УДК 519.1:519.7:007.5 DOI https://doi.org/10.35546/kntu2078-4481.2026.1.46

А. В. СТЬОПКІН
кандидат фізико-математичних наук, доцент,
доцент кафедри математики, фізики та інформатики
ДВНЗ «Донбаський державний педагогічний університет»
ORCID: 0000-0002-6130-9920

АЛГОРИТМ РОЗПІЗНАВАННЯ ГРАФІВ КОЛЕКТИВОМ АГЕНТІВ

Робота присвячена проблемі розпізнавання скінчених зв'язних неорієнтованих графів без петель та кратних ребер колективом агентів. Метою роботи є розробка нового методу розпізнавання неорієнтованих графів колективом агентів та побудова алгоритму, що базується на цьому методі. В роботі запропоновано наступну методологію до досягнення поставленої мети. Для розпізнавання використовується два агенти: нерухомий агент – агент-експериментатор, що знаходиться поза межами графа та виконує побудову мапи досліджуваного графа у вигляді списків ребер та вершин. Обчислення виконуються на базі повідомлень, які він отримує від іншого агента. Агент-експериментатор має скінчену на кожному кроці, але необмежено зростаючу внутрішню пам'ять. Другий агент – агент-дослідник, що представлений у вигляді мобільного агента, який може рухатися досліджуваним графом, зчитувати та змінювати відмітки на елементах графа, а також відправляти повідомлення агенту-експериментатору. Під час роботи агент-дослідник фарбуватиме необхідні елементи графа в чорний колір і наприкінці роботи всі елементи графа будуть пофарбовані. Агент-дослідник має скінчену пам'ять, яка не залежить від розмірності досліджуваного графа та використовує одну чорну фарбу. Функціонування колективу агентів забезпечується взаємодією двох принципово різних алгоритмів: алгоритму обходу графа агентом-дослідником та алгоритму побудови мапи графа агентом-експериментатором. Науковою новизною є отримання нового методу та алгоритму розпізнавання графів колективом агентів, який дозволяє використовувати для розпізнавання графів лише одну фарбу та дає можливість в подальшому використати даний алгоритм як основу для використання більшої кількості агентів-дослідників. Алгоритм має кубічну часову, квадратичну ємнісну та кубічну комунікаційну складності алгоритму розпізнавання, при цьому верхня оцінка числа переходів по ребрах, що здійснює агент-дослідник оцінюється як $O(n^3)$.

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

A. V. STOPKIN
Candidate of Physical and Mathematical Sciences, Associate Professor,
Associate Professor at the Department of Mathematics, Physics and Informatics
State Higher Educational Institution "Donbas State Pedagogical University"
ORCID: 0000-0002-6130-9920

GRAPH EXPLORATION ALGORITHM BY A TEAM OF AGENTS

This paper addresses the problem of the graph exploration of finite connected undirected graphs without loops and multiple edges by a team of agents. The aim of the study is to develop a new method for the exploration of undirected graphs by a team of agents and to construct an algorithm based on this method. The following methodology is proposed to achieve the stated objective. The exploration process involves two agents. The first one is a stationary agent – the agent-experimenter – located outside the graph and responsible for constructing a map of the explored graph in the form of node and edge lists. All computations are performed on the basis of messages received from the second agent. The agent-experimenter possesses finite memory at each step, which may grow without bound over time. The second agent – the agent-researcher – is a mobile agent capable of traversing the explored graph, reading and modifying marks on graph elements, and sending messages to the agent-experimenter. During the execution of the algorithm, the agent-researcher colors the explored graph elements black, and upon completion all graph elements are colored. The agent-researcher has finite memory independent of the size of the explored graph and uses a single black color. The functioning of the agent team is ensured by the interaction of two fundamentally different algorithms: a graph traversal algorithm executed by the agent-researcher and a graph mapping algorithm executed by the agent-experimenter. The scientific novelty of the work lies in the development of a new method and algorithm for graph exploration by a team of agents that requires only a single color for exploration and can serve as a foundation for further extensions involving multiple agent-researchers. The algorithm has cubic time complexity, quadratic space complexity, and cubic communication complexity for the graph exploration problem. The upper bound on the number of edge traversals performed by the agent-researcher is estimated as $O(n^3)$.

Key words: *graph exploration, undirected graph, mobile agent, exploration complexities, depth-first traversal method.*

© А. В. Стьопкін, 2026
Стаття поширюється на умовах відкритої ліцензії CC BY 4.0

Постановка проблеми

В наш час досить важливою проблемою кібернетики, якій приділяється особлива увага, залишається проблема взаємодії керованої та керуючої систем. В цій статті ми розглядаємо цю взаємодію як процес переміщення мобільного агента деяким середовищем, яке можна представити у вигляді скінченого неорієнтованого графа [1, 2]. Зрозуміло, що для можливості подібного переміщення мобільного агента операційним середовищем, повинна існувати достатньо повна модель даного середовища. Інакше цілеспрямоване переміщення агента по цьому середовищу буде неможливе. В питаннях моделювання операційних середовищ визначено ряд підходів, одним з яких є топологічний. В такому випадку агенту доступна інформація тільки про зв'язки між різними частинами середовища та недоступна метрична та алгоритмічна інформація про дане середовище [3]. Подібні ситуації досить часто виникають в робототехніці [4].

Метою роботи є побудова ефективного алгоритму розпізнавання невідомих скінчених неорієнтованих графів колективом агентів [5–7] та дослідження часової, ємнісної та комунікаційної складностей, а також кількості переходів по ребрах, які необхідно здійснити мобільному агенту, що рухається графом.

Аналіз останніх досліджень і публікацій

Залишаються актуальними дослідження поведінки агентів при виконанні завдань в деяких середовищах [1, 2, 5, 6, 8, 9]. Одним з них є завдання розпізнавання графів мобільними агентами [1, 2, 10], але дедалі більш популярними стають дослідження присвячені вивченню поведінки саме колективів агентів при роботі на графах [5–9]. При цьому детально аналізуються ключові аспекти, притаманні функціонуванню таких систем: проблема керування колективом агентів [11], проблема формування колективу агентів [12] та проблема консенсусу мультиагентної системи [13].

Формулювання мети дослідження

Як можна бачити, сучасні дослідження демонструють стабільний інтерес до проблеми розпізнавання графів мобільними агентами, що зумовлено широким спектром практичних застосувань та теоретичною складністю задачі. Особлива увага приділяється питанням побудови маршрутів руху агентів у невідомому середовищі, локальної розмітки його елементів, збору та обробки інформації в умовах обмежених ресурсів, формування повного представлення графа на основі локальних спостережень. А також питанням, пов'язаним з функціонуванням колективів агентів, таким як проблема керування колективом, проблема формування колективу та проблема консенсусу мультиагентної системи. Не менш важливими залишаються задачі мінімізації витрат часу, пам'яті та інших ресурсів, що використовуються під час розпізнавання.

Метою даної роботи є розробка та дослідження ефективного методу розпізнавання графів, а також створення алгоритму, що реалізує запропонований метод. У роботі планується дослідити часову, ємнісну та комунікаційну складності, а також кількість переходів по ребрах графа, які необхідно здійснити мобільному агенту для повного розпізнавання графа. Під ефективністю методу розуміється зменшення ресурсних витрат або покращення верхніх оцінок складності у порівнянні з відомими результатами, представленими в попередніх дослідженнях [7,14,15].

Викладення основного матеріалу дослідження

В роботі розглядаються скінчені, неорієнтовані графи без петель та кратних ребер. Нехай $G = (V, E)$ – граф, у якого V – множина вершин, E – множина ребер. Під ребрами ми будемо розуміти двоелементні підмножини (v, u), де $u, v \in V$. Трійку$((v, u), u)$ будемо називати інцидентором ребра(v, u) та вершини u. Інцидентор п суті є точкою з'єднання ребра з вершиною. Множину інциденторів позначимо I. Множину $L = V \cup E \cup I$ назвемо множиною всіх елементів графа G. Функцією розфарбування графа G назвемо сюр'єктивне відображення $\mu : L \to \{w, b\}$, де w – білий колір, а b – чорний. Пара(G, μ) називається розфарбованим графом. Послідовність $u_1, u_2, \ldots, u_k$ попарно суміжних вершин називається шляхом в графі G, а k – довжиною шляху. За умови $u_1 = u_k$ шлях називається циклом. Околом $Q(v)$ вершини v будемо називати множину елементів графа, що складається з вершини v, всіх вершин u суміжних з v, всіх ребер(v, u) та всіх інциденторів$((v, u), v)$, $((v, u), u)$. Потужність множин вершин V і ребер E позначимо через n і m відповідно. Зрозуміло, що $m \le \dfrac{n(n-1)}{2}$. Ізоморфізмом графа G і графа H назвемо таку бієкцію: $\phi : V_G \to V_H$, що $(v, u) \in E_G$ тоді й тільки тоді, коли $(\phi(v), \phi(u)) \in E_H$. Таким чином, ізоморфні графи рівні з точністю до позначення вершин і розфарбування їх елементів.

Для розпізнавання графів використовується два агенти: нерухомий агент – агент-експериментатор, що знаходиться поза межами графа та виконує всі обчислення, які необхідні для побудови мапи досліджуваного графа. Обчислення виконуються на базі повідомлень, які він отримує від іншого агента. Агент-експериментатор має скінчену на кожному кроці, але необмежено зростаючу внутрішню пам'ять. Другий агент – агент-дослідник, що представлений у вигляді мобільного агента, який може рухатися досліджуваним графом, зчитувати та змінювати відмітки на елементах графа, а також відправляти повідомлення про свої дії агенту-експерименатору. Агент-дослідник має скінчену пам'ять, яка не залежить від розмірності досліджуваного графа та використовує одну чорну фарбу.

Функціонування колективу агентів забезпечується взаємодією двох принципово різних алгоритмів: алгоритму обходу графа агентом-дослідником та алгоритму побудови мапи графа агентом-експериментатором. Ця взаємодія відбувається наступним чином: агент-дослідник виконує деяку дію з алгоритму обходу та відправляє повідомлення агенту-експериментатору, повідомлення обробляється та, за необхідності, відправляється відповідь агенту-досліднику, далі агент-дослідник знову робить крок і т.д.

На початку роботи алгоритму всі елементи графа пофарбовані в чорний колір. Далі агент-дослідник розміщується в довільній вершині графа і одразу зафарбовує її в чорний колір, а агент-дослідник додає до своєї пам'яті першу вершину. Під час роботи агент-дослідник фарбуватиме необхідні елементи в чорний колір і наприкінці роботи всі елементи графа будуть пофарбовані.

Для зручності читання наведемо принцип функціонування колективу агентів за допомогою режимів, після чого наведемо безпосередньо самі алгоритми роботи різних агентів. При описі режимів функціонування мультиагентної системи ми будемо зазначати повідомлення, які відправляються агентом-дослідником агенту-експериментатору у форматі: «Message», де Message – текст повідомлення. Також поточною вершиною будемо називати вершину, в якій знаходиться агент-дослідник.

На початку роботи агент-дослідник працює в *звичайному режимі роботи*. Він сканує окіл поточної вершини, знаходить білі вершини, до яких ведуть білі ребра з білими інцидентoрами і переходить по цим ребрам у знайдені вершини, фарбуючи при цьому пройдені ребра, дальні інцидентори та вершину, в яку він потрапив в чорний колір. На кожному кроці агенту-експериментатору відправляється повідомлення «Forward». Завдяки чому агент-дослідник додає в свою пам'ять до множин V і E нові вершини та ребра відповідно. Якщо при роботі в звичайному режимі агент-дослідник знаходить біле ребро з білими ближнім та дальніми інциденторами, яке веде в чорну вершину, то це означає, що знайдено зворотне ребро. В такому випадку агент-дослідник призупиняє роботу в звичайному режимі та перемикається в режим розпізнавання зворотних ребер, в якому функціонуватиме до того моменту, поки не розпізнає всі зворотні ребра з поточної вершини. Після чого знову буде активовано звичайний режим роботи. Зрозуміло, враховуючи, що граф скінчений настане момент, коли в околі поточної вершини не виявиться білих вершин до яких ведуть білі ребра. В такому випадку агент-дослідник сканує окіл поточної вершини, знаходить чорне ребро з ближнім чорним інцидентором, яке веде в чорну вершину і починає рух назад по таким ребрам, фарбуючи дальні інцидентори в чорний колір і відправляючи агенту-експериментатору кожен такий крок повідомлення «Back». Це відбувається до тих пір, поки в поточній вершині не виявиться ще не розпізнаних вершин та ребер, в яких знову почнеться рух агента-дослідника вперед. Якщо ж рухаючись назад агент-дослідник потрапив в стартову вершину і в її околі відсутні нерозпізнані елементи графа, він відправляє агенту-експериментатору повідомлення «Stop» і завершує роботу, так як досліджуваний граф повністю розпізнано. В цей момент агент-експериментатор в своїй пам'яті вже має повний перелік вершин та ребер досліджуваного графа у вигляді списку ребер та списку вершин.

Режим розпізнавання зворотних ребер. Як вже зазначалось вище під час роботи в звичайному режимі агент-дослідник може зустріти так зване зворотне ребро (біле ребро з білими ближнім та дальнім інциденторами, що веде в чорну вершину). Зустрівши таке ребро, агент-дослідник, одразу перемикається в режим розпізнавання зворотних ребер. Він переходить по знайденому ребру, фарбуючи його та дальній інцидентор в чорний колір. Після чого повертається по своєму раніше пройденому шляху в вершину, з якої почав розпізнавання зворотного ребра. На кожному кроці цього руху він відправляє агенту-експериментатору повідомлення «Forward_ie». Повідомлення відправляється для підрахунку кількості кроків, які агенту-досліднику необхідно зробити назад для потрапляння у дальню вершину поміченого зворотного ребра. Знання кількості кроків дозволить агенту-експериментатору обчислити номер дальньої вершини поміченого зворотного ребра, який необхідно знати щоб додати зворотне ребро в множину ребер. Повернувшись у вершину з якої розпочалося розпізнавання зворотного ребра, агент-дослідник зафарбовує ближній інцидентор цього ребра та відправляє повідомлення «Add_ie». Агент-експериментатор, отримавши це повідомлення, визначає номер дальньої вершини зворотного ребра та додає ребро в список в список ребер у своїй пам'яті. Після чого, агент-дослідник перевіряє наявність інших нерозпізнаних зворотних ребер з даної вершини і якщо вони є аналогічно розпізнає їх, а якщо немає, то продовжує роботу в звичайному режимі.

Тепер розглянемо алгоритми функціонування кожного агента окремо.

Алгоритм роботи агента-дослідника.

1. $\mu(v) := b$;
2. if $\exists\ (v, u) \in Q(v) | (\mu(v, u) = w)$ and $(\mu(u) = b)$ and $(\mu((v, u), v) = w)$ and $(\mu((v, u), u) = w)$ then do
3. EXPL_IE(v);
4. go to 2;
5. end do;
6. else if $\exists(v, u) \in Q(v) | (\mu(v, u) = w)$ and $(\mu(u) = w)$ then do
7. FORWARD(v);
8. go to 2;
9. end do;

10. else if $\exists(v, u) \in Q(v)|(\mu((v, \mathrm{u}), v) = b)$ and $(\mu((\mathrm{v}, \mathrm{u}), u) = w)$ and $(\mu(v, u) = b$ then do
11. BACK(v);
12. go to 6;
13. end do;
14. else STOP(v);

EXPL_IE(v):
1. if $\exists(v, u) \in Q(v)|(\mu(v, u) = w)$ and $(\mu(u) = b)$ and $(\mu((v, u), v) = w)$ and $(\mu((v, u), u) = w$ then do
2. агент обирає $(v, u) \in Q(v)|(\mu(v, u) = w)$ and $(\mu(u) = b)$ and $(\mu((v, u), v) = \mathrm{w})$ and $(\mu((v, u), u) = w)$ та переходить по ньому в вершину u;
3. $\mu((v, u), v) := b$;
4. $\mu(v, u) := b$;
5. $\mu((v, u), u) := b$;
6. if $\exists(v, u) \in Q(v)|(\mu((v, u), v) = w)$ and $(\mu(v, u) = b)$ and $(\mu((v, u), u) = b$ then do
7. агент обирає$(v, u) \in Q(v)|(\mu((v, u), v) = w)$ and $(\mu(v, u) = b)$ and $(\mu((v, u), u) = b)$ та переходить по ньому в вершину *u;*
8. $v := u$;
9. агент записує в список *LM* повідомлення: Forward_ie;
10. go to 6 даної процедури;
11. end do;
12. else do
13. агент записує в список *LM* повідомлення: Add_ie;
14. go to 1 даної процедури;
15. end do;

FORWARD(v):
1. Агент обирає $(v, u) \in Q(v)|(\mu(v, u) = w)$ and $(\mu(u) = w)$ та переходить по ньому в вершину u, фарбуючи $\mu(v, u) := b$, $\mu((v, u), u) := b$, $\mu(u) := b$;
2. $v := u$;
3. агент записує в *LM* повідомлення: Forward;

BACK(v):
1. Агент обирає $(v, u) \in Q(v)|(\mu((v, u), v) = b)$ and $(\mu((v, u), u) = w)$ та переходить по ньому в вершину u, фарбуючи $\mu((v, u), u) := b$;
2. $v := u$;
3. агент записує в *LM* повідомлення: Back;

STOP(v):
1. Агент фарбує $\mu(v) := b$ та завершує роботу;
2. агент записує в *LM* повідомлення: Stop;

Алгоритм роботи агента-експериментатора:

Вхід: список повідомлень від агента-дослідника.

Вихід: список вершин V_H і список ребер E_H графа H, ізоморфного графу G.

Дані: V_H – список вершин графа. E_H – список ребер графа H. ct – лічильник числа відвіданих агентом-дослідником вершин. i – лічильник, що використовується при визначенні номерів дальніх вершин знайдених зворотних ребер. *STOP* – змінна, що використовується агентом-дослідником для повідомлення агента-експериментатора, про завершення розпізнавання графа. *work*(1), *work*(2), …, *work*(t) – список номерів вершин робочого шляху, де t – довжина цього списку. *Mes* – повідомлення, яке оброблюється в даний момент.

Алгоритм:
1. $LM := \varnothing$; $ct := 1$, $i := 0$, $E_H := \varnothing$, $t := 1$, $work(t) := ct$, $V_H := \{1\}$;
3. while $LM \neq \varnothing$ do
4. if $LM \neq \varnothing$ then do
5. читаємо повідомлення в *Mes*;
6. $LM := LM \backslash Mes$;
7. List_Processing ();
8. end do;
9. print V_H, E_H.

List_processing_A():
1. if *Mes* = "Forward" then Forward();
2. if *Mes* = "Back" then Back();
3. if *Mes* = "Forward_ie" then Forward_ie();

4. if *Mes* = “Add_ie” then Add_ie();
5. if *Mes* = “Stop” then Stop().

Forward(): $ct := ct + 1$; $t := t + 1$; $work(t) := ct$; $V_H := V_H \cup \{ct\}$; $E_H := E_H \cup \{(work(t-1), work(t))\}$;
Back(): *delete* $work(t)$; $t := t - 1$;
Forward_ie(): $i := i + 1$;
Add_ie(): $E_H := E_H \cup \{(work(t), work(t-i))\}$; $i := 0$;
Stop(): The agent finishes the operation;
Аналіз роботи мультиагентної системи.

При виконанні свого алгоритму роботи, за умови, що $n \geq 3$ не менше двох разів агент-дослідник виконає процедуру FORWARD(v) при виконанні якої будуть відвідані нові білі вершини досліджуваного графа G. Після виконання цієї процедури, в пам’ять агента-експериментатора кожного разу додається нова вершина графа H. Таким чином, поступово формується неявна нумерація φ: $V_G \to V_H$, де рівність $\varphi(v) = ct$ встановлюється, коли вершина v фарбується в чорний колір. Зазначена нумерація є бієкцією, оскільки в зв’язному графі G між будь-якими вершинами знайдеться шлях, а отже агентом-дослідником будуть відвідані всі вершини досліджуваного графа. Враховуючи те, що робота алгоритму базується на методі обходу в глибину, всі ребра графа поділяються на ребра, що належать дереву та зворотні ребра. Виконуючи процедуру FORWARD(v) агент-дослідник також розпізнає одне ребро дерева (v, u) і так нумерує вершину u, що ребру (v, u) графа G однозначно відповідає ребро ($\varphi(v)$, $\varphi(u)$) графа H. Виконуючи процедуру *EXPL_IE*(v), агент розпізнає всі зворотні ребра (v, u) з поточної вершини v графа G та ставить їм в однозначну відповідність ребра ($\varphi(v)$, $\varphi(u)$) графа H. Таким чином, відображення φ є ізоморфізмом графа G на граф H. Враховуючи сказане має місце наступна теорема.

Теорема 1. *Мультиагентна система, виконавши алгоритм розпізнавання графа, розпізнає будь-який граф з точністю до ізоморфізму.*

Обчислимо складності роботи побудованого алгоритму. Розглянемо часову, ємнісну та комунікаційну складності, також підрахуємо кількість переходів по ребрах, які необхідно здійснити агенту-досліднику для повного розпізнавання досліджуваного графа. Зазначимо, що при обчисленні будемо вважати, що перехід агента-дослідника по ребру займає час, що дорівнює деякій константі, а обробка одного повідомлення агентом-експериментатором відбувається не довше переходу агента-дослідника з однієї вершину в іншу.

Загальний час, який буде витрачено агентом-дослідником на аналіз околу робочої вершини $Q(v)$ та вибір необхідних ребер оцінюється зверху як $O(n^2)$. При роботі агента-дослідника, в будь-який момент часу є деякий шлях, від початкової вершини агента-дослідника до його поточної вершини – це простий шлях, від вершини з номером $\varphi(v) = 1$ до вершини з номером $\varphi(u) = ct$. Зрозуміло, що довжина будь-якого такого шляху не перевищує n. Таким чином, легко бачити, що на одноразове виконання кожної з процедур *FORWARD*(v) та *BACK*(v) агент-дослідник витрачає один крок, проходячи при цьому одне ребро. При одноразовому виконанні процедури *EXPL_IE*(v) агент дослідник проходить одне зворотне ребро та не більше ніж $n - 1$ ребро вперед. В кінці процедури відбувається відправка повідомлення для додавання зворотного ребра в список ребер, на що також витрачається один крок. З кожної вершини можливо розпізнати не більше $n - 2$ зворотних ребра. Всього ж вершин, з яких можуть бути зворотні ребра не більше $n - 2$. Таким чином, співвідношення, що визначають часову складність алгоритму виглядають наступним чином:

1. Процедура *FORWARD*(v) виконується максимум $n - 1$ раз і загальний час її виконання оцінюється зверху як $O(n)$.
2. Процедура *BACK*(v) виконується не більше ніж $n - 1$ раз та загальний час її виконання оцінюється як $O(n)$.
3. Процедура *EXPL_IE*(v) виконується не більше $n - 2$ раз, кожен з яких займає не більше n одиниць часу, тобто загальний час виконання процедури оцінюється як $(n - 2)^2 \times n$, тобто $O(n^3)$.
4. Процедура *STOP*(v) виконується один раз і її асимптотична складність дорівнює $O(1)$.

Таким чином загальна кількість числа переходів по ребрах, що здійснює агент-дослідник не перевищує $O(n) + O(n) + O(n^3) + O(1)$, тобто верхня оцінка числа переходів задовольняє співвідношенню: $M(n) = O(n^3)$.

В той же час верхня оцінка часової складності запропонованого алгоритму задовольняє співвідношення: $T(n) = O(n^3)$.

Ємнісна складність $S(n)$ запропонованого алгоритму визначається складністю списків V_H, E_H, $work(1), \ldots, work(t)$, складність яких визначається відповідно величинами $O(n)$ $O(n^2)$, $O(n)$, а отже $S(n) = O(n^2)$.

Також, враховуючи, що на кожному кроці агент-дослідник може відправити не більше одного повідомлення, яке не містить ніякої інформації граф (тобто його об’єм можна вважати рівним константі), легко бачити, що об’єм переданої інформації оцінюється зверху як $O(n^3)$. Отже, $K(n) = O(n^3)$. Враховуючи сказане, має місце наступна теорема.

Теорема 2. *Часова складність алгоритму розпізнавання дорівнює* $O(n^3)$, *ємнісна складність –* $O(n^2)$, *комунікаційна складність складає* $O(n^3)$, *верхня оцінка числа переходів по ребрах, що здійснює агент-дослідник оцінюється як* $O(n^3)$. *При цьому алгоритм використовує одну фарбу.*

Висновки

В роботі запропоновано новий алгоритм розпізнавання простих зв'язних скінчених неорієнтованих графів колективом агентів. Алгоритм має кубічну часову, квадратичну ємнісну та кубічну комунікаційну складності. Верхня оцінка числа переходів по ребрах, які необхідно виконати агенту-досліднику для повного розпізнавання досліджуваного графа – $O(n^3)$. Мобільний агент має скінчену внутрішню пам'ять, яка не залежить від розмірності досліджуваного графа.

Список використаної літератури

1. 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
2. Стьопкін А. В. Розпізнавання зв'язних неорієнтованих графів мобільним агентом. *Вісник Херсонського національного технічного університету*. Одеса, 2025. Т.2 1(92). С. 215–221. DOI: https://doi.org/10.35546/kntu2078-4481.2025.1.2.29
3. Kuipers B. The spatial semantic hierarchy. *Artificial Intelligence*. 2000. V. 119, № 1–2. pp. 191–233.
4. Dudek G, Jenkin M. Computational principles of mobile robotics. Cambridge, 2000. – 280 p.
5. 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
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. Стьопкін А. В. Алгоритм розпізнавання простих неорієнтованих графів колективом агентів. *Вчені записки таврійського національного університету імені В. І. Вернадського Серія: Технічні науки*. Київ, 2024. Т. 35 (74) № 5. С. 303–309. https://doi.org/10.32782/2663-5941/2024.5.1/43
8. Stopkin A. V. Exploration of simple graphs by a collective of agents. *Mathematical Modeling and Computing*, 2025. Vol. 12, No. 2, pp. 603–610 https://doi.org/10.23939/mmc2025.02.603
9. Dey, S.; Xu, H. Intelligent Distributed Swarm Control for Large-Scale Multi-UAV Systems: A Hierarchical Learning Approach. *Electronics* 2023, 12(1):89. https://doi.org/10.3390/electronics12010089
10. Dhar A.K., Gorain B., Mondal K., Patra Sh., Singh R. Edge exploration of anonymous graph by mobile agent with external help. *Computing*, 2023. V.105, 483-506. https://doi.org/10.1007/s00607-022-01136-8
11. Dey, S.; Xu, H. Intelligent Distributed Swarm Control for Large-Scale Multi-UAV Systems: A Hierarchical Learning Approach. *Electronics* 2023, 12(1):89. https://doi.org/10.3390/electronics12010089
12. Dongyu Li, Shuzhi Sam Ge, Wei He, Guangfu Ma, Lihua Xie. Multilayer formation control of multi-agent systems. *Automatica*, 2019. V 109. https://doi.org/10.1016/j.automatica.2019.108558
13. Amirkhani, A., Barshooi, A.H. Consensus in multi-agent systems: a review. *Artif Intell*. Rev 55, 3897–3935 (2022). https://doi.org/10.1007/s10462-021-10097-x
14. Стьопкін А. В. Мультиагентна система розпізнавання неорієнтованих графів. *Інформаційні технології та суспільство*. Київ, 2024. Вип. 3 (14). 38-43 с. DOI: https://doi.org/10.32689/maup.it.2024.3.5
15. Grunsky, I. S., & Tatarinov, E. A. (2009). Recognition of a finite graph by a wandering agent. Bulletin of Donetsk National University, 1, 492–497.

References

1. Stopkin, A. V. (2025). Finite graph exploration by a mobile agent. *Mathematical Modeling and Computing*, 12(1), 75–82. https://doi.org/10.23939/mmc2025.01.075
2. Stepkin, A. V. (2025). Graph exploration of connected undirected graphs by a mobile agent. *Visnyk of Kherson National Technical University*, 2(1), 215–221. https://doi.org/10.35546/kntu2078-4481.2025.1.2.29
3. Kuipers, B. (2000). The spatial semantic hierarchy. *Artificial Intelligence, 119*(1–2), 191–233.
4. Dudek, G., & Jenkin, M. (2000). *Computational principles of mobile robotics*. Cambridge University Press., p. 280
5. Sapunov, S. V. (2024). Directional movement of a collective of compassless automata on a square lattice of width 2. *Cybernetics and Systems Analysis, 60*(6), 899–906. https://doi.org/10.1007/s10559-024-00727-x
6. Nagavarapu, S. C., Vachhani, L., Sinha, A., et al. (2020). Generalizing multi-agent graph exploration techniques. *International Journal of Control, Automation and Systems*. https://doi.org/10.1007/s12555-019-0067-8
7. Stepkin, A. V. (2024). Algorithm for recognizing simple undirected graphs by a team of agents. *Scientific Notes of the V. I. Vernadsky Taurida National University. Series: Technical Sciences, 35(74)*, 303–309. https://doi.org/10.32782/2663-5941/2024.5.1/43
8. Stopkin, A. V. (2025). Exploration of simple graphs by a collective of agents. *Mathematical Modeling and Computing, 12*(2), 603–610. https://doi.org/10.23939/mmc2025.02.603
9. Dey, S.; Xu, H. (2023). Intelligent Distributed Swarm Control for Large-Scale Multi-UAV Systems: A Hierarchical Learning Approach. *Electronics*, *12*(1):*89*. https://doi.org/10.3390/electronics12010089

10. Dhar, A. K., Gorain, B., Mondal, K., Patra, S., & Singh, R. (2023). Edge exploration of anonymous graph by mobile agent with external help. *Computing, 105*, 483–506. https://doi.org/10.1007/s00607-022-01136-8
11. Dey, S., & Xu, H. (2023). Intelligent distributed swarm control for large-scale multi-UAV systems: A hierarchical learning approach. *Electronics, 12*(1), 89. https://doi.org/10.3390/electronics12010089
12. Dongyu Li, Shuzhi Sam Ge, Wei He, Guangfu Ma, Lihua Xie. (2019) Multilayer formation control of multi-agent systems. *Automatica. V 109*. https://doi.org/10.1016/j.automatica.2019.108558
13. Amirkhani, A., & Barshooi, A. H. (2022). Consensus in multi-agent systems: A review. *Artificial Intelligence Review, 55*, 3897–3935. https://doi.org/10.1007/s10462-021-10097-x
14. Stepkin, A. V. (2024). Multi-agent system for recognizing undirected graphs. *Information Technologies and Society*, 3(14), 38–43. DOI: https://doi.org/10.32689/maup.it.2024.3.5
15. Grunsky, I. S., & Tatarinov, E. A. (2009). Recognition of a finite graph by a wandering agent. *Bulletin of Donetsk National University*, 1, 492–497.

Дата першого надходження статті до видання: 09.01.2026
Дата прийняття статті до друку після рецензування: 13.02.2026
Дата публікації (оприлюднення) статті: 30.04.2026