УПРАВЛІННЯ У ТЕХНІЧНИХ СИСТЕМАХ

CONTROL IN TECHNICAL SYSTEMS

UDK 519.853: 658.52

METHOD OF ROUTING A GROUP OF MOBILE ROBOTS IN A FIXED NETWORK FOR SEARCHING THE MISSING OBJECTS IN A TECHNOLOGICAL DISASTER ZONE

Batsamut V. M. – Dr. Sc., Professor, The Deputy Head of the Scientific Research Center of the National Academy of the National Guard of Ukraine, Kharkiv, Ukraine.

Hodlevsky S. O. – The Researcher of the Scientific Research Center of the National Academy of the National Guard of Ukraine, Kharkiv, Ukraine.

ABSTRACT

Context. The relevance of the article is determined by the need for further development of models of collective behavior of systems with multi-agent structure construction endowed with intelligence that ensures synchronization of the joint efforts of various agents while achieving the goals set for the system. The method proposed in the article solves the problem of competition between different agents of a multi-agent system, which is important while performing search, rescue, and monitoring tasks in crisis areas of various origins.

Objective is to develop a method for determining the sufficient population of a multi-agent system and the optimal routes of movement of its individual elements in a stationary network for the most complete examination of a technological disaster zone (any given zone based on a certain transport network).

Method. We implemented the concept of a dynamic programming to search for all possible edge-simple longest paths connecting the directed subsets of vertices-sources and vertices-sinks in the structure of the model weighted directed graph. To this end, the modified Dijkstra method was applied. The modification comprises representing the weights of the arcs of the modeling directed graph with the negative values, which are further used in calculations according to the Dijkstra method. After finding the next edge-simple longest path, the arcs that make up it are fixed in the memory of the computer system (in the route plan) and removed from the graph structure, and the process is iteratively repeated. The search for paths takes place as long as the transitive closure between the vertices that are part of the specified subsets of source vertices and sink vertices is preserved. The developed method makes it possible to find such a set of traffic routes for the elements of the multi-agent system, which maximizes the area examined by them in a technological disaster zone (or the number of checked objects on the traffic routes) in one "wave" of the search and distributes the elements of a multi-agent system by routes that do not have common areas. A derivative of the application of the developed method is the determination of a sufficient population of a multi-agent system for effective search activities within the defined zone.

Results. 1) A method of routing a group of mobile robots in a stationary network for searching the missing objects in a technological disaster zone has been developed. 2) The working expression of the Dijkstra method for searching in the structure of a network object (in the structure of a model graph) for the longest paths has been formalized. 3) We have suggested a set of indicators for a comprehensive evaluation of route plans of a multi-agent system. 4) The method has been verified on test problems.

Conclusions. Theoretical studies and several experiments confirm the efficiency of the developed method. The solutions made using the developed method are accurate, which allows recommending it for practical use in determining in an automated mode route plans for multi-agent systems, as well as the required number of agents in such systems to perform the required amount of search tasks in a particular crisis area.

KEYWORDS: multi-agent system, group of mobile robots, routing, network object, weighted undirected (directed) graph, extreme paths, optimization criterion, method.

ABBREVIATIONS

MAS is a multi-agent system;

TDZ is a technological disaster zone.

NOMENCLATURE

G is a undirected weighted graph simulating the transport network in a technological disaster zone;

$\vec{G}$ is a directed weighted graph simulating the transport network in a technological disaster zone;

P is a set of graph vertices simulating turning points (intersections) of the transport network within a technological disaster zone;

A is a subset of vertices-sources such that $A \subset P$;

B is a subset of vertices-sinks such that $B \subset P$;

E is a set of edges (arcs) of the graph modeling paths (communications) inside a technological disaster zone;

$w(e_{ij})$ is a weighting coefficient of some edge (arc) e_{ij};

$Direct$ is a general search direction;

DOI 10.15588/1607-3274-2023-1-13

k_{wvs} is a number of searching waves;

M_i is the longest edge-simple path between some vertices a_x and b_y from sets A and B respectively;

$\vec{G}'$, $\vec{G}''$,…, $\vec{G}^n$ is a sequence of substructures arising because of splitting the initial graph $\vec{G}$;

n is the number of found longest edge-simple paths between subsets A and B respectively;

L_i is the length of *i*-th edge-simple longest path;

i is the number of the edge-simple longest paths;

L_{total} is the total length of the transport network in the TDZ;

L_{PL} is a total length (weight) of the defined routes;

T_s is a total time of conducting search activities;

$\bar{\nu}_{agt}$ is the average speed of the agents moving along the defined routes;

T_{dir} is an established (directive) time for performing search activities;

K_{MAS} is the required number of MAS agents;

P_{dso} is a probability of detecting search objects according to a certain route plan of the MAS;

$\xrightarrow{T}$ is a transitive closure between the selected pair of vertices of the model graph $\vec{G}$.

INTRODUCTION

In the event of a large-scale technological disaster of a certain origin in any region of the country, which could be accompanied by the release of various types of poisonous or ionizing substances, a task of searching the missing objects and/or monitoring the operational (chemical, radiation) situation in a technological disaster zone (TDZ) may arise. In this case, the bodies for obtaining data on operational situation in such a zone can be both chemical or radiation reconnaissance groups, staffed with trained personnel and appropriate special equipment, protective gear and devices, and the groups of mobile robots equipped with appropriate devices for monitoring, measuring, and recording the values of individual parameters, as well as mixed groups.

The set of such bodies constitutes a multi-agent system (MAS), which in order to perform practical missions (achieving the goals) must have a common logic of behavior, as well as the logic of conduct of its individual agents [1, 2]. It should be noted here that the struggle to preserve the life and health of rescue and search units' personnel has recently outlined a global trend towards the use of exclusively robotic MAS in dangerous zones [3, 4]. A human in such a system endows it with its intelligence, performs control, logistical support of the system and is a consumer of the results of its functioning. The materials of this article are aimed at ensuring the functioning of robotic MAS.

Endowing the MAS with a certain behavior (intelligence) in matters of rational or optimal movement of its elements through the network structure while performing search (or monitoring) task, requires the development of certain routing methods.

In general, the problem of routing in any network object is to find some extreme paths (shortest or k-shortest, Hamiltonian) in its structure [5, 6, 7], less often trees [8] or minimal covering trees [9], etc. In this matter, the nature of the applied problem to be solved is decisive.

While search activities using a robotic MAS, the application of the above approaches to the routing of its agents is unacceptable, since:

1) Various conditions in a certain way determine the available points of input of MAS elements in the search and the number of such points, which in turn determines the general direction of the search within TDZ.

2) In order to achieve the search goals, it is necessary to check as many areas as possible (in the best case, all areas, although it is practically impossible to fulfill this requirement in one "pass" of the MAS in dense structures).

3) It is necessary that the same areas of the terrain are not revisited by different agents (implementation of the "checked and crossed out from the list" principle), which will increase the chances of the positive result of a search operation.

4) It is desirable to withstand the general direction of advancement of the entire MAS within the network object (a technological disaster zone).

The object of the study is to determine the set of optimal movement routes for the elements of a multi-agent system within the network object.

The subject of the study is the method of optimal routing of a group (swarm) of mobile robots within the structure of a fixed network for conducting search activities in a limited area.

The purpose of the study is to develop a method for determining the sufficient population of a multi-agent system and the optimal movement routes of its individual agents in a stationary network for the most complete examination of a technological disaster zone (any given zone based on a certain transport network).

1 PROBLEM STATEMENT

Routing problems are, for the most part, formalized and solved using graph theory models and methods [10]. That is why we will model the transport network in TDZ with some weighted undirected graph $G=(P,E)$, where $P=\{p_1,...,p_v\}$ – the set containing the vertices of the graph; $E=\{e_1,...,e_m\}$ – the set containing the edges of the graph; v – the number of graph vertices; m – the number of edges of the graph. At the same time, the set P is simulating the intersections of the transport network (points at which the agent decides about the direction of its further movement), and the set E – certain sections of paths connecting different intersections in the transport network of the TDZ. Each edge from the set E will have a certain weight coefficient $w(e_{ij})$, which will quantita-

DOI 10.15588/1607-3274-2023-1-13

tively characterize the edge e_{ij}. It could be a certain number of objects to be inspected located on this site; the length of the route measured in certain units; the amount of work to be done on the site, etc.

Considering the peculiarities of carrying out search activities in a certain area, the formulation of the task of determining traffic routes for the MAS will look as follows.

Assume that the undirected weighted graph $G=(P,E)$ has selected subset of vertices $A=\{a_x\}_{x=\overline{1,k}}$ (vertices-sources) and a subset of vertices $B=\{b_y\}_{y=\overline{1,z}}$ (vertices-sinks), where k and z are the numbers of such vertices, respectively, and $B \notin \varnothing$, $|A| << |P|$, $|B| << |P|$, $A \cap B = \varnothing$. Weighting coefficients $w(e_{ij})$ on the edges of the graph $G=(P,E)$ quantitatively characterize the corresponding area of the TDZ.

It is necessary to find all possible edge-simple longest paths $M_{i=\overline{1,n}}=(a_x, p_1, p_2, ..., p_q, b_y)$, connecting vertices from subsets A and B, and the totality of which satisfies the following target function:

$$F=\sum_{i=1}^{n} L_i \to \max, \qquad (1)$$

under the conditions:

$$T_s \le T_{dir}, \qquad (2)$$

$$n = f\left[G(P,E), |A|, |B|, Direct\right], \qquad (3)$$

$$e_{ij} \in M_1 \cap e_{ij} \in M_2 \cap ... \cap e_{ij} \in M_n = \varnothing, \qquad (4)$$

$$k_{wvs} = 1. \qquad (5)$$

Therefore, the solution of the problem (1)–(5) requires searching within the structure of the initial weighted graph $G=(P,E)$ of all extremal (edge-simple longest) paths between defined subsets of its vertices.

An edge-simple path is a path in which each edge (arc) of the graph occurs only once [10].

2 REVIEW OF THE LITERATURE

At present, the theoretical basis for the functioning of robotic vehicles is the theory of algorithms. The development of the theory of algorithms takes place in two directions: first, the expansion of the range of practical problems solved by the existing algorithms; secondly, the development and improvement of algorithms for solving new problems that arise during the creation and functioning of the MAS. Well-developed graph theory plays a leading role in routing of individual MAS agents.

A well-known and studied problem of graph theory having numerous practical applications is the problem of Hamiltonian paths, that is, whether there is a simple path in a graph in which each vertex of the graph occurs exactly once. Such a graph is called Hamiltonian. In the case when the graph does not contain a Hamiltonian path, in some applications it makes sense to search for a path of a maximum length in the graph. Finding such a path is known as the longest path problem. Like finding the Hamiltonian path, finding the longest path is also a difficult task.

The longest path problem is NP-complete on every class of graphs in which the Hamiltonian path problem is also NP-complete. Thus, in [11] the authors prove that even if a graph has a Hamiltonian path, the problem of finding a path of the length $n-n^{\varepsilon}$ for some $\varepsilon < 1$ is a NP-complete, where n – is the number of vertices of the initial graph. The authors claim that there is no polynomial approximation algorithm with a constant factor for the longest path problem unless $P = NP$ [11]. Similar research results are also given in works [12, 13, 14, 15, 16].

It should be noted here that the Hamiltonian path problem is NP-complete on general graphs [17, 18] and remains NP-complete even when restricted to some small classes of graphs, such as splitting graphs [19], chordal bipartite graphs, strongly chordal graphs [20], directed path graphs [21], circular graphs [22], planar graphs [18] and grid graphs [23].

Polynomial solutions of this problem are known only for certain classes of graphs. Such algorithms were developed for graphs of intervals and presented in studies [24, 25, 26, 27], for doubly convex graphs – in a study [28], for graphs of arcs of circles – in a study [27] and graphs of comparability – in a study [29].

Unlike the Hamiltonian path problem, several polynomial complexity algorithms are known for the longest path problem that work with the structures of tree-type and some classes of graphs. A linear algorithm for finding the longest path on a tree-structure was proposed by Dijkstra in 1960, a formal description of which can be found in [30]. Later, based on the results of improving the Dijkstra algorithm for trees, the authors of [31] have solved the problem of finding the longest path for weighted trees and block graphs with a linear calculation time, and for cactuses with a polynomial calculation time – $O(n^2)$, where n – the number of vertices of the initial graph. Recently, polynomial algorithms were proposed that solve the problem of finding the longest path on bipartite graphs with computational complexity $O(n)$ [32], on Ptolemaic graphs with computational complexity $O(n^5)$ [33]. In [34], the authors presented their polynomial algorithm for interval graphs, which is based on the idea of dynamic programming and has a computational complexity of $O(n^4)$. In [35], the authors proposed a polynomial algorithm in which they also used the dynamic programming approach but applied lexicographic search in depth (the so-called LDFS graph routing) for co-comparable graphs. The computational complexity of such an algorithm is also limited to $O(n^4)$.

The analysis of the literature shows that the problem in the formal statement (1)–(5) was not posed or solved

DOI 10.15588/1607-3274-2023-1-13

by anyone. If the approach is not strict, then the task is to find in the structure of the model weighted graph $G=(P,E)$ every (or k, where k>0) of the longest edge-simple paths between the selected subsets of the vertices of the graph, which lie on opposite edges of it and the total sum of the weights of which (paths) maximizes the target function F, expression (1). At the same time, an additional and mandatory condition for the sought route plan is the requirement that there are no common edges in different paths (routes). Our article is dedicated to solving this problem.

3 MATERIALS AND METHODS

It is quite clear that the solutions of the problem (1)–(5) formulated above require not only estimates of the lengths of the critical paths, but also the critical paths themselves, that is a consecutive set of edges that make them (the paths). It should be noted that such identification capabilities, in contrast to the well-known in graph theory algorithms Floyd-Warshall [36, 37, 38], Shimbel [39], Danzig [5] and some others, which solve the problem of finding the lengths of extreme paths, the Dijkstra's algorithm provides.

In addition, to solve the problem (1)–(5), it is necessary to use the modified Dijkstra algorithm, since it (the problem) in this formulation belongs to the class of NP-complete and, therefore, it cannot be solved in polynomial time [41].

Modification of the initial structure. The problem of NP-completeness of the task of finding the longest paths (paths of the greatest total weight) in the structure of a network object is associated with the possible presence of cycles, which will lead to an unjustified increase in the total weight of the searched path (so-called "looping"), and therefore to the inability to adequately identify the path itself in the future. To solve such a "problem" when determining maximum (search for the longest paths), usually the initial undirected graph $G=(P,E)$ is represented in the form of an oriented graph $\vec{G}=\left(P,\vec{E}\right)$ without cycles. For this, the edges of the graph are directed along the general search direction, which is determined based on the nature of the practical problem to be solved (see Fig. 1). As a result of this transformation, the edges of the graph become arcs, that is, they get directionality.

Since a group of mobile robots during search activities moves from one side of the TDZ to the opposite (which can be imagined in the form of a "wave" that passes through this zone), then such a general direction also exists.

As can be seen from Fig. 1, there are no cycles in the structure of the oriented graph, which makes it possible to apply the modified Dijkstra algorithm to it to find the longest paths (paths of the greatest total weight).

Modification of initial data. The next problematic point that needs to be solved is the representation of the weight coefficients of the edges when solving the problem of maximizing the lengths of the routes by which the elements of the MAS advance (search for the longest paths). This problem appears because in Dijkstra's algorithm, at each iteration, the value of the accumulated total weight for some vertex is compared with the estimate of the total weight that this vertex can receive through another edge (arc) and the smallest of these values is chosen. The working expression of Dijkstra's algorithm in its classical form is as follows [5]:

$$d(x):=\min\{d(x),\, d(y)+w(y,x)\}. \tag{6}$$

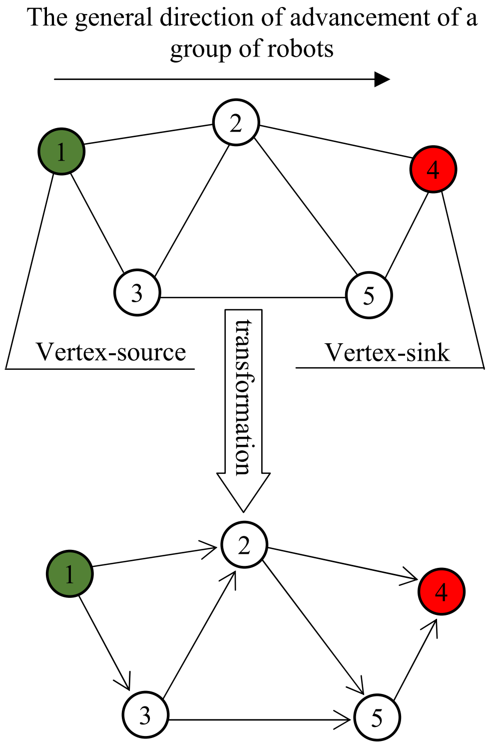


Figure 1 – Transformation of an unoriented graph $G=(P,E)$ into oriented $\vec{G}=\left(P,\vec{E}\right)$

Since the problem (1)–(5) is solved for the maximum of the function, the application of the classic Dijkstra algorithm according to the expression (6) will lead to incorrect results – the search for the shortest paths. To solve such a problem, we suggest representing the weighting coefficients of the arcs of the graph $\vec{G}=\left(P,\vec{E}\right)$ as numbers from the negative domain and applying the following modification of the working expression of the algorithm, namely:

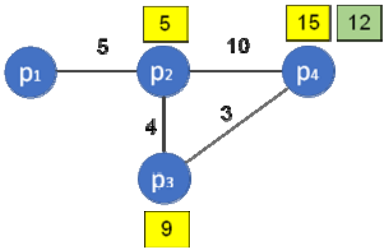
$$-d(x):=\min\{-d(x),\, -d(y)+(-w(y,x))\}. \tag{7}$$

As a result of applying expression (7), the current negative score will be compared with an alternative also negative score, and if the alternative score is lower, then

© Batsamut V. M., Hodlevskyi S. O., 2023
DOI 10.15588/1607-3274-2023-1-13

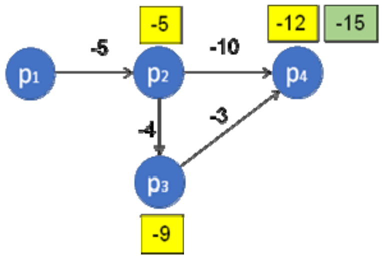
the current vertex will be assigned exactly that score. In other words, if in the structure of a weighted directed graph it is possible to increase the length of the current path due to the addition of a certain arc with a negative weight, then such an arc will be added, which will ensure that the longest path is found.

It should be noted here that estimates of path lengths will be presented in negative form, which will in no way interfere with their subsequent identification. In the future, the obtained estimates will be taken by the module, which will return them to their original physical meaning, Fig. 2.



$M_1=\{p_1, p_2, p_3, p_4\}=12$

a



$M_2=\{p_1, p_2, p_4\}=15$

b

Figure 2 – The result of applying classical and modified Dijkstra methods to find the extreme paths: a – finding the shortest path M_1 (classical method); b – finding the longest path M_2 (modified method)

As can be seen from Fig. 2, *b* that between the vertices p_1 and p_4 there are two alternative ways: $\{p_1, p_2, p_3, p_4\}$ and $\{p_1, p_2, p_4\}$. At the vertex p_4 you need to decide which of these paths is considered the longest. Two arcs are incidental to it: (p_3, p_4) and (p_2, p_4). Since the vertex p_3 has a label of –9 (the total weight of the path connecting this vertex to the source vertex p_1), the vertex p_4 through the arc (p_3, p_4) will receive a label of –12. Through an incident arc (p_2, p_4) having weight of –10, the vertex p_4 will receive a label of –15, because the vertex p_2 already has a label of –5. Therefore, the length of the path $\{p_1, p_2, p_3, p_4\}= -12$, and the path $\{p_1, p_2, p_4\}= -15$. Based on expression (7), the second of them is unambiguously chosen as the longest path. In the future, taking the received estimate of the length of the path by the module, we consider 15 units of the conditional length of the path.

On more branched structures, the process continues until all arcs are analyzed, and all vertices (including vertices-sinks) receive the corresponding conditional weight labels according to the expression (7). The identification of the extreme paths takes place on the reverse course, just as in the classic Dijkstra method.

Application of dynamic programming. The characteristics of edge-simple extreme paths are of particular importance in the context of search activities in TDZ, since it is undesirable for different agents of the MAS to move along paths that have common sections (arcs), even if the number of such sections is relatively small. In this sense, the movement of different MAS agents needs to be "separated" and redirected by different ways to increase the overall effectiveness of search activities (Fig. 3). The only place where the routes of different agents can intersect is the intersection (vertices in the structure of the initial model graph), which in no way hinders their simultaneous progress along the assigned routes and, in this part, does not reduce the overall effectiveness of search activities.

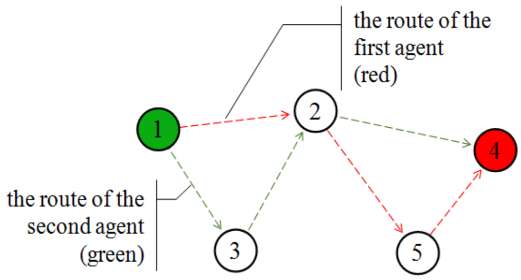


Figure 3 – Separated traffic routes of various MAS agents (option)

When solving the problem (1)–(5), to obtain a set of routes that do not have common sections (according to the additional condition (4) of the given problem) we propose to use the method of dynamic programming, which is to divide some general problem $\hat{Z}$ on a number of subproblems $Z', Z'',..., Z^n$, where n – the number of such subproblems, and to find such their corresponding solutions $R', R'',..., R^n$, that $R' \cup R'' \cup ,..., \cup R^n$ will be the solution to a general problem $\hat{Z}$. At the same time, it is considered if separate solutions $R', R'',..., R^n$, are rational (or optimal), then the solutions to the general problem $\hat{Z}$ are also rational (or optimal).

Using dynamic programming approaches and solving the problem of finding all edge-simple longest paths, we propose to split the structure of the initial graph $\vec{G}$ into a sequence of substructures $\vec{G}'$, $\vec{G}''$,…, $\vec{G}^n$ and search for the appropriate junctions (routes) $M_1, M_2,..., M_n$ on each of them. The set of such routes will make up the general route plan for the MAS in a technological disaster zone.

Splitting the initial graph $\vec{G}$ into a set of substructures $\vec{G}', \vec{G}'',..., \vec{G}^n$ will be based on the iterative re-

DOI 10.15588/1607-3274-2023-1-13

moval from the current substructure of the arcs included in the composition of the extreme path found on this substructure.

The process of splitting the initial graph $\vec{G}$ will continue until there remains no path (transitive closure [42]) connecting the vertices from the sets A and B (see the statement of the problem), or until the fulfillment of another (additional) condition at the choice of the decision maker.

The developed method is structurally presented in Fig. 4.

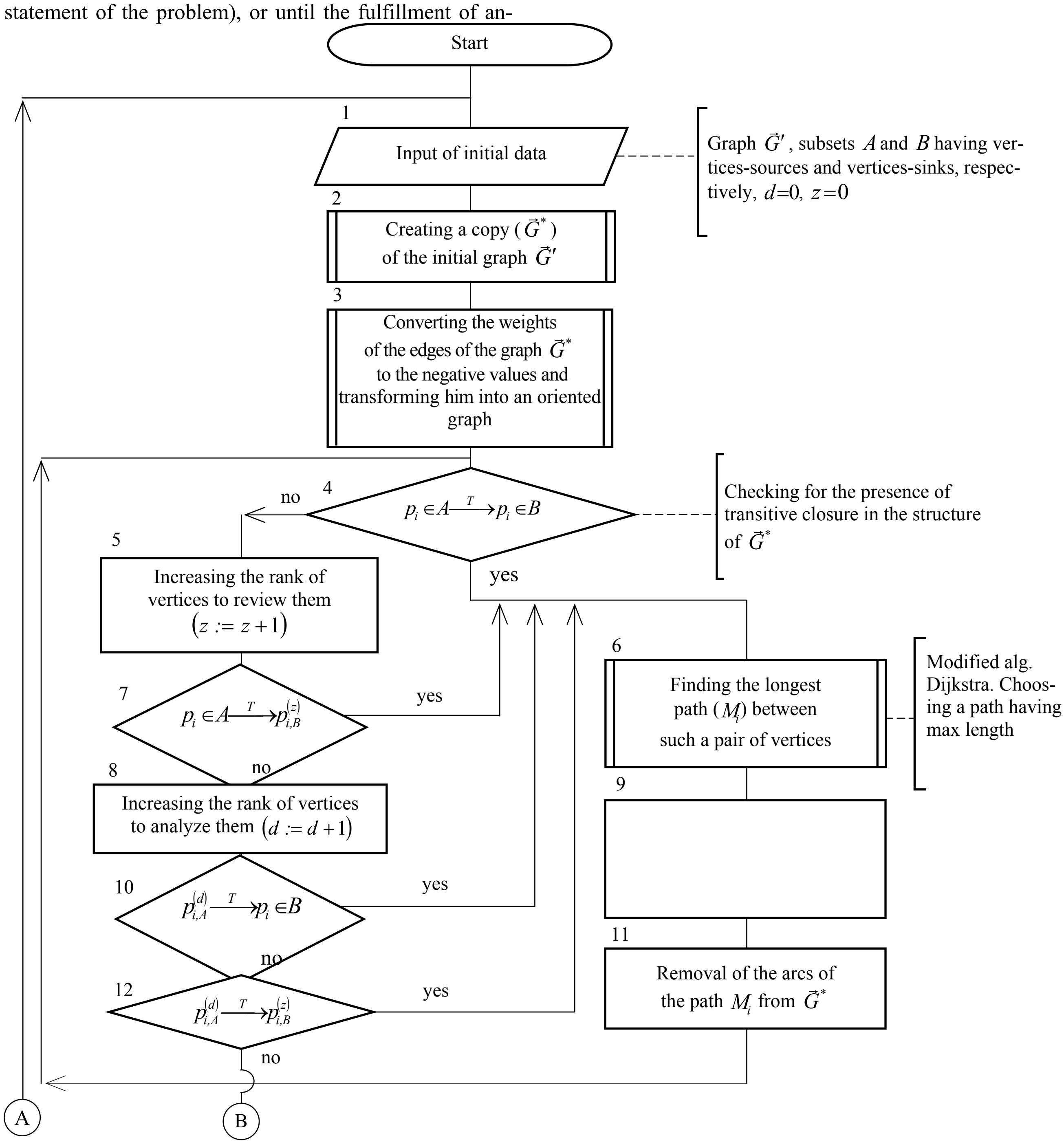


Figure 4 – Block diagram of the method of routing a group of mobile robots on a fixed network to search the missing objects in a technological disaster zone

DOI 10.15588/1607-3274-2023-1-13

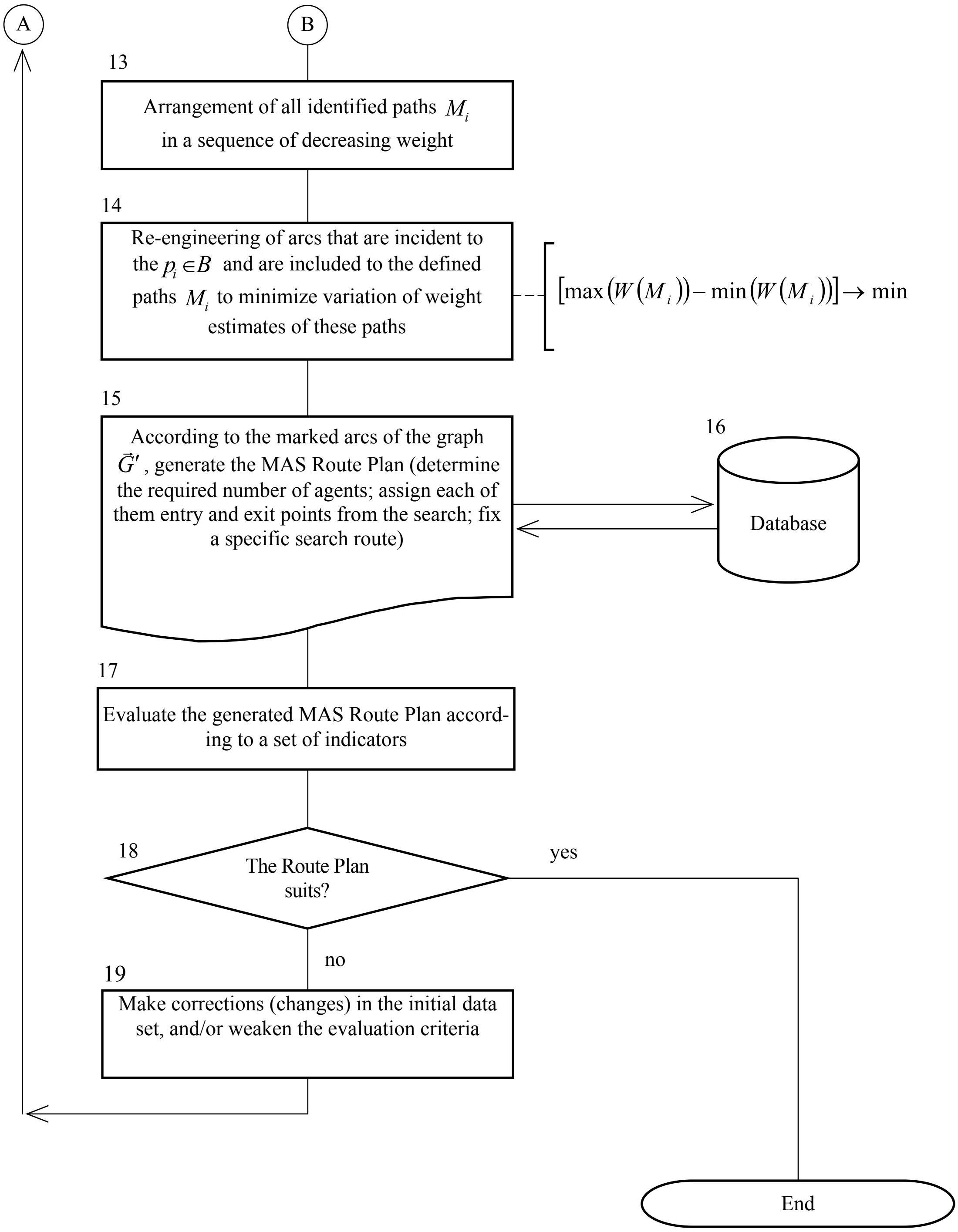


Figure 4, sheet 2

4 EXPERIMENTS

Let's assume that the TDZ transport network is modeled as weighted G' (Fig. 5).

Let's give the edges of the graph a general direction from the vertex p_5 towards the opposite edge of the graph. Thus, subset A will contain only one vertex p_5. We assume the subset B consists of vertices p_3 and p_{10}. Let's represent the weights of the edges as negative values.

Step I. Let's find the longest path from the vertex p_5 to any vertex from the subset B. The current weight estimates obtained due to the modified Dijkstra method and the identified paths themselves are presented in Fig. 6.

DOI 10.15588/1607-3274-2023-1-13

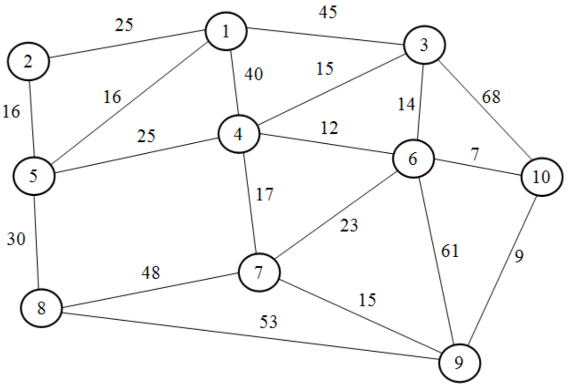


Figure 5 – Weighted graph $\vec{G}'$ simulating the transport network in a technological disaster zone (the weights of edges are conditional lengths of the corresponding communications)

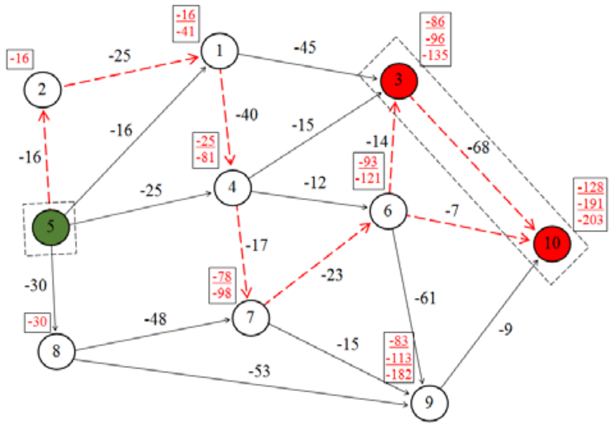


Figure 6 – Initial weighted directed graph $\vec{G}'$

We obtained two paths (routes) to the vertices from the subset B, namely: $M_1=\{p_5,p_2,p_1,p_4,p_7,p_6,p_3,p_{10}\}$ and $M_2=\{p_5,p_2,p_1,p_4,p_7,p_6,p_{10}\}$. The absolute weight of route M_1 is greater than that of route M_2, because (|–203|>|–135|). Therefore, the route M_1 is included in the MAS route plan.

Step II. Subsequently, all arcs that make up the route M_2 are removed from the structure $\vec{G}'$. Along with them, all hanging vertices arising from the removal of arcs are removed. As a result, we get the following structure $\vec{G}''$ on which the next longest paths are to be searched (see Fig. 7).

After applying the modified Dijkstra method to the structure of $\vec{G}''$, two paths (routes) to the vertices from the subset B are obtained, namely: $M_3=\{p_5,p_1,p_3\}$ and $M_4=\{p_5,p_4,p_6,p_9,p_{10}\}$. The absolute weight of route M_4 is greater than that of route M_3, because $|-107|>|-61|$. Therefore, route M_4 is included in the MAS route plan.

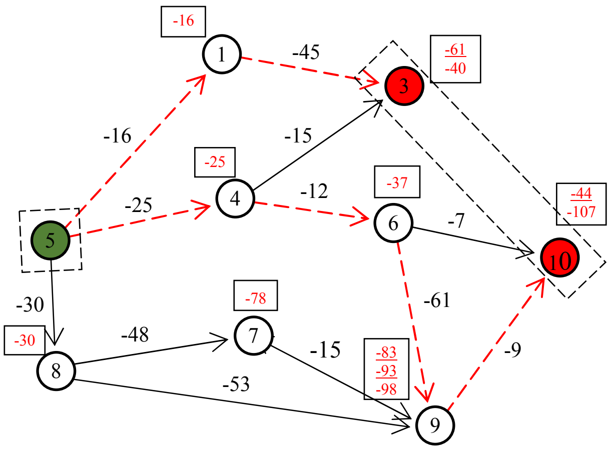


Figure 7 – Weighted graph $\vec{G}''$

Step III. All arcs that make up the route M_4 are to be removed from structure $\vec{G}''$. All hanging vertices are also to be removed. We obtain the following structure $\vec{G}'''$, on which the longest path to any vertex from subset B is also being searched (see Fig. 8). Such a route is the only route $M_5=\{p_5,p_1,p_3\}$ and its weight is |–61|. Route M_5 is to be included into the MAS route plan.

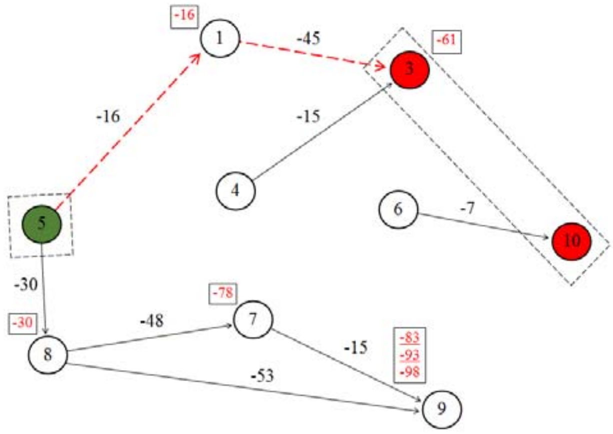


Figure 8 – Weighted graph $\vec{G}'''$

Step IV. All arcs that make up the route M_5 from the structure of $\vec{G}'''$ are to be deleted. Hanging vertices are also removed. As a result, we get the following structure of $\vec{G}''''$ (see Fig. 9). We can see from the figure that in this structure there are no paths connecting vertex p_5 to any vertex from subset B. So, based on the results of this step, the problem (1)–(5) has been basically solved. On the initial structure, all edge-simple longest paths between the defined sets of vertices in the structure of the initial network object $\vec{G}'$ have been determined. We will name such paths as first-level paths because they directly connect vertices from subsets A and B.

DOI 10.15588/1607-3274-2023-1-13

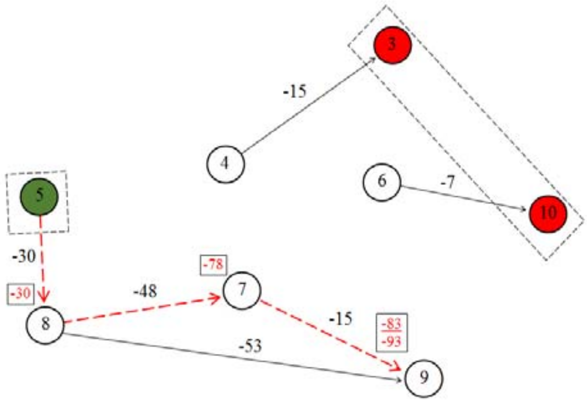


Figure 9 – Weighted graph $\vec{G}''''$

However, vertex p_5 has another unused incident arc (p_5, p_8). Let's find the maximum path to any vertex (in this case, to one that does not belong to subset B). We will consider such paths as paths of the second level: they connect the initial vertex-source with some vertex that is adjacent to some vertex from the subset B. Such paths can also be included into the MAS route plan, thereby increasing its efficiency.

In structure $\vec{G}''''$, such a route is the only route $M_6 = \{p_5, p_8, p_7, p_9\}$ with total weight |–93|. Route M_6 is also included into the MAS route plan.

So, as a result of successive splitting of the initial graph $\vec{G}'$, namely: $\vec{G}' \rightarrow \vec{G}'' \rightarrow \vec{G}''' \rightarrow \vec{G}''''$, four edge-simple longest paths (those that do not have mutual arcs) were found in its structure, namely: $M_1 = \{p_5, p_2, p_1, p_4, p_7, p_6, p_3, p_{10}\}$; $M_4 = \{p_5, p_4, p_6, p_9, p_{10}\}$; $M_5 = \{p_5, p_1, p_3\}$; $M_6 = \{p_5, p_8, p_7, p_9\}$. The paths and their absolute weights are presented in the figure (Fig. 10).

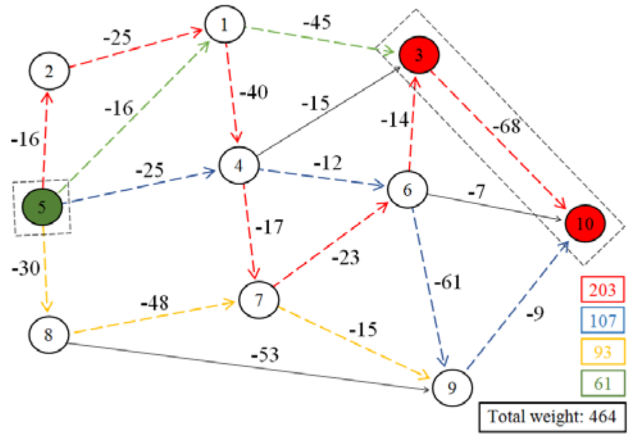


Figure 10 – The set of searched edge-simple longest paths in the structure of the initial network object $\vec{G}'$ (marked with different colors)

5 RESULTS

The scheme of routes presented in Fig. 10, contains the following information:

– to carry out search activities within the TDZ, whose transport network is modeled by graph $\vec{G}'$, it is sufficient to have four agents as part of the MAS.

– for each MAS agent, its starting point (boundary) of entering the search and the final point (boundary) of exiting the search are defined.

– each MAS agent within the TDZ is assigned a specific search (movement) route.

– routes of movement of various MAS agents do not cross sections (arcs).

– in total, the traffic routes of the MAS are optimized both in terms of the direction of traffic and the length of the routes.

The assessments of the developed MAS route plan, based on the totality of the proposed indicators, are presented in Table 1.

Table 1 – Assessments of the developed MAS route plan according to a set of indicators

№	Indicator (parameter)	Indicator value
1	Required number of MAS agents, K_{MAS}, (units): $$K_{MAS} \cong n,$$ where n – the number of defined (selected) search routes	4
2	MAS route plan, PL_{MAS}: $$PL_{MAS} = \{M_1, M_2, ..., M_n\},$$ where n – the number of defined (selected) search routes	$M_1 = \{p_5, p_2, p_1, p_4, p_7, p_6, p_3, p_{10}\}$; $M_4 = \{p_5, p_4, p_6, p_9, p_{10}\}$; $M_5 = \{p_5, p_1, p_3\}$; $M_6 = \{p_5, p_8, p_7, p_9\}$

DOI 10.15588/1607-3274-2023-1-13

Table 1 – Assessments of the developed MAS route plan according to a set of indicators (continuation)

№	Indicator (parameter)	Indicator value
3	Route length, L_i, (km)	$L_1 = 203$; $L_4 = 107$; $L_5 = 61$; $L_6 = 93$
4	Total length (weight) of routes, L_{PL}, (km): $$L_{PL} = \sum_{i=1}^{n} L_i ,$$ where n – the number of defined (selected) search routes	464
5	Assessment of the search time in TDZ, T_s, (hrs.): $$T_s = \frac{\max\{L_i\}}{\bar{\nu}_{agt}},\ i = 1...n ,$$ where $\bar{\nu}_{agt}$ – the average speed of the agents moving along the defined routes, km/h.	If: $\bar{\nu}_{agt} = 5$ km/h. $T_s \approx 40{,}6$ hours
6	Probability of finding search objects in case of implementation of the MAS route plan, P_{dso}: $$P_{dso} = \frac{L_{PL}}{L_{total}} ,$$ where L_{total} – the total length of the transport network in the TDZ	If: L_{total} =539 km P_{dso} =0,86
7	Probability of finding search objects within the prescribed time for conducting search activities, $P_{dso}(T_s \le T_{dir})$: $$P_{dso}(T_s \le T_{dir}) = 1 - e^{\left(-\frac{T_{dir}}{T_s}\right)} ,$$ where T_{dir} – the prescribed time	If: T_{dir} =30 hours $P_{dso}(T_s \le T_{dir})$ =0,52

6 DISCUSSION

The set of approaches proposed in the article, together with the modified Dijkstra's method, allowed to develop a method that searches in the structure of any network object, between subsets of its vertices, all existing edge-simple longest paths, which allows to organize the routing of MAS agents in a certain area by paths that do not intersect their individual sections.

At the same time, it is worth noting that the structure of the paths themselves, as well as their number, will depend on the defined (prescribed) points of entry into the search and points of withdrawal from the search, which allows, when planning search activities, to quickly consider several options for future actions and choose the most appropriate one (to order the options by the degree of their advantage).

In addition, having carefully studied the scheme of routes presented in Figure 10, it is possible to conclude that it is expedient to reengineer the routes to reduce the variation of values by their absolute weight (see Fig. 4, Block 14). The idea of reengineering is as follows. If there is an arc (edge) connecting them between any pair of vertices from a subset B, and this arc (edge) is included in some extreme path, and this path has the maximum (higher) weight score among other paths, then this arc (edge) should be included in the path with the minimum (lower) weight. It should be noted that based on the results of using the proposed method, the minimally required number of MAS agents to conduct search activities in an area with a certain transport network could be found.

It is obvious that in the real structures, and hence in the structures which are denser and having more vertices, the number of iterations to split the initial structure and, accordingly, to find the edge-simple longest path, can be close to the very dimension of the initial structure itself. Therefore, the computational complexity of the combinatorial algorithm that implements the developed method will be determined by the computational complexity of its "basic element" – the Dijkstra's algorithm, which is estimated as $O(n^2)$ [40], as well as the computational complexity of the algorithm that determines the presence of transitive closure between vertices from subsets A and B, which is evaluated as $O(n^3)$ [41]. Considering the strengths of subsets A and B, the total computational complexity of the combinatorial algorithm could be estimated as $O(|A| \cdot |B| \cdot (n^3 + n^2))$, where $|A|$ and $|B|$ – the strengths of the corresponding subsets; n – the number of vertices of the model graph G.

The obtained polynomial estimate of computational complexity is quite acceptable for the use of such an algorithm in a real-time framework.

CONCLUSIONS

DOI 10.15588/1607-3274-2023-1-13

The article solves the actual scientific and applied problem of finding all possible edge-simple longest paths between defined subsets of vertices of an initial undirected graph.

The scientific novelty of the developed method is as follows:

1) the representation of the weights of the edges of the initial model graph by the negative values, which allows finding the longest paths between a specified pair of vertices using the classical Dijkstra method;

2) the application of the dynamic programming method, which makes it possible to find the longest paths $M_1, M_2,...,M_n$ in the set of obtained substructures $\vec{G}', \vec{G}'',..., \vec{G}^n$ of the initial model graph $\vec{G}$, which will constitute the complete combination of the edge-simple longest paths.

Since the basic element of the developed method is the Dijkstra's method, which belongs to the accurate class, it can be assumed that the developed method is also accurate.

The practical value of the method is that its application significantly simplifies the process of developing effective route plans for the elements of the MAS in a particular transport network. The projection of general theoretical and methodological statements and conclusions made during the study on the problem of routing of the MAS elements in a certain area, allows to make the search activities effective and quickly develop several options for search actions. The developed method will have a practical use if it is implemented based on a geoinformation system.

A promising direction for further research is the development of routing methods of the MAS performing search activities in a non-stationary network, as well as the development of a set of indicators and criteria for prompt decision-making regarding the optimal (rational) bypassing by agents of the obstacles that suddenly appear on the route path.

ACKNOWLEDGEMENTS

This article highlights one of the results obtained by the authors in 2020–2021 while implementing the research project (state registration number 0120U002173) at the Research Center of the National Academy of the National Guard of Ukraine. The authors are grateful to their colleagues for their support during the research and active participation in the discussion of the results. All authors declare that they have neither financial support nor obligations.

REFERENCES

1. Konovalenko O., Brusentsev V. Multi-agent management and decision support systems, *Bulletin of the National Technical University "KhPI". Ser.: Mechanical engineering and CAD,* 2019, Vol. 1, pp. 18–27. DOI: 10.20998/2079-0775.2019.1.03
2. Kravari K., Bassiliades N. A Survey of Agent Platforms, *Journal of Artificial Societies and Social Simulation,* 2015, Vol. 18, № 1, pp. 1–18. DOI: 10.18564/jasss.2661
3. Schurr N., Schurr N., Marecki J. et al. The Future of Disaster Response: Humans Working with Multiagent Teams using DEFACTO, *Conference: AI Technologies for Homeland Security : Papers from the 2005 AAAI Spring Symposium, Technical Report SS-05-01.* Stanford, California, USA, March 21–23, 2005 : proceedings, Menlo Park, California : The AAAI Press, 2005, pp. 9–16.
4. Ayanian N. DART: Diversity-enhanced Autonomy in Robot Teams, *The International Journal of Robotics Research.* – 2018, pp. 1–8. DOI: 10.1177/0278364919839137.
5. Minieka E. Optimization algorithms on networks and graphs. Moscow, Mir, 1981, 323 p.
6. Held M., Karp R. The Travelling-Salesman Problem and Minimum Spanning Trees: Part II, *Math. Programming,* 1971, Vol. 1, pp. 6–25. DOI:10.1007/BF01584070
7. Bellman R. On a Routing Problem, *Quarterly of Applied Mathematics*, 1958, Vol. 16 (1), pp. 87–90. DOI: 10.1090/qam/102435
8. Hadjiconstantinou E., Christofides N., Mingozzi A. A new exact algorithm for the vehicle routing problem based on q-paths and k-shortest paths relaxations, *Annals of Operations Research,* 1995, Vol. 61, pp. 21–43. DOI:10.1007/BF02098280
9. Christofides N., Mingozzi A., Toth P. Exact algorithm for the Vehicle Routing Problem Based on Spanning Tree and Shortest Paths Relaxations, *Mathematical Programming*, 1981, Vol. 20, No. 1, pp. 255–282. DOI:10.1007/BF01589353
10. Christofides N. Theory of graphs. Algorithmic approach. Moscow, Mir, 1978, 432 p.
11. Karger D. Motwani R., Ramkumar G.D.S. On approximating the longest path in a graph, *Algorithmica,* 1997, Vol. 18, pp. 82–98. DOI: 10.1007/BF02523689
12. Feder T., Motwani R. Finding large cycles in Hamiltonian graphs, *16th annual ACM-SIAM Symp. on Discrete Algorithms (SODA), Vancouver, 23–25 January 2005 : proceedings*. Philadelphia, United States, Society for Industrial and Applied Mathematics, 2005, pp. 166–175. DOI: 10.5555/1070432
13. Gabow H. N. Finding paths and cycles of super polylogarithmic length, *36th annual ACM Symp. on Theory of Computing (STOC), Chicago, 13–16 June 2004 : proceedings*. New York, United States : Association for Computing Machinery, 2004, pp. 407–416. DOI: 10.1145/1007352.1007418
14. Gabow H. N., Nie S. Finding long paths, cycles, and circuits, *19th annual International Symp. on Algorithms and Computation (ISAAC), Gold Coast, Australia, December 2008 : proceedings*. Berlin, Springer-Verlag, 2008, pp. 752–763. DOI: 10.1007/978-3-540-92182-0_66
15. Vishwanathan S. An approximation algorithm for finding a long path in Hamiltonian graphs, *11th annual ACM-SIAM Symp. on Discrete Algorithms (SODA), San Francisco, 9–11 January 2000 : proceedings.* Philadelphia, United States : Society for Industrial and Applied Mathematics, 2000, pp. 680–685. DOI: 10.5555/338219
16. Zhang Z., Li H. Algorithms for long paths in graphs, *Theoretical Computer Science,* 2007. Vol. 377, Issue 1–3. pp. 25–34. DOI: 10.1016/j.tcs.2007.02.012
17. Garey M. R., Johnson D. S. Computers and Intractability: A Guide to the Theory of NP-completeness. New York, W.H. Freeman, 1979, 340 p.
18. Garey M. R., Johnson D. S., Tarjan R. E. The planar Hamiltonian circuit problem is NP-complete, *SIAM*

DOI 10.15588/1607-3274-2023-1-13

J. Computing, 1976. Vol. 5, pp. 704–714. DOI: 10.1137/0205049
19. Golumbic M. C. Algorithmic Graph Theory and Perfect Graphs, *Vol. 57 : Annals of Discrete Mathematics. 2nd Edition*. Amsterdam, North-Holland Publishing Co., 2004, 592 p. ISBN: 9780080526966
20. Müller H. Hamiltonian circuits in chordal bipartite graphs, *Discrete Math.*, 1996, Vol. 156, pp. 291–298. DOI: 10.1016/0012-365X(95)00057-4
21. Narasimhan G. A note on the Hamiltonian circuit problem on directed path graphs, *Information Processing Letters*, 1989, Vol. 32, pp. 167–170. DOI: 10.1016/0020-0190(89)90038-0
22. Damaschke P. The Hamiltonian circuit problem for circle graphs is NP-complete, *Information Processing Letters.* – 1989, Vol. 32, pp. 1–2. DOI: 10.1016/0020-0190(89)90059-8
23. Itai A., Papadimitriou C. H., Szwarcfiter J. L. Hamiltonian paths in grid graphs, *SIAM J. Computing*, 1982, Vol. 11, pp. 676–686. DOI: 10.1137/0211056
24. Arikati S. R., Pandu Rangan C. Linear algorithm for optimal path cover problem on interval graphs, *Information Processing Letters,* 1990, Vol. 35, pp. 149–153. DOI: 10.1016/0020-0190(90)90064-5
25. Bertossi A. A. Finding Hamiltonian circuits in proper interval graphs, *Information Processing Letters*, 1983, Vol. 17, pp. 97–101. DOI: 10.1016/0020-0190(83)90078-9
26. Chang M. S., Peng S. L., Liaw J. L. Deferred-query: An efficient approach for some problems on interval graphs, *Networks,* 1999, Vol. 34, Issue 1, pp. 1–10. DOI: 10.1002/(sici)1097-0037(199908)34:1<1::aid-net1>3.0.co
27. Damaschke P. Paths in interval graphs and circular arc graphs, *Discrete Math.*, 1993, Vol. 112, pp. 49–64. DOI: 10.1016/0012-365X(93)90223-G
28. Asdre K., Nikolopoulos S. D. The 1-fixed-endpoint path cover problem is polynomial on interval graphs, *Algorithmica,* 2009, Vol. 58, pp. 679–710. DOI: 10.1007/s00453-009-9292-5
29. Damaschke P., Deogun J. S., Kratsch D. et al. Finding Hamiltonian paths in cocomparability graphs using the bump number algorithm, *Order,* 1992, Vol. 8, pp. 383–391. DOI: 10.1007/bf00571188
30. Bulterman R., Sommen F. van der, Zwaan G. et al. On computing a longest path in a tee, *Information Processing Letters*, 2002, Vol. 81, pp. 93–96. DOI: 10.1016/S0020-0190(01)00198-3
31. Uehara R., Uno Y. Efficient algorithms for the longest path problem, *15th annual International Symp. on Algorithms and Computation (ISAAC), Hong Kong*, December 2004 : *proceedings.* Berlin, Springer-Verlag, 2004, pp. 871–883. DOI:10.1007/978-3-540-30551-4_74
32. Uehara R., Valiente G. Linear structure of bipartite permutation graphs and the longest path problem, *Information Processing Letters,* 2007, Vol. 103, pp. 71–77. DOI: 10.1016/j.ipl.2007.02.010
33. Takahara Y., Teramoto S., Uehara R. Longest path problems on ptolemaic graphs, *IEICE Trans. Inf. and Syst.*, 2008, Vol. 91-D, pp. 170–177. DOI: 10.1093/ietisy/e91-d.2.170
34. Ioannidou K., Mertzios G., Nikolopoulos S. The Longest Path Problem has a Polynomial Solution on Interval Graphs, *Algorithmica*, 2011, Vol. 61, pp. 320–341. DOI: 10.1007/s00453-010-9411-3
35. Mertzios G. B., Corneil D. G. A Simple Polynomial Algorithm for the Longest Path Problem on Cocomparability Graphs, *SIAM Journal on Discrete Mathematics*, 2012, Vol. 26, Issue 3, pp. 940–963. DOI: 10.1137/100793529
36. Floyd R. Algorithm 97: Shortest Path, *Communications of the ACM*, 1961, Vol. 5 (6), pp. 344–348. DOI: 10.1145/367766.368168
37. Hougardy S. The Floyd-Warshall algorithm on graphs with negative cycles, *Information Processing Letters*, 2010, Vol. 110 (8–9), pp. 279–281. DOI: 10.1016/j.ipe.2010.02.001
38. Warshall S. Algorithm on Boolean matrices, *Journal of the ACM,* 1962, Vol. 9 (1), pp. 11–12. DOI: 10.1145/321105.321107
39. Shimbel A. Structural parameters of communication networks, *Bulletin of Mathematical Biophysics*, 1953, Vol. 15 (4), pp. 501–507. DOI: 10.1007/BF02476438
40. Dijkstra E. W. A note on two problems in connexion with graphs, *Numerische Mathematik,* 1959, Vol. 1, Issue 1, pp. 269–271. DOI:10.1007/BF01386390
41. Lipskyy V. Combinatory for programmers, Moscow, Mir Press., 1988, 213 p.
42. Batsamut V., Manzura S., Kosiak O. et al. Fast Algorithm for Calculating Transitive Closures of Binary Relations in the Structure of a Network Object, *International Journal of Computing*, 2021, Vol. 20(4), pp. 560–566. DOI: 10.47839/ijc.20.4.2444

Received 22.11.2022.
Accepted 03.02.2023.

УДК 519.853: 658.52

МЕТОД МАРШРУТИЗАЦІЇ ГРУПИ МОБІЛЬНИХ РОБОТІВ НА СТАЦІОНАРНІЙ МЕРЕЖІ ДЛЯ ВИКОНАННЯ ЗАВДАНЬ ПОШУКУ ЗНИКЛИХ ОБ'ЄКТІВ В ЗОНІ ТЕХНОГЕННОЇ АВАРІЇ

Бацамут В. М. – д-р військ. наук, професор, заступник начальника науково-дослідного центру службово-бойової діяльності Національної гвардії України Національної академії Національної гвардії України, Харків, Україна.

Годлевський С. О. – науковий співробітник науково-дослідного центру службово-бойової діяльності Національної гвардії України Національної академії Національної гвардії України, Харків, Україна.

АНОТАЦІЯ

Актуальність. Актуальність статті обумовлюється потребою у подальшому розвитку моделей колективної поведінки систем із мультиагентною побудовою структури, у наділенні таких систем інтелектом, який забезпечує синхронізацію спільних зусиль різних агентів у ході досягнення поставлених перед системою цілей. Запропонований у статті метод усуває проблему конкуренції між різними агентами мультиагентної системи, що є важливим у ході виконання пошукових, рятувальних, моніторингових завдань у кризових районах різного характеру походження.

DOI 10.15588/1607-3274-2023-1-13

Мета роботи полягає у розробленні методу визначення достатньої чисельності мультиагентної системи та оптимальних маршрутів руху її окремих елементів на стаціонарній мережі для максимально повного обстеження зони техногенної аварії (будь-якої заданої зони, в основі якої лежить певна транспортна мережа).

Метод. Застосовано ідею динамічного програмування для пошуку в структурі модельного зваженого орієнтованого графа всіх можливих реберно-простих найдовших шляхів, що з'єднують директивно визначені підмножини вершин-істоків та вершин-стоків. З цією метою застосовано модифікований метод Дейкстри. Модифікація полягає у представленні ваг дуг моделюючого орієнтованого графа значеннями з від'ємної області з подальшою роботою метода Дейкстри з цими значеннями. Після відшукування чергового реберно-простого найдовшого шляху, дуги, що його складають, фіксуються у пам'яті обчислювальної системи (у маршрутному плані) та видаляються зі структури графа і процес ітераційно повторюється. Пошук шляхів відбувається доти, поки зберігається транзитивне замкнення між вершинами, що входять до складу визначених підмножин вершин-істоків та вершин-стоків. Розроблений метод дозволяє знайти таку сукупність маршрутів руху для елементів мультиагентної системи, яка максимізує обстежену ними площу в зоні техногенної аварії (або кількість перевірених об'єктів на маршрутах руху) за одну "хвилю" пошуку, та розподіляє елементи мультиагентної системи маршрутами, що не мають спільних ділянок. Похідною застосування розробленого методу є визначення достатньої чисельності мультиагентної системи для ефективного проведення пошукових заходів у межах визначеної зони.

Результати. 1) Розроблено метод маршрутизації групи мобільних роботів на стаціонарній мережі для виконання завдань пошуку зниклих об'єктів в зоні техногенної аварії; 2) Формалізовано робочий вираз методу Дейкстри для пошуку в структурі мережевого об'єкту (в структурі модельного графа) шляхів найбільшої довжини; 3) Запропонована сукупність показників для комплексного оцінювання маршрутних планів мультиагентної системи; 4) Виконано верифікацію методу на тестових задачах.

Висновки. Проведені теоретичні дослідження та низка експериментів підтверджують працездатність розробленого методу. Рішення, що виробляються із використанням розробленого методу, є точними, що дозволяє рекомендувати його до практичного використання при визначенні в автоматизованому режимі маршрутних планів для мультиагентних систем, а також потрібної кількості агентів в таких системах для виконання необхідного обсягу пошукових завдань у певному кризовому районі.

КЛЮЧОВІ СЛОВА: мультиагентна система, група мобільних роботів, маршрутизація, мережевий об'єкт, зважений неорієнтований (орієнтований) граф, екстремальні шляхи, критерій оптимізації, метод.

ЛІТЕРАТУРА

1. Коноваленко О. Мультиагентні системи управління та підтримки прийняття рішень / О. Коноваленко, В. Брусенцев // Bulletin of the National Technical University "KhPI". Ser.: Mechanical engineering and CAD. – 2019. – № 1. – С. 18–27. DOI: 10.20998/2079-0775.2019.1.03
2. Kravari K. A Survey of Agent Platforms / K. Kravari, N. Bassiliades // Journal of Artificial Societies and Social Simulation. – 2015. – Vol. 18, № 1. – P. 1–18. DOI: 10.18564/jasss.2661
3. The Future of Disaster Response: Humans Working with Multiagent Teams using DEFACTO / [N. Schurr, N. Schurr, J. Marecki et al.] // Conference: AI Technologies for Homeland Security : Papers from the 2005 AAAI Spring Symposium, Technical Report SS-05-01, Stanford, California, USA, March 21–23, 2005 : proceedings. – Menlo Park, California : The AAAI Press, 2005. P. 9–16.
4. Ayanian N. DART: Diversity-enhanced Autonomy in Robot Teams / N. Ayanian // The International Journal of Robotics Research. – 2018. – P. 1–8. DOI: 10.1177/0278364919839137
5. Майника Э. Алгоритмы оптимизации на сетях и графах / Э. Майника. – Москва : Мир, 1981. – 323 с.
6. Held M. The Travelling-Salesman Problem and Minimum Spanning Trees: Part II / M. Held, R. Karp // Math. Programming. – 1971. – Vol. 1 – P. 6–25. DOI:10.1007/BF01584070
7. Bellman R. On a Routing Problem / R. Bellman // Quarterly of Applied Mathematics. – 1958. – Vol. 16 (1). – P. 87–90. DOI: 10.1090/qam/102435
8. Hadjiconstantinou E. A new exact algorithm for the vehicle routing problem based on q-paths and k-shortest paths relaxations / E. Hadjiconstantinou, N. Christofides, A. Mingozzi // Annals of Operations Research. – 1995. – Vol. 61. – P. 21–43. DOI:10.1007/BF02098280
9. Christofides N. Exact algorithm for the Vehicle Routing Problem Based on Spanning Tree and Shortest Paths Relaxations / N. Christofides, A. Mingozzi, P. Toth // Mathematical Programming. – 1981. – Vol. 20, No. 1. – P. 255–282. DOI:10.1007/BF01589353
10. Кристофидес Н. Теория графов. Алгоритмический подход / Н. Кристофидес. – Москва : Мир, 1978. – 432 с.
11. Karger D. On approximating the longest path in a graph / D. Karger, R. Motwani, G.D.S. Ramkumar // Algorithmica. – 1997. – Vol. 18. – P. 82–98. DOI: 10.1007/BF02523689
12. Feder T. Finding large cycles in Hamiltonian graphs / T. Feder R. Motwani // 16th annual ACM-SIAM Symp. on Discrete Algorithms (SODA), Vancouver, 23–25 January 2005 : proceedings. – Philadelphia, United States : Society for Industrial and Applied Mathematics, 2005. – P. 166–175. DOI: 10.5555/1070432
13. Gabow H. N. Finding paths and cycles of super polylogarithmic length / H. N. Gabow // 36th annual ACM Symp. on Theory of Computing (STOC), Chicago, 13–16 June 2004 : proceedings. – New York, United States : Association for Computing Machinery, 2004, P. 407–416. DOI: 10.1145/1007352.1007418
14. Gabow H.N. Finding long paths, cycles, and circuits / H. N. Gabow, S. Nie // 19th annual International Symp. on Algorithms and Computation (ISAAC), Gold Coast, Australia, December 2008 : proceedings. – Berlin : Springer-Verlag, 2008. – P. 752–763. DOI: 10.1007/978-3-540-92182-0_66
15. Vishwanathan S. An approximation algorithm for finding a long path in Hamiltonian graphs / S. Vishwanathan // 11th annual ACM-SIAM Symp. on Discrete Algorithms (SODA), San Francisco, 9–11 January 2000 : proceedings. – Philadelphia, United States : Society for Industrial and Applied Mathematics, 2000. – P. 680–685. DOI: 10.5555/338219
16. Zhang Z. Algorithms for long paths in graphs / Z. Zhang, H. Li // Theoretical Computer Science. – 2007. – Vol. 377, Issue 1–3. – P. 25–34. DOI: 10.1016/j.tcs.2007.02.012

DOI 10.15588/1607-3274-2023-1-13

17. Garey M. R. Computers and Intractability: A Guide to the Theory of NP-completeness / M. R. Garey, D. S. Johnson. – New York : W.H. Freeman, 1979. – 340 p.
18. Garey M. R. The planar Hamiltonian circuit problem is NP-complete / M. R. Garey, D. S. Johnson, R. E. Tarjan // SIAM J. Computing. – 1976. – Vol. 5. – P. 704–714. DOI: 10.1137/0205049
19. Golumbic M. C. Algorithmic Graph Theory and Perfect Graphs / M. C. Golumbic. – Vol. 57 : Annals of Discrete Mathematics. 2nd Edition. – Amsterdam : North-Holland Publishing Co., 2004. – 592 p. ISBN: 9780080526966
20. Müller H. Hamiltonian circuits in chordal bipartite graphs / H. Müller // Discrete Math. – 1996. – Vol. 156. – P. 291–298. DOI: 10.1016/0012-365X(95)00057-4
21. Narasimhan G. A note on the Hamiltonian circuit problem on directed path graphs / G. Narasimhan // Information Processing Letters. – 1989. – Vol. 32. – P. 167–170. DOI: 10.1016/0020-0190(89)90038-0
22. Damaschke P. The Hamiltonian circuit problem for circle graphs is NP-complete / P. Damaschke // Information Processing Letters. – 1989. – Vol. 32. – P. 1–2. DOI: 10.1016/0020-0190(89)90059-8
23. Itai A. Hamiltonian paths in grid graphs / A. Itai, C. H. Papadimitriou, J. L. Szwarcfiter // SIAM J. Computing. – 1982. – Vol. 11. – P. 676–686. DOI: 10.1137/0211056
24. Arikati S. R. Linear algorithm for optimal path cover problem on interval graphs / S. R. Arikati, Rangan C. Pandu. // Information Processing Letters. – 1990. – Vol. 35. – P. 149–153. DOI: 10.1016/0020-0190(90)90064-5
25. Bertossi A. A. Finding Hamiltonian circuits in proper interval graphs / A. A. Bertossi // Information Processing Letters. – 1983. – Vol. 17. – P. 97–101. DOI: 10.1016/0020-0190(83)90078-9
26. Chang M. S. Deferred-query: An efficient approach for some problems on interval graphs / M. S. Chang, S. L. Peng, J. L. Liaw // Networks. – 1999. – Vol. 34, Issue 1. – P. 1–10. DOI: 10.1002/(sici)1097-0037(199908)34:1<1::aid-net1>3.0.co
27. Damaschke P. Paths in interval graphs and circular arc graphs / P. Damaschke // Discrete Math. – 1993. – Vol. 112. – P. 49–64. DOI: 10.1016/0012-365X(93)90223-G
28. Asdre K. The 1-fixed-endpoint path cover problem is polynomial on interval graphs / K. Asdre, S. D. Nikolopoulos // Algorithmica. – 2009. – Vol. 58. – P. 679–710. DOI: 10.1007/s00453-009-9292-5
29. Finding Hamiltonian paths in cocomparability graphs using the bump number algorithm / [P. Damaschke, J. S. Deogun, D. Kratsch et al.] // Order. – 1992. – Vol. 8. – P. 383–391. DOI: 10.1007/bf00571188
30. On computing a longest path in a tee / [R. Bulterman, F. van der Sommen, G. Zwaan et al.] // Information Processing Letters. – 2002. – Vol. 81. – P. 93–96. DOI: 10.1016/S0020-0190(01)00198-3
31. Uehara R. Efficient algorithms for the longest path problem / R. Uehara, Y. Uno // 15th annual International Symp. on Algorithms and Computation (ISAAC), Hong Kong, December 2004 : proceedings. – Berlin : Springer-Verlag, 2004 – P. 871–883. DOI:10.1007/978-3-540-30551-4_74
32. Uehara R. Linear structure of bipartite permutation graphs and the longest path problem / R. Uehara, G. Valiente // Information Processing Letters. – 2007. – Vol. 103. – P. 71–77. DOI: 10.1016/j.ipl.2007.02.010
33. Takahara Y. Longest path problems on ptolemaic graphs / Y. Takahara, S. Teramoto, R. Uehara // IEICE Trans. Inf. and Syst. – 2008. – Vol. 91-D. – P. 170–177. DOI: 10.1093/ietisy/e91-d.2.170
34. Ioannidou K. The Longest Path Problem has a Polynomial Solution on Interval Graphs / K. Ioannidou, G. Mertzios, S. Nikolopoulos // Algorithmica. – 2011. – Vol. 61. – P. 320–341. DOI: 10.1007/s00453-010-9411-3
35. Mertzios G. B. A Simple Polynomial Algorithm for the Longest Path Problem on Cocomparability Graphs / G. B. Mertzios , D. G. Corneil // SIAM Journal on Discrete Mathematics. – 2012. – Vol. 26, Issue 3, – P. 940–963. DOI: 10.1137/100793529
36. Floyd R. Algorithm 97: Shortest Path / R. Floyd // Communications of the ACM. – 1961 – Vol. 5 (6). – P. 344–348. DOI: 10.1145/367766.368168
37. Hougardy S. The Floyd-Warshall algorithm on graphs with negative cycles / S. Hougardy // Information Processing Letters. – 2010. – Vol. 110 (8–9). – P. 279–281. DOI: 10.1016/j.ipe.2010.02.001
38. Warshall S. Algorithm on Boolean matrices / S. Warshall // Journal of the ACM. – 1962. – Vol. 9 (1). – P. 11–12. DOI: 10.1145/321105.321107
39. Shimbel A. Structural parameters of communication networks / A. Shimbel // Bulletin of Mathematical Biophysics. – 1953. – Vol. 15 (4). – P. 501–507. DOI: 10.1007/BF02476438
40. Dijkstra E. W. A note on two problems in connexion with graphs / E. W. Dijkstra // Numerische Mathematik. – 1959. – Vol. 1, Issue 1. – P. 269–271. DOI:10.1007/BF01386390.
41. Lipskyy V. Combinatory for programmers / V. Lipskyy. Moscow : Mir Press., 1988. – 213 p.
42. Fast Algorithm for Calculating Transitive Closures of Binary Relations in the Structure of a Network Object / [V. Batsamut, S. Manzura, O. Kosiak et al.] // International Journal of Computing. – 2021. – Vol. 20(4). – P. 560–566. DOI: 10.47839/ijc.20.4.2444

DOI 10.15588/1607-3274-2023-1-13