UDC 519.1:519.7:007.5
DOI https://doi.org/10.32689/maup.it.2025.1.32

Andrii STOPKIN
Candidate of Physical and Mathematical Sciences, Associate Professor,
Associate Professor at the Department of Mathematics and Informatics,
State Higher Educational Institution «Donbas State Pedagogical University», stepkin.andrej@gmail.com
***ORCID:** 0000-0002-6130-9920*

THE USE OF A COLLECTIVE OF MOBILE AGENTS FOR THE EXPLORATION OF UNDIRECTED GRAPHS

Abstract. *The study proposed in this work is devoted to the problem of graph exploration by a collective of agents consisting of one stationary agent, which performs the necessary computations, and two mobile agents, which move through the graph and collect information about its structure.*

The purpose *of this work is to develop a new efficient algorithm for the exploration of finite undirected graphs without loops and multiple edges by a collective of agents. The paper proposes the following* ***methodology*** *to achieve this goal: utilizing a collective of three agents. Two of them are agent-researchers, which can move through the graph, read labels on graph elements, and place or remove these labels. The agent-researchers have finite memory that does not depend on the number of nodes in the explored graph and use two colors each (three distinct colors in total) for graph exploration. During the process, the agent-researchers send messages to the third agent, the agent-experimenter, which is a stationary agent that stores the results of the agent-researchers' operations. Based on this information, the agent-experimenter gradually constructs a representation of the explored graph in its memory in the form of edge and node lists. The paper describes the working modes of the agent-researchers in detail. It also thoroughly examines the algorithm for processing the messages received by the agent-experimenter from the agent-researchers, which serves as the basis for constructing a map of the explored graph. The study analyzes the time, space, and communication complexities of the developed algorithm and evaluates the number of edge transitions required for agents to explore the graph.*

The scientific novelty *lies in the development of an efficient algorithm for graph exploration by a collective of agents.*

Conclusions. *The paper proposes a new graph exploration algorithm that has quadratic time, space, and communication complexities, with the number of edge transitions performed by the agent-researchers being estimated as* $O\left(n^2\right)$*, where* n *is the number of nodes in the explored graph. The functioning of the proposed exploration algorithm is based on the depth-first traversal method.*

Key words: *graph exploration, undirected graphs, algorithm complexity, depth-first traversal method, collective of agents.*

Андрій СТЬОПКІН. ВИКОРИСТАННЯ КОЛЕКТИВУ МОБІЛЬНИХ АГЕНТІВ ДЛЯ РОЗПІЗНАВАННЯ НЕОРІЄНТОВАНИХ ГРАФІВ

Анотація. *Дослідження, запропоноване в роботі присвячено вивченню проблеми розпізнавання графів колективом агентів, що складається з одного нерухомого агента, що виконує необхідні обчислення та двох мобільних агентів, що рухаються графом та збирають інформацію про його структуру.*

Метою роботи *є побудова нового ефективного алгоритму розпізнавання скінчених неорієнтованих графів без петель та кратних ребер колективом агентів. В статті запропоновано наступну* ***методологію*** *до досягнення поставленої мети: використати колектив агентів, що складається з трьох агентів. Два агенти є агентами-дослідниками, які можуть переміщуватися по графу, зчитувати мітки на елементах графу й залишати або видаляти ці мітки. Агенти-дослідники мають скінчену пам'ять, яка не залежить від кількості вершин досліджуваного графа та для розпізнавання графу використовують по дві фарби кожен (всього три фарби різного кольору). Під час роботи агенти-дослідники відправляють повідомлення третьому агенту – агенту-експериментатору, який представляє собою нерухомого агента, в пам'яті якого фіксується результат функціонування агентів-дослідників. На основі цієї інформації він поступово вибудовує в своїй пам'яті представлення досліджуваного графа списками ребер і вершин. У статті наведено режими роботи агентів-дослідників з їх детальним описом. Також детально розглянуто алгоритм обробки агентом-експериментатором повідомлень, отриманих від агентів-дослідників в процесі роботи, на підставі яких відбувається побудова мапи досліджуваного графа. В роботі проаналізовано часову, ємнісну, комунікаційну складності побудованого алгоритму та проведено аналіз кількості переходів по ребрах, які необхідно здійснити агентам для розпізнавання графа.*

Науковою новизною *є отримання ефективного алгоритму розпізнавання графів колективом агентів.*

Висновки. *В роботі запропоновано новий алгоритм розпізнавання графів, який має квадратичні часову, ємнісну та комунікаційну складності та кількість переходів по ребрах, які виконують агенти-дослідники, що оцінюється як* $O\left(n^2\right)$ *де* n *– кількість вершин досліджуваного графа. Робота запропонованого алгоритму розпізнавання ґрунтується на методі обходу графа в глибину.*

Ключові слова: *розпізнавання графів, неорієнтований граф, складність алгоритму, обхід в глибину, колектив агентів.*

Problem statement in general and its connection with the important scientific or practical tasks. Rapidly developing and increasingly relevant, robotics has become a major field within cybernetics. At the origins of robotics lies the desire to assist humans in performing heavy and even hazardous tasks or, in some cases, to replace humans entirely in such work. Given that many environments remain unexplored

[5] due to their inaccessibility or danger to humans, it becomes evident that the problem of environment exploration by robots is highly relevant [7, 14, 15]. Moreover, using multiple robots instead of a single one [6, 8, 9, 11] increases the likelihood of successfully completing the task, especially considering the potential failure of agents during operation. Additionally, it may even improve the time complexity of the exploration algorithm.

Many works have been dedicated to the exploration of graphs by a single agent, and numerous results have been obtained regarding the feasibility and complexity of such exploration. However, the exploration of a graph by a team of agents wandering through it remains underexplored. This makes the tasks of systematically studying experiments on graph exploration by multiple wandering agents highly relevant. Specifically, these tasks include creating movement routes for agents through an unknown graph, marking its elements, collecting and processing local information about the graph, and constructing the graph based on this information, down to the markings on the graph's elements. Additionally, tasks focused on finding methods to optimize resource usage, time costs, communication channel load, etc., during graph exploration remain important.

In the existing works, little attention has been paid to the study of information exchange between agents, as well as its complexity and potential impact on time and space complexity.

The goal of this work is to create an efficient method and develop a corresponding algorithm for exploring unknown graphs using a collective of agents [12, 13], as well as to investigate the time, space, and communication complexities of the constructed algorithm.

The analysis of recent research and publications, which have initiated the solution of the relevant problems. The beginning of research in this scientific direction is commonly attributed to the work of C. Shannon [10], in which the problem of an automaton-mouse searching for a given target in a maze was considered.

Active studies of automata behavior in mazes began in 1971 after the publication of C. Dopp's work [2], which described the traversal of chessboard mazes by finite automata. Over time, the problem conditions expanded, requiring the automaton to visit all nodes and/or edges of the explored graph (maze).

Later, in [3], research was conducted on analyzing the properties of an unknown environment under various methods of interaction between the automaton and its operating environment, as well as under different levels of prior information about it.

Analysis of a graph includes a number of specific tasks. Let's consider the main ones: the problem of self-localization (determining the node of the graph where the agent is initially located), the problem of map control (checking the isomorphism between the explored graph and the reference graph (map)), and the problem of complete graph exploration (constructing the reference graph (map) of the explored graph). Several approaches have been defined for solving the latter problem, and a number of algorithms for agent wandering through the graph and methods for marking graph elements with colors or stones have been proposed, allowing the exploration of the graph with precision up to isomorphism. Thus, in the work [4], a method is proposed in which the agent, during its movement, marks only the incidentors (the point of connection of edges with nodes) of the edges it has passed. To explore the edges, the agent moves along them and then returns to the initial node via the shortest path. At the same time, it remembers the marks of the passed incidentors, which unambiguously determine the node the agent reached when moving along the explored edge.

In the work [1], agents exchange information about their completed tasks with the base station via a special network. This network has some limitations on data exchange. In other words, to coordinate the actions between agents, it is necessary to first form a network that meets these conditions. Data exchange with the base station occurs in pre-designated locations. The article's feature is asynchronous strategies that work with arbitrary communication models. Asynchrony refers to the possibility of issuing instructions to subgroups of agents at the moment when they are ready to accept them.

In the article [6], an attempt was made to develop general concepts and requirements for graph exploration by multi-agent systems. The proposed representation for the multi-agent system is aimed at providing general conditions for declaring the completion of graph exploration and completing such exploration within a finite time. The work proposes modifications to the incidence matrix for organized information exchange between agents. It also presents a general algorithm for the collective work of agents exploring the graph. The algorithm is based on the proposed generalized structure and the modified incidence matrix.

In the article [9], the author considers the problem of preserving the direction of movement by a collective of finite automata on a two-dimensional integer lattice of width two, where the elements (vertices) are anonymous. The automata do not distinguish between equally labeled vertices by their direction coordinates (i.e., each automaton lacks a compass). The study considers collectives consisting of a single automaton and several pebbles, where the layout of the pebbles is entirely determined by the automaton. It is proven that a collective of an automaton and at most three pebbles cannot maintain the direction of movement on the lattice, but a collective of an automaton and four pebbles can.

The presentation of the main material. The paper considers simple, connected undirected finite graphs $G=(V,E)$ where V is the set of nodes, E is the set of edges (two-element subsets (u,v), where $u,v \in V$). The three $((u,v),v)$ we call the incidentor (a contact point) of the edge (u,v) and the node v. Through I we denote a set of all the incidentors of the graph. A set $L=V \cup E \cup I$ is called a set of graph G elements. A surjective map $\mu: L \to \{w,r,y,ry,b\}$, where w is interpreted as white, r is red, y is yellow, ry is red-yellow and b is black, we call the coloring function of the graph G. A pair (G,μ) is called a colored graph. A sequence $u_1,u_2,...u_k$ of pairwise adjacent nodes of the graph G is called a length path of k. The vicinity $Q(v)$ of the node v we call the set of elements of the graph, consisting of a node v, all nodes u adjacent to v, all edges (v,u) and all the incidentors $((v,u),v),((v,u),u)$. The cardinality of the sets of nodes V and edges E we denote through n and m, respectively. It is clear that $m \le \frac{n(n-1)}{2}$. We call the isomorphism of the graph G and the graph H such a bijection $\varphi: V_G \to V_H$ that $(v,u) \in E_G$ exactly when $(\varphi(v),\varphi(u)) \in E_H$. Thus, isomorphic graphs are equal up to the designation of nodes and the coloring of their elements.

A collective of agents consists of two types of agents:

Agent-Researcher (AR) – an agent that can move along a finite graph, during the process of its work, it can color the nodes, edges, and incidentors of the graph, perceive these marks (based on these marks, the AR moves along the graph), and also send messages to the Agent-Experimenter (AE);

Agent-Experimenter (AE) – a stationary agent that, during its work, can send, receive, and identify messages from AR, possesses a finite, unbounded growing internal memory, where the results of AR's functioning at each step are recorded, and in addition, a representation of the graph (initially unknown to the agents) is gradually built from lists of edges and nodes.

The paper proposes an algorithm that uses two ARs (let's call them agents A and B) and one AE. For the work of each AR, two colors are required: for agent A it is r, for agent B it is b, so in total, the algorithm uses three colors: r, y, b.

At the beginning of the work, AR A and AR B are placed in arbitrary non-coincident nodes of the graph G. The agents go «in depth» as far as possible, return, search for another path with unvisited nodes and untraversed edges. If an adjacent node is discovered, colored in a «foreign» color, the agent marks all the isthmuses (edges connecting subgraphs explored by different ARs) from the current node to the foreign area and informs the second AR, through the AE, about the need to explore the marked isthmuses. While the second AR is exploring the isthmuses, the first AR cannot mark new isthmuses. If there are no other possible movement options except for marking a new isthmus, the first AR stops until the second AR has explored all the marked isthmuses. When moving to white nodes in the graph, ARs A and B color its elements in red and yellow, respectively. When moving «in depth», the red (yellow) path is extended, and when moving back, it shortens. When moving back, the length of the path does not change for exploring back edges or isthmuses. A node, from which only backward movement along its path is possible, or where no movement options exist, is colored black. Thus, the red and yellow paths of each agent are formed. The algorithm ends when the red and yellow paths become empty, and all nodes are black. During the graph traversal, the agents create an implicit numbering of visited nodes. Upon first visiting a node, the agent A colors it red (agent B – yellow), and the AE assigns it a number equal to the value of the variable Ct_A (Ct_B for agent B). Note that the variables and are stored and processed by the AE and can only take odd and even values, respectively. The graph reconstruction is carried out based on the numbering created by the agents by constructing a graph isomorphic to G.

Now let's consider the modes in which the ARs can operate. The messages sent by the ARs in the corresponding situation are indicated in parentheses, specifying the name of the agent sending the message («MESSAGE_A»; «MESSAGE_B»).

1. *Normal operating mode.* In this mode, the AR moves along nodes colored white, painting these nodes, their connecting edges, and distant incidentors in its designated color (red for agent A, yellow for agent B) («FORWARD_A»; «FORWARD_B»). If there are no white nodes or other possible movement paths, the AR retraces its path, coloring the visited nodes, their connecting edges, and nearby incidentors black («BACK_A»; «BACK_B»). The AR stops working when its starting node is colored black due to the absence of possible movement paths («STOP_A», «STOP_B»).

2. *Back edge exploration mode*. If, while moving forward, the AR detects a white edge in node v whose distant node is colored in its «own» color (a back edge), it switches to back edge exploration mode and colors the near incidentors of all back edges incident to node v in its «own» color («MARK_BE_A»; «MARK_BE_B»). After completing the coloring of the incidentors, the AR moves back along its path, sending a message at each step («RETREAT_A»; «RETREAT_B»), until it reaches the node incident to the previously marked back edge. It then traverses this edge, coloring it black («FORWARD_BE_A»; «FORWARD_BE_B»). If there are still unexplored back edges, the AR moves back along the previously traversed edge, coloring the near incidentor black, and

continues searching for back edges. After detecting the last back edge, the AR colors its near incidentor black («BE_EXPLORED_A»; «BE_EXPLORED_B») and completes its work in back edge exploration mode.

3. *Isthmus Marking Mode.* If, during the graph exploration, an isthmus is detected in node v, the AR switches to isthmus marking mode, provided that all previously marked isthmuses by this AR have been explored by the second AR. If the second AR has not yet explored all the marked isthmuses, the first AR cannot mark new ones. If the first AR has no other possible moves except marking a new isthmus, it stops until the second AR explores all previously marked isthmuses. In this mode, the AR colors the near incidentors of all isthmuses incident to node v in black («MARK_AB»; «MARK_BA»). When all isthmuses are marked, the AR completes its work in this mode («FIX_A»; «FIX_B»). Upon completion of the isthmus marking mode, the agent stores information about the number of marked isthmuses. In this mode, agent A has priority over agent B. Therefore, if both ARs detect the same isthmus simultaneously, it will be marked by agent A.

4. *Mode of Isthmus Exploration.* Upon receiving a command from the AE to explore isthmuses, the AR switches to the isthmus exploration mode. If, at that moment, the AR is operating in the back edge exploration mode, it will not switch to the isthmus exploration mode until the current mode is completed. Upon switching, the AR checks whether there are any possible movement paths from its current node other than moving backward along its path. If such paths exist, the AR moves back along its path without coloring anything («RETREAT_A»; «RETREAT_B») until it detects the nearest node incident to a white edge whose distant incidentor is colored black and whose distant node is colored in the «foreign» color (isthmus edge). If no such paths exist, the AR continues moving back along its path, coloring it black («BACK_A»; «BACK_B») until it reaches a node incident to a marked isthmus or a node with other possible movement paths. In the latter case, the AR continues moving back along its path without coloring anything (sending a message «RETREAT_A»; «RETREAT_B» at each step) until it detects the nearest node incident to a marked isthmus.

If during the marking of isthmuses one isthmus has been marked, the AR colors the nearest incidentor of the marked isthmus in black («EXPLORED_AB»; «EXPLORED_BA») and then moves forward to the end of its path, marked in «own» color.

If, however, at least two isthmuses have been marked, the AR colors the nearest incidentor of the marked isthmus in «own» color («EXPLORED_AB»; «EXPLORED_BA»). The AR then moves backward along its path («RETREAT_A»; «RETREAT_B») until the next marked isthmus is found. In this case, if the marked isthmus is not the last one, the AR colors the nearest incidentor in black («EXPLORED_AB»; «EXPLORED_BA»), and on the next step, the AR returns backward along its path to the next marked isthmus («RETREAT_A»; «RETREAT_B»). If the marked isthmus is the last one, the AR colors the nearest incidentor in «own» color («EXPLORED_AB»; «EXPLORED_BA»). On the next step, the AR notifies the AE about the exploration of all marked isthmuses by the other AR («RESET_A»; «RESET_B»). After that, the AR moves along the last isthmus to the foreign area, coloring the nearest incidentor in black. On the next step, the AR moves along the first explored isthmus back to its area, coloring the distant incidentor in black. The AR then moves forward to the end of its path, marked in «own» color.

5. *Simultaneous arrival of two agents at the same white node.* In the case of both agents arriving at the same white node at the same time, each agent colors half of the node, making it red-yellow. On the next step, agent B steps back along its path, removes the paint applied on the previous step from the nearer incident edge and node, colors the farther incident edge black («RETURN_B»), and switches to the mode of marking isthmuses. The edge on which the transition was made is already counted as the first isthmus due to the AE message «RETURN_B», and the length of the yellow path is reduced by one node. Note that agent A treats the red-yellow node as a red node while working along its path.

Priority of switching the ARs to one of the possible operating modes is distributed as follows: first, we check for the presence of marked isthmuses, so the highest priority is given to the isthmus exploration mode; then we check if there are isthmuses from the current node that can be marked for exploration by agent A, so the second priority will be given to the isthmus marking mode; next, we check for the presence of back edges from the current node, so the third priority will be the back edge exploration mode; lastly, the regular operating mode has the lowest priority. The operating mode of the ARs in case of both arriving at the same white node simultaneously is not considered, as in this case, agent A continues its work in the regular mode, and for agent B, choosing any other mode will be unavailable at that moment.

Let us consider the algorithm of the agent-experimenter:

Input: lists of messages M and N from the ARs.

Output: a list of nodes V_H and edges E_H of the graph H, isomorphic to the graph G.

Data: V_H, E_H lists of nodes and edges of the graph H. Ct_A, Ct_B – counters of the number of visited nodes of the graph G by agents A and B, respectively. AN, BN – variables, where a value of «1» gives agents A and B, respectively, a signal to return and explore the isthmuses, while a value of «0» informs the agents

that the crossings marked for exploration by the specific agent are absent. N_A, N_B – variables that store the numbers of the nodes from which agents A and B, respectively, last marked the isthmuses. F,K – the number of isthmuses from nodes N_A, N_B respectively, marked for exploration. Q,Z – variables used certain subroutines for working with isthmuses to store the values of variables F,K respectively, as they were at the beginning of the subroutines. M,N – lists of messages from agents A and B, respectively. E,L – variables used to mark whether the isthmus was marked by agent A or B on the previous step (value «1») or not (value «0»). i, j – counters used by agents A and B, respectively, to determine the numbers of the second nodes of marked isthmuses or the numbers of the second nodes of marked back edges. $STOP\_A$, $STOP\_B$ – variables used by agents A and B, respectively, to signal to the AE the completion of exploring their subgraph. UDP_A, UDP_B – logical variables used by agents A and B, respectively, to determine the coloring method of the incidentors of the isthmus being considered at a given moment. $UDOBR\_A$, $UDOBR\_B$ – logical variables used by agents A and B, respectively, to determine whether the considered back edge is the last of the marked ones. $KOBR\_A$, $KOBR\_B$ – variables in which agents A and B, respectively, store the number of marked back edges. $r(1),r(2),...r(t)$ – a list of node numbers of the red path, where t is the length of this list. $y(1), y(2),...y(p)$ – a list of node numbers of the yellow path, where p is the length of this list. Mes – the processing message.

1. $AN := 0; BN := 0; N\_A := 0; N\_B := 0; F := 0; K := 0; M := \varnothing; N := \varnothing; E := 0; L := 0;$
$i := 0; j := 0; E_H := \varnothing; Ct\_A := 1; Ct\_B := 2; V_H := \{Ct\_A, Ct\_B\}; t := 1; p := 1;$
$r(t) := Ct\_A; y(p) := Ct\_B; UDP\_A := False;$ $UDP\_B := False;$
$UDOBR\_A := False;$ $UDOBR\_B := False;$ $KOBR\_A := 0;$ $KOBR\_B := 0;$
$STOP\_A := 0; STOP\_B := 0;$
2. while $(STOP\_A = 0) or (STOP\_B = 0)$ do
3. if $M \neq \varnothing$ then do
4. read the message in Mes and delete it from M;
5. LIST_PROC_A();
6. end do;
7. if $N \neq \varnothing$ then do
8. read the message in Mes and delete it from N;
9. LIST_PROC_B();
10. end do;
11. end do;
12. print V_H, E_H.

LIST_PROC_A():

1. if Mes = "FORWARD_A" then FORWARD_A();
2. if Mes = "BACK_A" then BACK_A();
3. if Mes = "STOP_A" then STOP_A();
4. if Mes = "MARK_BE_A" then MARK_BE_A();
5. if Mes = "RETREAT_A" then RETREAT_A();
6. if Mes = "FORWARD_BE_A" then FORWARD_BE_A();
7. if Mes = "BE_EXPLORED_A" then BE_EXPLORED_A();
8. if Mes = "MARK_AB" then MARK_AB();
9. if Mes = "FIX_A" then FIX_A();
10. if Mes = "EXPLORED_AB" then EXPLORED_AB();
11. if Mes = "RESET_A" then RESET_A();

FORWARD_A(): $Ct\_A := Ct\_A + 2; t := t + 1; r(t) := Ct\_A; V_H := V_H \cup \{Ct\_A\}$
$E_H := E_H \cup \{(r(t-1), r(t))\};$
BACK_A(): an element $r(t)$ is removed from the list $r(1),r(2),...r(t).$; $t := t - 1;$
STOP_A(): $STOP\_A := 1;$
MARK_BE_A(): $KOBR\_A := KOBR\_A + 1;$
RETREAT_A(): $i := i + 1;$
FORWARD_BE_A(): $KOBR\_A := KOBR\_A - 1; UDOBR := (KOBR\_A = 0);$
$E_H := E_H \cup \{(r(t), r(t-i))\};$
BE_EXPLORED_A(): $i := 0;$
MARK_AB(): $F := F + 1; E := 1;$
RESET_A(): $N\_A := Ct\_A; BN := 1; E := 0; Q := F; UDP\_A := (((F = Q) or (F = 1)) and (Q \neq 1));$
EXPLORED_AB(): $E_H := E_H \cup \{(N\_B, r(t-i))\}; K := K - 1;$
$UDP\_B := (((K = Z) or (K = 1)) and (Z \neq 1));$
ОБН_A(): $AN := 0; i := 0;$

The procedures for working with the message list from agent B that are not considered below are analogous to the procedures for working with the message list from agent A.

The procedure LIST_PROC_B() is the same as the procedure LIST_PROC_A(), except for an additional condition check and a subroutine call: «if *Mes*= "RETURN_B" then RETURN_B();», which applies only to agent B and relates to the mode of simultaneous arrival of two ARs at the same white node.

RETURN_B(): $E_H := E_H \setminus \{(y(p-1), y(p))\}; V_H := V_H \setminus \{Ct\_B\}; Ct\_B := Ct\_B - 2;$

$p := p-1; y(p) := Ct\_B; L := 1; K := K+1;$

Let us consider the properties of the presented graph exploration algorithm.

Provided that $n \geq 3$, the procedures FORWARD_A() and FORWARD_B() are each executed at least once.

In each of these procedures, one node of graph H is created. If both ARs simultaneously enter the same white node, these procedures will create two new nodes in graph H. The node created by agent B will be removed in the next step by the **RETURN_B()** procedure, as it duplicates the node created by agent A. Thus, the execution of the described algorithm induces a mapping $\varphi: V_G \to V_H$ of the nodes of graph G into the nodes of graph H. Moreover, $\varphi(v) = t$ (when node v is colored red and $t = Ct\_A$) and $\varphi(s) = p$ (when vertex s is colored yellow and $p = Ct\_B$). The given mapping naturally establishes an implicit numbering of the nodes of graph G. Moreover, the mapping φ is a bijection, since in a connected graph G, all nodes are reachable from the starting nodes. This means that all nodes are visited by the agents, i.e., they are colored red and yellow.

During the execution of the FORWARD_A() or FORWARD_B() procedure, the agent-experimenter explores the tree edge (v,u) and numbers node u in such a way that the edge (v,u) uniquely corresponds to the edge $(\varphi(v),\varphi(u))$ in graph H.

During the execution of the EXPLORED_AB() or EXPLORED_BA() procedures, the AE explores the isthmus (v,u) of graph G and assigns it a unique correspondence with the edge $(\varphi(v),\varphi(u))$ of graph H. Therefore, φ is an isomorphism from graph G to graph H.

Theorem 1. Three agents, upon executing the exploration algorithm on graph G, explore the graph up to isomorphism.

Let's calculate the time, space, and communication complexities of the algorithm in a uniform scale. When calculating the time complexity of the algorithm, we assume that the initialization of the algorithm, the analysis of the neighborhood of the working node, and the selection of one of the possible procedures take a constant number of time units. We also assume that the selection of edges, the traversal over them by the AR, and the processing of messages from the AE received at this stage from the AR are all done in one time unit.

From the description of the algorithm, it follows that at each step of the algorithm, the red (yellow) path is a simple path connecting the initial node v (or s in the case of agent B) with number $\varphi(v) = 1$ ($\varphi(s) = 2$) to the node u (z) with number $\varphi(u) = Ct\_A$ ($\varphi(z) = Ct\_B$). Therefore, the total length of the red and yellow paths does not exceed n. Each time the procedures of the normal operating mode are executed, each agent explores one edge. This means that the procedures of the normal operating mode are executed no more than $2 \times (n-1)$ times (considering that each AR will traverse each edge of its path twice). Therefore, the total execution time is estimated as $O(n)$.

During a single execution of the back edge exploration mode procedure, the agents mark no more than $n-2$ back edges (since the graph is simple and at least one node is colored in the 'foreign' color), traverse no more than $n-2$ edges of the red (yellow) path once, and traverse no more than $n-2$ back edges twice. The time spent on the back edge exploration mode is estimated as $4 \times n \times O(n)$, i.e., $O(n^2)$.

During the execution of the isthmus marking mode procedures, the agents do not traverse isthmuses but simply color their adjacent incidentors. After that, they send a message to the AE about the completion of the isthmus marking mode. It also takes one move. Thus, the working time in this mode is estimated as $n \times O(n) + O(n)$, i.e., as $O(n^2)$.

During a single execution of the isthmus exploration mode procedure, the agents traverse no more than $n-2$ edges of the red (yellow) path. After that, when marking the isthmuses as explored, the agents do not traverse the isthmus but simply color its adjacent incidentor. Once all marked isthmuses have been colored, the agents notify the AE, which takes one move. Upon returning after exploring all isthmuses, the agents may traverse at most two isthmuses once each, as well as no more than $n-2$ edges of the red (yellow) path. The execution time of this mode is estimated as $O(n^2) + O(n^2) + O(n) + O(n) + O(n^2)$, i.e., as $O(n^2)$.

The agents' idle time while waiting is estimated in total as $O(n^2)$. Consequently, the overall time complexity of the algorithm satisfies the relation $T(n) = O(n^2)$. Thus, the upper bound on the number of edge traversals performed by the agent-researchers is estimated as O(n). The space complexity of the algorithm is determined by the complexity of the lists $V_H, E_H, r(1)...r(t), y(1)...y(p),$, whose complexity is, in turn, defined by the values $O(n), O(n^2), O(n), O(n)$. Therefore, $S(n) = O(n^2)$.

At each step of the algorithm, the agents can send no more than four messages to the AE (including requests for values necessary to determine the operating mode) either separately or simultaneously and receive no more than three messages each. Therefore, the amount of information transmitted by the agents is estimated as $14 \times O\left(n^2\right)$. Consequently, $K(n) = O\left(n^2\right)$.

Taking into account the above assumptions about the method of calculating time complexity, the following theorem holds.

Theorem 2. The time, space, and communication complexities of the algorithm are $O\left(n^2\right)$, and the upper estimate of the number of transitions along the edges made by the agents is $O\left(n^2\right)$ where n is the number of nodes in the graph. The algorithm uses three colors.

Conclusions. This work presents a new method and a corresponding algorithm for graph exploration by a team of agents. The team consists of two agent-researchers that traverse an unknown graph and one stationary agent-experimenter, who explores the graph based on the results of the two traversing agents.

The main advantages of the proposed algorithm lie in the optimization of agent operations and communication processes, enabling the use of agent-researchers with finite memory independent of the graph size. This allows the same agents to be employed in environments where no prior information about the graph size is available. In our algorithm, only the stationary agent-experimenter possesses unboundedly growing internal memory, where the graph map is constructed.

Additionally, we analyzed the time, space, and communication complexities of the algorithm, which are bounded above by $O\left(n^2\right)$. The graph exploration process in our algorithm is implemented using only three colors. The number of colors does not depend on the graph parameters and remains a constant value.

Bibliography:

1. Banfi J., Quattrini Li, A., Rekleitis I. et al. Strategies for coordinated multirobot exploration with recurrent connectivity constraints. *Autonomous Robots*. 2018. Vol. 42. P. 875–894. https://doi.org/10.1007/s10514-017-9652-y

2. Dopp K. Automaten in labirinthen. *Electronische Informationsverarbeitung und Kybernetik*. 1971. V.7, № 2. P. 79–94.

3. Dudek G., Jenkin M. Computational principles of mobile robotics. *Cambridge Univ. press*. 2000. 280 p.

4. Dudek G., Jenkin M., Milios E., Wilkes D. Map validation in a graphlike world. *Proceedings of the 13th International Joint Conference on Artifical Intelligence.* San Fransisco: Morgan Kaufmann Publishers Inc., 1993. P. 1648–1653.

5. Goth G. Exploring new frontiers. *Communications of the ACM*. 2009. 52(11). C. 21–23.

6. Nagavarapu S. C., Vachhani L., Sinha A. et al. Generalizing Multi-agent Graph Exploration Techniques. *International Journal of Control, Automation and Systems*. 2020. https://doi.org/10.1007/s12555-019-0067-8

7. Sapunov S. V. Experiments on Recognition of Infinite Grid Graph Labelling. *Праці ІПММ НАН України*, 2021. 35 (1), 67–78. https://doi.org/10.37069/1683-4720-2021-35-6

8. Sapunov S. V. Minimal Deterministic Traversable Vertex Labelling of Infinite Square Grid Graph. Праці ІПММ НАН України, 2020. 34, 118–132. https://doi.org/10.37069/1683-4720-2020-34-12

9. Sapunov S. V. Directional Movement of a Collective of Compassless Automata on a Square Lattice of Width 2. Cybernetics and Systems Analysis, 2024. vol. 60, iss. 6, pp. 899–906. https://doi.org/10.1007/s10559-024-00727-x

10. Shannon C. E. Presentation of a maze-solving machine. *Cybernetics Trans, of the 8 th Conf. of the JosiahMacy Jr. Found* / Editor: H. Foerster. 1951. P. 173–180.

11. Stepkin A. Using a Collective of Agents for Exploration of Undirected Graphs. *Cybernetics and Systems Analysis*. V.51, 2. 2015 223–233. https://doi.org/10.1007/s10559-015-9715-z

12. Stopkin A. V. Algorithm for exploration of a simple undirected graph by a collective of agents. *Scientific Notes of Taurida VI Vernadsky National University*, 2024, vol. 35 (74), no. 5, pp. 303–309. https://doi.org/10.32782/2663-5941/2024.5.1/43

13. Stopkin A. V. Multi-agent system for non-oriented graphs exploration. *Information Technology and Society*, Kyiv, 2024, vol. 3, no. 14, pp. 38–43. https://doi.org/10.32689/maup.it.2024.3.5

14. Stopkin A. V. Finite graph exploration by a mobile agent. Mathematical Modeling and Computing, 2025, vol. 12, no. 1, pp. 75–82. https://doi.org/10.23939/mmc2025.01.075

15. Wang H., Jenkin M., Dymond P. Enhancing exploration in graph-like worlds. *Proceedings of the Canadian Conference on Computer and Robot Vision (CRV)*. 2008. P. 53–60.