,n}$, and a partial graph of equalities is constructed G_f; $l = 0$.

S1. Choose in G_f initial matching π_l.

S2. If all the nodes in X saturated in π_l, so π_l is an ideal matching and correspondingly $\pi_l = \pi^*$. Then to calculate $C\left(\pi^*\right)$, that's all. Otherwise u is unsaturated node in X, set $S = \{u\}$, $T = \varnothing$.

S3. Let $\Gamma_f(S)$ – set of nodes with Y, adjacent in a graph G_f with nodes in set S. If $T \subset \Gamma_f(S)$, then go to S6, otherwise $T = \Gamma_f(S)$, go to S4.

S4. If for all $x_i \in S$ и $y_j \notin T$ $d_{ij} = -\infty$, then it's the end: AP is at the maximum and there is no solution or calculate then:

$$d_f = \min\left\{ f\left(x_i\right) + f\left(y_j\right) - d_{ij} \,\middle|\, x_i \in S, y_j \notin T, d_{ij} \neq -\infty \right\}$$

And get new allowable layout

$$f'(v) \begin{cases} f(v) - d_f, \text{ if } v \in S, \\ f(v) + d_f, \text{ if } v \in T, \\ f(v) \text{ in all other cases.} \end{cases}$$

S5. Change f into f' and G_f into $G_{f'}$, go to S1; $l = 0$.

S6. Choose a node $y \in \Gamma_f(S) - T$. If y is unsaturated in π_l, then go to S7, otherwise set z – opposite number y in π_l, $S = S \cup \{z\}$, $T = T = \{y\}$ and go to S3.

S7. Increasing way with a set of line P is built. Set $\pi_{l+1} = \pi_l \oplus P$ i $l = l + 1$. Go to S2.

It should be noted that the Kuhn-Munkres algorithm is not the only algorithm of AP. Well-known algorithms are with the best temporal and capacitive estimated cost of achieving optimum. Interest to the Kuhn-Munkres algorithm is due to its features that create a mechanism of aggregation, which is the base unit action sequences on the calculation of allowable markup vertex construction of subgraph equalities and known procedure of perfect matching in a graph (X, Y, U), $|X| = |Y|$.

In the Kuhn-Munkres algorithm we start with allowable node layout f, where

$$\sum_{x \in X} f(x) + \sum_{y \in Y} f(y) = \sum_{x \in X} f(x) = const = \sum_{i=1}^{n} \max_{1 \leq j \leq n} d_{ij} \geq C\left(\pi^*\right),$$

and then build a corresponding partial graph of equalities G_f, where search of ideal matching is performed. If an ideal matching is built in G_f, so on the basis of the theorem it is optimal and the algorithm on S2 stops. If there is no optimal solution π^* in a partial graph of equalities G_f, then one of two variants is possible. Firstly, it is possible when in matrix $\left[d_{ij}\right]_n$ number of elements $\beta_{ij} = -\infty$ and their location presupposes the absence of a task solution (3). Then calculations stop on S4. Secondly, if G_f doesn't contain π^*, but to the graph G_f we can add links which extend it to $G_{f'}$, then on S4 changes of allowable vertex layout f are performed and a process of finding an optimal solution is repeated in a partial graph of equalities $G_{f'}$, which corresponds to f'. Variation of values $f(x)$ i $f(y)$, which don't change their sum are performed till the ideal matching will be found.

It is obvious that the Kuhn-Munkres algorithm finds a solution to PA during polynomial time.

Example. Schedule in bus stations 1 and 2 defined the following table:

Point 1			Point 2		
i	t_{1i} Departure time	τ_{1i} Way time	j	t_{2j} Departure time	τ_{2j} Way time
1	7.00	5	1	6.00	5
2	8.00	6	2	7.00	6
3	13.00	6	3	14.00	6
4	16.00	5	4	15.00	6
5	18.00	6	5	19.00	5

The table with the limit setting the duration of each route after these elementary transformations defines the inputs of (1).

Over the following table we find the matrix

$$\left[d_{ij}^1\right]_5 = \begin{bmatrix} 28 & 30 & 13 & 14 & 17 \\ 27 & 29 & 12 & 13 & 16 \\ 22 & 24 & 31 & 32 & 11 \\ 19 & 21 & 28 & 29 & 32 \\ 17 & 19 & 26 & 27 & 30 \end{bmatrix}, \quad \left[d_{ij}^2\right]_5 = \begin{bmatrix} 30 & 32 & 13 & 15 & 18 \\ 29 & 31 & 12 & 14 & 17 \\ 22 & 24 & 29 & 31 & 34 \\ 21 & 23 & 28 & 30 & 33 \\ 17 & 19 & 24 & 26 & 29 \end{bmatrix}, \quad \left[d_{ij}\right]_5 = \begin{bmatrix} 28 & 30 & 13 & 14 & 17 \\ 27 & 29 & 12 & 13 & 16 \\ 22 & 24 & 29 & 31 & 11 \\ 19 & 21 & 28 & 29 & 32 \\ 17 & 19 & 24 & 26 & 29 \end{bmatrix}.$$

Setting the restrictions $d_{ij} \le d = 26$, we get

$$\left[d_{ij}\right]_5 = \begin{bmatrix} \infty & \infty & 13 & 14 & 17 \\ \infty & \infty & 12 & 13 & 16 \\ 22 & 24 & \infty & \infty & 11 \\ 19 & 21 & \infty & \infty & \infty \\ 17 & 19 & 24 & 26 & \infty \end{bmatrix}.$$

We will solve AP for maximum of matrix

$$\left[d_{ij}\right]_5 = \begin{bmatrix} -\infty & -\infty & 13 & 12 & 9 \\ -\infty & -\infty & 14 & 13 & 10 \\ 4 & 2 & -\infty & -\infty & 15 \\ 7 & 5 & -\infty & -\infty & -\infty \\ 9 & 7 & 2 & 0 & -\infty \end{bmatrix}.$$

Perform the Kuhn-Munkres algorithm.

S0. According to the matrix $\left[d_{ij}\right]_5$ we get the following allowable node layout:

$$f(x_i) = 13,\ 14,\ 15,\ 7,\ 9;\quad f(y_i) = 0,\quad j = \overline{1,5},$$

and then we find a partial graph of equalities G_f (fig. 1); $l = 0$.

S1. Choose in G_f initial matching $\pi_0 = \{(x_1, y_3)\}$.

S2. Node x_2 is unsaturated in π_0. Set $S = \{x_2\}$, $T = \varnothing$.

S3. $\Gamma_f(S) = \{y_3\}$, go to S6.

S6. Choose the node $y_3 \in \Gamma_f(S) - T$. Since y_3 is saturated in π_0, we define its opposite number – node x_1 in π_0; $S = \{x_1, x_2\}$, $T = \{y_3\}$; go to S3.

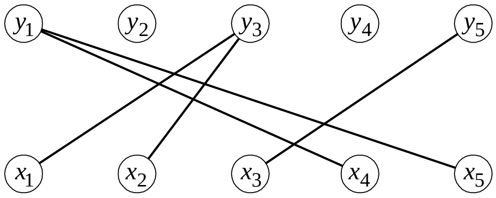

Fig. 1. Partial graph G_f, corresponded to the first node layout f

S3. $\Gamma_f(S) = \{y_3\}$, $\Gamma_f(S) = T$.

S4. Calculate $d_f = min\{f(x_1)+f(y_4)-d_{14}, f(x_1)+f(y_5)-d_{15}, f(x_2)+f(y_4)-d_{24}, f(x_2)+f(y_5)-d_{25}\}=$

$= min\{13+0-12, 13+0-9, 14+0-13, 14+0-10\} = 1$. Define new node layout:

$f(x_1) = 12$, $f(x_2) = 13$, $f(x_3) = 15$, $f(x_4) = 7$, $f(x_5) = 9$;

$f(y_1) = 0$, $f(y_2) = 0$, $f(y_3) = 1$, $f(y_4) = 0$, $f(y_5) = 0$.

Partial graph of equalities G_f corresponds to obtained layout and is shown on fig. 2.

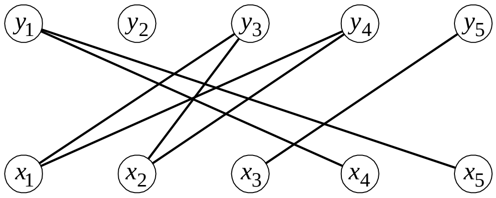

Fig. 2. To a partial graph of equalities links $\{x_1, y_4\}$, $\{x_2, y_4\}$ are added:

$$f(x_1) + f(y_4) = max\left\{d(x_1, y_j) \middle| j = \overline{1, 5}\right\}, \quad f(x_2) + f(y_4) = max\left\{d(x_2, y_j) \middle| j = \overline{1, 5}\right\}.$$

S1. As an initial matching can be chosen:

$$\pi_0 = \{(x_1, y_3), (x_2, y_4), (x_3, y_5)\}.$$

S2. Node x_4 unsaturated, $S = \{x_4\}$, $T = \varnothing$.

S3. $\Gamma_f(S) = (y_1)$.

S6. Choose unsaturated node y_1.

S7. Increasing way consists of one link: $P = \{(x_4, y_1)\}$.

Find $\pi_1 = \{(x_1, y_3), (x_2, y_4), (x_3, y_5), (x_4, y_1)\}$, $l = 1$.

S2. x_5 – unsaturated node, $S = \{x_5\}$, $T = \varnothing$.

S3. $\Gamma_f(S) = y_1$.

S6. Node y_1 saturated, its opposite number in π_1 – node x_4, $S = \{x_5, x_4\}$, $T = \{y_1\}$.

S3. $\Gamma_f(S) = f(y_1)$, $\Gamma_f(S) = T$.

S4. Find

$$d_f = min\ f\{(x_4) + f(y_2) - d_{42}, f(x_5) + f(y_2) - d_{52},$$

$$f(x_5) + f(y_3) - d_{53}, f(x_5) + f(y_4) - d_{54}\} = min\{7+0-5, 9+0-7, 9+1-2, 9+0-0\} = 2.$$

Define the following allowable node layout:

$f(x_1) = 12$, $f(x_2) = 13$, $f(x_3) = 15$, $f(x_4) = 5$, $f(x_5) = 7$;

$f(y_1) = 2$, $f(y_2) = 0$, $f(y_3) = 1$, $f(y_4) = 0$, $f(y_5) = 0$.

Partial graph of equalities G_f, which corresponds to obtained layout is depicted on fig. 3.

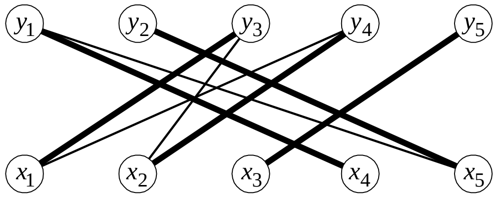

Fig. 3. Partial graph of equalities which contains the ideal matching

Starting with S1 we repeat actions as to finding an ideal matching in obtained partial graph of equalities. The algorithm finishes its work on S2 creating an optimal solution of a task $\pi^* = (3,4,5,1,2)$, $\Pi^* = (d_{13}, d_{24}, d_{35}, d_{41}, d_{52}) = (13,13,15,7,7)$,

$max\ C(\pi) = 55$. This repositioning provides the minimum of functional $\sum_{i=1}^{n} \left(d - d_{\pi[i]} \right)$.

We get $B\left(\pi^*\right) = 13+13+11+19+19 = 75$.

While recoursing the table $\left[d_{ij}^1\right]_5$, $\left[d_{ij}^2\right]_5$ from the matrix $\left[d_{ij}\right]_5$ it is clear that

$$d_{13} = d_{13}^1 = d_{13}^2,\ d_{24} = d_{24}^1,\ d_{35} = d_{35}^1,\ d_{41} = d_{41}^1,\ d_{52} = d_{52}^1 = d_{52}^2.$$

Hence, each of the sequences $\left(d_{13}^1, d_{24}^1, d_{35}^1, d_{41}^1, d_{52}^1\right)$, $\left(d_{13}^2, d_{24}^1, d_{35}^1, d_{41}^1, d_{52}^1\right)$, $\left(d_{13}^1, d_{24}^1, d_{35}^1, d_{41}^1, d_{52}^2\right)$, $\left(d_{13}^2, d_{24}^1, d_{35}^1, d_{41}^1, d_{52}^2\right)$ sets the minimum total run time of five routes in accordance with the schedule for the bus stations of points 1 and 2. In the first sequence duration of routes that begin and end in point 1 is listed, so that the traffic between the bus station is provided by a transport company, which is located at this point. In the fourth sequence there are lengths of routes d_{13}^2, d_{52}^2 from point 2 to point 1 and back from point 1 to point 2. In this case, the schedule should be provided with three buses of a company, situated in point 1 and two buses of a company situated in point 2.

Conclusions. The mathematical model of search *n* bus routes between two points and the algorithm of its solution, can increase the efficiency of the bus fleet by two transport companies.

This mathematical model in the future can be developed for *k* transport companies with regard to need to visit certain areas of roads. Using mathematical and algorithmic apparatus developed in [8–10], will allow more effective planning and make schedules and routes of public transport.

References:

1. Panishev, A.V. and Morozov, A.V. (2014), *Modeli i metody optimizatsii zamknutykh marshrutov na transportnoy seti*, ZhDTU, Zhitomir, 316 p.
2. Panishev, A.V. and Plechistyy, D.D. (2006), *Modeli i metody optimizatsii v probleme kommivoyazhera*, ZhDTU, Zhitomir, 300 p.
3. Morozov, A.V. (2015), «Pro matematychni modeli zadach klasiv lystonoshi ta komivojazhera», Visnyk ZhDTU, Vol. 2 (73), pp. 167–173.
4. Bronshteyn, E.M. and Zaiko, T.A. (2010) «Determenirovannye optimizatsionnye zadachi transportnoy logistiki», *Avtomatika i telemekhanika*, Vol. 10, pp. 133–147.
5. Maynika, E. (1981), *Algoritmy optimizatsii na setyakh i grafakh*, Mir, Moskva, 323 p.
6. Lovas, L. and Plammer, M. (1998), «Prikladnye zadachi teorii grafov». *Teoriya paroso-chetaniy v matematike, fizike, khimii*, Mir, Moskva, 653 p.
7. Papadimitriu, Kh. and Stayglits, K. «Kombinatornaya optimizatsiya. Algoritmy i slozhnost'», Mir, Moskva, 510 p.
8. Matsiy, O.B., Morozov, A.V. and Panishev, A.V. (2015), «The Recurrent Method to Solve the Assignment Problem», *Cybernetics and Systems Analysis*, Vol. 51, issue 6, pp. 939–946.

9. Matsiy, O.B., Morozov, A.V. and Panishev, A.V. (2016), «Fast Algorithm to Find 2-Factor of Minimum Weight», *Cybernetics and Systems Analysis*, Vol. 52, issue 3, pp. 467–474.
10. Matsiy, O.B., Morozov, A.V., Panishev, A.V. (2016), «A Recurrent Algorithm to Solve the Weighted Matching Problem», *Cybernetics and Systems Analysis*, Vol. 52, issue 5, pp. 748–757.

MOROZOV Andrii Vasyliovych – the dean of the faculty of information computer technologies, Associate Professor of department of computer engineering of Zhytomyr State Technological University.

Research interests:

– combinatorial optimization;

– distributed and parallel systems;

KUSHNIR Nadia Oleksandrivna – Ph.D. student of the department of software systems of Zhytomyr State Technological University.

Research interests:

– discrete and combinatorial optimization.

LOKTIKOVA Tamara Mykolaivna – senior lecturer of department of computer systems management and automation of Zhytomyr State Technological University.

Research interests:

- digital image processing and pattern recognition;

- combinatorial optimization.

The article was sent to the publishing department on 02.09.2016.