

А. В. СТЬОПКІН

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

РОЗПІЗНАВАННЯ ЗВ'ЯЗНИХ НЕОРІЄНТОВАНИХ ГРАФІВ МОБІЛЬНИМ АГЕНТОМ

Дослідження присвячено проблемі розпізнавання скінчених зв'язних неорієнтованих графів без петель та кратних ребер мобільним агентом. Метою роботи є побудова нового ефективного методу розпізнавання графів та алгоритму, що базується на цьому методі. В роботі запропоновано наступну методологію до досягнення поставленої мети. Використати мобільного агента, який може пересуватися по графу, зчитувати й залишати мітки на елементах графа. Також агент має скінчену на кожному кроці, але необмежено зростаючу внутрішню пам'ять (ємність пам'яті залежить від досліджуваного графа) та для розпізнавання графу використовує фарбу одного кольору. На основі даних, отриманих при переміщенні по графу, мобільний агент в своїй пам'яті поступово вибудовує представлення досліджуваного графа списком ребер і списком вершин. Алгоритм розпізнавання графа базується на методі обходу графа в глибину. У статті детально розглянуто режими роботи мобільного агента із зазначенням пріоритетності активації цих режимів в процесі роботи. Також в роботі проведено аналіз часової й ємнісної складностей побудованого алгоритму та проаналізовано кількість переходів по ребрах, які необхідно виконати мобільному агенту для повного розпізнавання досліджуваного графа. Науковою новизною є отримання нового більш ефективного методу та алгоритму розпізнавання графів, що базується на цьому методі, який дозволяє використовувати для розпізнавання графів лише одну фарбу та дає можливість в подальшому використати даний алгоритм як основу для роботи мультиагентної системи. Алгоритм має квадратичну часову й квадратичну ємнісну складності алгоритму розпізнавання, при цьому верхня оцінка числа переходів по ребрах, що здійснює мобільний агент оцінюється як $O(n^2)$.

Ключові слова: розпізнавання графа, неорієнтований граф, мобільний агент, складності 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
SHEI "Donbas State Pedagogical University"
ORCID: 0000-0002-6130-9920

GRAPH EXPLORATION OF CONNECTED UNDIRECTED GRAPHS BY A MOBILE AGENT

The research addresses the problem of exploring finite connected undirected graphs without loops and multiple edges by a mobile agent. The goal of the study is to develop a new efficient method for graph exploration and an algorithm based on this method. The proposed methodology involves the use of a mobile agent capable of traversing the graph, reading, and leaving markers on graph elements. The agent has finite memory at each step, but its capacity can grow unboundedly depending on the graph being explored. The exploration process is performed using a single color of paint. Based on the data collected during graph traversal, the mobile agent gradually constructs a representation of the explored graph in its memory in the form of an edge list and a node list. The graph exploration algorithm based on the depth-first traversal method. The article provides a detailed analysis of the operational modes of the mobile agent, highlighting the priority of activating these modes during the exploration process. Additionally, the study analyzes the time and space complexities of the proposed algorithm and evaluates the number of edge transitions required for the mobile agent to fully explore the graph. The scientific novelty lies in the development of a new efficient method and algorithm for graph exploration. This method requires only a single color of paint and serves as a foundation for the development of multi-agent systems in the future. The algorithm demonstrates quadratic time and quadratic memory complexities, while the upper bound of the number of edge transitions performed by the mobile agent is estimated as $O(n^2)$.

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

Постановка проблеми

В наш час досить інтенсивно розвивається така прикладна наука як робототехніка. З'являються різного роду роботи, які автоматизують виконання тих чи інших робіт. Досить часто роботами замінюють працю людини у важкодоступних або небезпечних середовищах. Звісно, що роботизація процесів не обійшла й такий важливий напрямок як дослідження невідомих середовищ [1]. Одним з найвідоміших прикладів такого дослідження можна

вважати дослідження поверхні марса різноманітними марсоходами. Звісно, що недосліджені середовища є не тільки в космосі, але все ще залишаються й на поверхні Землі, що свідчить про актуальність вивчення проблеми розпізнавання середовищ засобами робототехніки. Як відомо більшість середовищ можна представити у вигляді графа, що в свою чергу робить актуальним вивчення проблеми розпізнавання графів мобільними агентами.

Сучасні дослідження в основному присвячені вивченню роботи колективів агентів на графах [2–6], але зрозуміло, що в оптимізації роботи колективу агентів велику роль відіграє оптимізація роботи кожного окремого агента [7]. А тому дослідження проблеми розпізнавання графів одним агентом залишається актуальним питанням.

Аналіз останніх досліджень і публікацій

Початком досліджень в напрямі вивчення поведінки автоматів на графах прийнято вважати роботу 1951 року С. Е. Shannon [8], в якій розглядалась задача пошуку автоматом заданої цілі в лабіринті. Трохи пізніше у 1971 році з'являються роботи К. Дорр [9], присвячені обходу шахових лабіринтів скінченими автоматами. Після чого почали активно вивчати поведінку автоматів в лабіринтах, поступово розширюючи задачі автоматів. У 1993 році групою вчених на чолі з G. Dudek проводилися дослідження [10] в області аналізу властивостей невідомого середовища [11] при різних способах взаємодії автомата з операційним середовищем, а також при різній апріорній інформації про нього. Паралельно виходять роботи цих вчених з дослідження роботи роїв агентів [12, 13].

Серед основних проблем вирішенню яких приділялося багато уваги можна виокремити проблему самолокалізації агента, проблему контролю карти та проблему повного розпізнавання графів. Зупинимось більш детально на останній проблемі, вирішенню якої приділяється багато уваги. Визначено різноманітні підходи та запропоновано ряд алгоритмів переміщення агентів по графу, фарбування його елементів чи позначення каменями, які дозволяють розпізнати досліджуваний граф з точністю до ізоморфізму. Наприклад, в роботі [10] запропоновано метод, в якому агент, при переміщенні фарбує тільки інцидентори відвіданих ребер. Для розпізнавання ребер агент переходить по ним, після чого найкоротшим шляхом повертається в стартову вершину. При цьому запам'ятовує мітки відвіданих інциденторів, які дозволяють однозначно ідентифікувати вершину, в яку потрапив агент, здійснюючи перехід по досліджуваному ребру. В роботі запропоновано алгоритм розпізнавання графів, що оснований на стратегії пошуку в глибину [14]. При роботі алгоритму агент на кожному кроці володіє інформацією про кольори елементів з околу робочої вершини та при русі через вершини графа він створює неявну нумерацію відвіданих вершин. На основі цієї нумерації й відбувається розпізнавання графа. На основі даного дослідження, починаючи з 2009 року з'являються дослідження алгоритмів роботи колективів агентів по розпізнаванню графів [5, 6, 14–17].

Формулювання мети дослідження

Як можна бачити з публікацій, проблема розпізнавання графів досить інтенсивно вивчається та залишається актуальним проведення експериментів по створенню маршрутів руху агентів по невідомому графу, розмітки його елементів, збору та обробки локальної інформації щодо графу та способів побудови представлення графа за цією інформацією. Також не втрачають своєї актуальності задачі, направлені на пошук методів оптимізації витрат ресурсів, навантаження на канал зв'язку і т. д.

Метою роботи є створення ефективного методу розпізнавання графу та побудова алгоритму розпізнавання, що базується на даному методі, з подальшим визначенням часової, ємнісної складностей та кількості переходів по ребрах, які здійснює мобільний агент під час роботи алгоритму. Під ефективністю методу ми розуміємо покращення складностей чи зниження витрат ресурсів в порівнянні з раніше опублікованими роботами [5, 6, 15–17].

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 p суті є точкою з'єднання ребра з вершиною. Множину інциденторів позначимо I . Множину $L = E \cup I$ назовемо множиною всіх елементів графа G . Функцією розфарбування графа G назовемо сюр'єктивне відображення $\mu: L \rightarrow \{w, b\}$, де w – білий колір, а b – чорний. Пара (G, μ) називається розфарбованим графом. Послідовність $u_1, u_2, \dots, u_k$ 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 $m \leq \frac{n(n-1)}{2}$. Ізоморфізмом графа G і графа H назовемо таку бієкцію: $\varphi: V_G \rightarrow V_H$, що $(v, u) \in E_G$ тоді й тільки тоді, коли $(\varphi(v), \varphi(u)) \in E_H$. Таким чином, ізоморфні графи рівні з точністю до позначення вершин і розфарбування їх елементів.

Для розпізнавання графів використовується один мобільний агент, який може рухатися по графу, зчитувати та змінювати відмітки на елементах графа, а також будувати представлення досліджуваного графа у вигляді списків вершин та ребер у своїй пам'яті. Мобільний агент має скінчену на кожному кроці, але необмежено зростаючу внутрішню пам'ять. Агент має одну чорну краску.

На початку роботи алгоритму агент розміщується в довільній вершині (називатимемо її стартовою вершиною) невідомого йому графа G та одразу фарбує її в чорний колір та додає у множину вершин V_H . Агент рухається по графу з вершини v в вершину u по ребру (v, u) , може змінювати колір вершин v, u , ребер (v, u) та інциденторів $((v, u), v)$, $((v, u), u)$. Знаходячись у вершині (називатимемо її поточною вершиною), мобільний агент може сприймати мітки всіх елементів околу $Q(v)$ і на основі цієї інформації визначає по якому ребру буде далі рухатись і як буде фарбувати елементи графа. Запропонований алгоритм базується на методі обходу графа в глибину [18].

Алгоритм роботи агента.

Природно, що до потрапляння агента на граф всі його елементи пофарбовані в білий колір. Таким чином при потраплянні мобільного агента на невідомий граф, тобто на початку роботи алгоритму чорною стане лише стартова вершина, так як агент фарбує першу вершину в чорний колір та додає її в список вершин одразу після потрапляння на граф. В процесі роботи агента на графі, елементи поступово будуть фарбуватися до того моменту поки весь граф не буде розпізнано, а всі елементи графа не будуть пофарбовані в чорний колір.

Розглянемо детально алгоритм функціонування мобільного агента. Одразу зазначимо, що функціонування агента можливо в двох режимах: звичайний режим роботи та режим розпізнавання зворотних ребер.

Звичайний режим роботи полягає в наступному. Попавши в довільну вершину на початку роботи мобільний агент одразу зафарбовує її в чорний колір і додає в список вершин у свій пам'яті. Далі агент рухається по білим вершинам, зафарбовуючи ці вершини, відвідані ребра і дальні інцидентори в чорний колір. На кожному кроці, агент додає нові вершини та ребра у відповідні списки в свій пам'яті. Якщо на своєму шляху мобільний агент зустрічає біле ребро з білими ближнім та дальніми інциденторами, яке веде в чорну вершину, тобто зустрічає зворотне ребро, то він перемикається в режим розпізнавання зворотних ребер до того моменту, поки не будуть розпізнані всі зворотні ребра з поточної вершини. Така робота продовжується до моменту поки в околі поточної вершини не стане білих вершин. В такому випадку агент починає рух назад, зафарбовуючи дальні інцидентори в чорний колір, в пошуках ще не розпізнаних вершин та ребер. Якщо при русі назад мобільний агент потрапляє в стартову вершину і в її околі також відсутні нерозпізнані вершини та ребра, то це означає, що весь граф розпізнано і мобільний агент завершує роботу. В цей момент в нього в пам'яті вже знаходяться повні списки ребер та вершин досліджуваного графа.

Режим розпізнавання зворотних ребер полягає в наступному. Після зміни режиму роботи мобільний агент фарбує в чорний колір ближні інцидентори всіх зворотних ребер, інцидентних поточній вершині, при цьому веде їх підрахунок. Завершивши фарбування інциденторів, мобільний агент починає рух назад по своєму шляху, ведучи підрахунок зроблених кроків, до виявлення вершини інцидентної позначеному та ще не розпізнаному зворотному ребру. Далі якщо кількість помічених зворотних ребер більше одного, мобільний агент здійснює перехід по цьому ребру, фарбуючи ближній інцидентор в чорний колір та додає це зворотне ребро в список ребер у свій пам'яті паралельно зменшуючи лічильник помічених зворотних ребер на одиницю. На наступному кроці мобільний агент повертається назад по пройденому на попередньому кроці ребру, фарбуючи його в чорний колір і продовжує рух назад в пошуках нерозпізнаних зворотних ребер. В момент, коли буде виявлено останнє зворотне ребро, помічене для розпізнавання (лічильник кількості помічених для розпізнавання ребер дорівнює одиниці) мобільний агент здійснює перехід по цьому ребру, фарбуючи ближній інцидентор та ребро в чорний колір та додає це зворотне ребро в список ребер у свій пам'яті паралельно зменшуючи лічильник помічених зворотних ребер на одиницю. Після чого агент обнуляє лічильник зроблених назад кроків і завершує роботу в режимі розпізнавання зворотних ребер та переключається в звичайний режим роботи. Слід зауважити, що лічильник помічених для розпізнавання зворотних ребер в цей момент вже дорівнює нулю, внаслідок його поступового зменшення при розпізнаванні зворотних ребер.

Алгоритм роботи мобільного агента.

Вхід: зв'язний неорієнтований граф G без петель та кратних ребер, невідомий агенту. Всі елементи графа білого кольору. Мобільний агент розміщений в довільній вершині v графа G .

Вихід: всі елементи графа G чорного кольору. Мобільний агент знаходиться в стартовій вершині v . Сформовано список вершин V_H та список ребер E_H графа H , ізоморфного графу G .

Дані: V_H, E_H списки вершин і ребер графа H . ct – лічильник, що використовується для підрахунку числа відвіданих мобільним агентом вершин графа G . NBE – змінна, що використовується для визначення кількості помічених для розпізнавання зворотних ребер. i – лічильник, що використовується для підрахунку кількості кроків зроблених назад мобільним агентом при розпізнаванні помічених зворотних ребер. $work(1), work(2), \dots, work(t)$ – список номерів вершин робочого шляху, де t – довжина цього списку.

Алгоритм:

1. $ct := 1, i := 0, E_H := \emptyset, t := 1, work(t) := ct, V_H := \{1\}, NBE := 0;$
2. begin
3. $\mu(v) := b;$
4. if $\exists (v, u) \in Q(v) \mid (\mu(v, u) = w) \text{ and } (\mu(u) = \mu(v) = b)$ then do

```

5. EXPL_IE(v);
6. go to 4;
7. end do;
8. else if  $\exists (v, u) \in Q(v) \mid (\mu(v, u) = w) \text{ and } (\mu(u) = w)$  then do
9. FORWARD (v);
10. go to 4;
11. end do;
12. else if  $\exists (v, u) \in Q(v) \mid (\mu((v, u), v) = b) \text{ and } (\mu((v, u), u) = w)$  then do
13. BACK (v);
14. go to 8;
15. end do;
16. else STOP (v);
17. print  $V_H, E_H$ .
18. end.
EXPL_IE(v):
1. while  $\exists (v, u) \in Q(v) \mid (\mu(v, u) = w) \text{ and } (\mu(v) = \mu(u) = b)$  do
2. агент обирає  $(v, u) \in Q(v) \mid (\mu(v, u) = w) \text{ and } (\mu(v) = \mu(u) = b)$  та фарбує  $(\mu(v, u), v) := b$ ;
3.  $NBE := NBE + 1$ ;
4. end do;
5. if  $\exists (v, u) \in Q(v) \mid (\mu((v, u), v) = b) \text{ and } (\mu(v, u) = b) \text{ and } (\mu((v, u), u) = w)$  then do
6. агент обирає  $(v, u) \in Q(v) \mid (\mu((v, u), v) = b) \text{ and } (\mu(v, u) = b) \text{ and } (\mu((v, u), u) = w)$  та переходить по ньому
в вершину  $u$ ;
7.  $v := u$ ;
8.  $i := i + 1$ ;
9. if  $\exists (v, u) \in Q(v) \mid (\mu((v, u), v) = w) \text{ and } (\mu(v, u) = w) \text{ and } (\mu((v, u), u) = b)$  then do
10. if  $NBE > 1$  then do
11. агент обирає  $(v, u) \in Q(v) \mid (\mu((v, u), v) = w) \text{ and } (\mu(v, u) = w) \text{ and } (\mu((v, u), u) = b)$  та переходить по ньому
в вершину  $u$ , фарбуючи  $(\mu((v, u), v) := b)$ ;
12.  $v := u$ ;
13.  $NBE := NBE - 1$ ;
14.  $E_H := E_H \cup \{(work(t), work(t - i))\}$ ;
15. агент обирає  $(v, u) \in Q(v) \mid (\mu((v, u), v) = b) \text{ and } (\mu(v, u) = w) \text{ and } (\mu((v, u), u) = b)$  та переходить по ньому
в вершину  $u$ , фарбуючи  $(\mu(v, u) := b)$ ;
16. go to 5 даної процедури;
17. end do;
18. else do
19. агент обирає  $(v, u) \in Q(v) \mid (\mu((v, u), v) = w) \text{ and } (\mu(v, u) = w) \text{ and } (\mu((v, u), u) = b)$  та переходить по ньому
в вершину  $u$ , фарбуючи  $(\mu((v, u), v) := b)$  та  $(\mu(v, u) := b)$ ;
20.  $v := u$ ;
21.  $NBE := NBE - 1$ ;
22.  $E_H := E_H \cup \{(work(t), work(t - i))\}$ ;
23.  $i := 0$ ;
24. end do;
25. end do;
26. end do;
FORWARD(v):
1. Агент обирає  $(v, u) \in Q(v) \mid (\mu(v, u) = w) \text{ and } (\mu(u) = w)$  та переходить по ньому в вершину  $u$ , фарбуючи
 $\mu(v, u) := b, \mu((v, u), u) := b, \mu(u) := b$ ;
2.  $v := u$ ;
3.  $ct := ct + 1$ ;
4.  $t := t + 1$ ;
5.  $work(t) := ct$ ;
6.  $V_H := V_H \cup (ct)$ ;
7.  $E_H := E_H \cup \{(work(t - 1), work(t))\}$ .
BACK(v):
1. Агент обирає  $(v, u) \in Q(v) \mid (\mu((v, u), v) = b) \text{ and } (\mu((v, u), u) = w)$  та переходить по ньому в вершину  $u$ , фар-
буючи  $\mu((v, u), u) := b$ ;

```

2. $v := u$;
3. delete $work(t)$;
4. $t := t - 1$.

STOP(v):

1. При виконанні процедури STOP(v), агент фарбує.

Властивості алгоритму розпізнавання.

Лема. Мобільний агент, виконавши алгоритм розпізнавання графа, однозначно розпізнає всі зворотні ребра.

Доведення. В запропонованому алгоритмі за розпізнавання зворотних ребер відповідає процедура EXPL_IE(v). Якщо в процесі руху вперед агент помічає зворотні ребра з поточної вершини, він мітить кожне зворотне ребро фарбуванням ближнього інцидентора в чорний колір паралельно підраховуючи їх кількість (рядки 1–4 процедури EXPL_IE(v)). Далі агент починає рух у зворотному напрямі, рахуючи кількість зроблених кроків (рядки 5–8 процедури EXPL_IE(v)). Якщо в якійсь вершині агент помічає біле ребро, дальній інцидентор якого пофарбований в чорний колір і це не останнє нерозпізнане зворотне ребро, то він переходить по цьому ребру, зафарбовуючи ближній інцидентор в чорний колір (рядок 11 процедури EXPL_IE(v)) та визначає раніше невідомий неявний номер другої вершини зворотного ребра. А так як агенту відомий неявний номер вершини, з якої помічалися зворотні ребра, то ми додаємо зворотне ребро у множину ребер графа H . Потім агент повертається назад по зворотному ребру, фарбуючи його в чорний колір (рядок 15 процедури EXPL_IE(v)) та продовжує рух назад в пошуках нерозпізнаних зворотних ребер. Так продовжується до моменту поки не знайдеться останнє нерозпізнане зворотне ребро з даної вершини (рядок 18 процедури EXPL_IE(v)). В цьому випадку агент переходить по зворотному ребру фарбуючи ближній інцидентор і саме ребро в чорний колір, додаючи ребро в список ребер. Після чого обнуляє лічильник числа назад і перемикається в звичайний режим роботи.

Таким чином, мобільний агент для кожного зворотного ребра, інцидентного вершині з якої помічено зворотні ребра для розпізнавання, підраховує кількість переходів по вершинах робочого шляху, які необхідно здійснити для того, щоб дістатися до другої вершини інцидентної конкретному зворотному ребру та однозначно визначити її номер, що в свою чергу дозволяє однозначно визначити зворотне ребро та додати його у множину ребер E_H графа H . Що й треба було довести.

Теорема 1. Мобільний агент, виконавши алгоритм розпізнавання графа, розпізнає будь-який граф з точністю до ізоморфізму.

Доведення. Розглянемо детально процедури додавання вершин в список V_H . Процедура FORWARD(v) виконується при відвідуванні білих вершин досліджуваного графа G . Ця процедура за одне виконання створює в пам'яті агента одну нову вершину графа H та додає її до списку V_H . Таким чином, при виконанні алгоритму створюється неявна нумерація $\varphi: V_G \rightarrow V_H$, де рівність $\varphi(v) = ct$ встановлюється, коли вершина v фарбується в чорний колір. Ця нумерація є бієкцією, оскільки в зв'язному графі всі вершини досяжні, а отже будуть відвідані агентом-дослідником.

Так як запропонований алгоритм розпізнавання базується на методі обходу графа в глибину, то всі ребра досліджуваного графа G поділяються на ребра, що належать дереву та зворотні ребра. При виконанні процедури FORWARD(v) мобільний агент 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 . При виконанні процедури EXPL_IE(v), агент розпізнає зворотні ребра (v, u) 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 . Теорему доведено.

Підрахуємо часову та емісню складності запропонованого алгоритму в рівномірній шкалі [19]. Також проведемо оцінку кількості переходів мобільного агента по ребрах досліджуваного графа, які йому необхідно здійснити для повного розпізнавання цього графа. З опису алгоритму видно, що на кожному кроці алгоритму робочий шлях – це простий шлях, що з'єднує стартову вершину v з номером $\varphi(v) = 1$ з вершиною u з номером $\varphi(u) = ct$. Таким чином можна зробити висновок, що довжина робочого шляху не перевищує n . Зазначимо, що на одноразове просування вперед чи назад (процедури FORWARD(v), BACK(v)) агент витрачає один крок та проходить одне ребро. При розпізнаванні всіх зворотних ребер з однієї вершини (процедура EXPL_IE(v)) агент проходить не більше $n - 2$ зворотних ребер і не більше $n - 1$ ребер робочого шляху. Розпізнаючи зворотні ребра, агент проходить цикл, що складається з одного ребра і деякого скінченного відрізка робочого шляху, що з'єднує вершини, інцидентні зворотному ребру.

При підрахунку часової складності алгоритму будемо вважати, що ініціалізація алгоритму та вибір однієї з можливих процедур займають деяку постійну кількість одиниць часу. Також вважатимемо, що перехід агента-дослідника з однієї вершини в іншу займає час, що дорівнює деякій константі. Очевидно, що загальний час аналізу околу $Q(v)$ робочої вершини v та вибір необхідних ребер оцінюється зверху як $O(n^2)$. Враховуючи сказане, співвідношення, що визначають часову складність алгоритму виглядають наступним чином:

1. Ініціалізація алгоритму виконується один раз і її асимптотична складність дорівнює $O(1)$.

2. Процедура FORWARD(v) виконується не більше $n - 1$ раз і загальний час її виконання оцінюється зверху як $O(n)$.

3. Процедура *BACK*(v) виконується не більше ніж $n - 1$ раз та загальний час її виконання оцінюється як $O(n)$.
 4. Процедура *EXPL_IE*(v) виконується не більше $n - 2$ раз, кожен з яких займає не більше $(n - 1) + 2 \cdot (n - 2)$ часу, тобто загальний час виконання процедури оцінюється як $((n - 1) + 2 \cdot (n - 2)) \cdot (n - 2)$, тобто $O(n^2)$.

5. 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 $(n - 1) + (n - 1) + (((n - 1) + 2 \cdot (n - 2)) \cdot (n - 2))$, тобто верхня оцінка числа переходів по ребрах, які необхідно виконати мобільному агенту для повного розпізнавання досліджуваного графа задовольняє співвідношенню: $M(n) = O(n^2)$.

Загальна часова складність запропонованого алгоритму розпізнавання графів задовольняє співвідношення: $T(n) = O(n^2)$.

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

Таким чином має місце наступна теорема.

Теорема. Часова й ємнісна складність алгоритму розпізнавання дорівнюють $O(n^2)$, верхня оцінка числа переходів по ребрах, що здійснює мобільний агент оцінюється як $O(n^2)$. При цьому алгоритм використовує одну фарбу.

Висновки

В роботі запропоновано новий алгоритм розпізнавання зв'язних скінчених неорієнтованих графів без петель та кратних ребер, що має квадратичні часову й ємнісну складності. Верхня оцінка числа переходів по ребрах, які необхідно виконати мобільному агенту для повного розпізнавання досліджуваного графа – $O(n^2)$. Мобільний агент має скінчену на кожному кроці, необмежено зростаючу внутрішню пам'ять, яка залежить від розмірності досліджуваного графа.

Список використаної літератури

1. Goth G. *Exploring new frontiers*. Communications of the ACM. 2009. 52(11). P. 21–23.
2. 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>
3. Dongyu Li, Shuzhi Sam Ge, Wei He, Guangfu Ma, Lihua Xie. Multilayer formation control of multi-agent systems. *Automatica*. V 109. 2019 <https://doi.org/10.1016/j.automatica.2019.108558>
4. Zhang T, Ma X, Li H, Wang Z, Xie S, Luo J. Ordered-Bipartite Consensus of Multi-Agent Systems under Finite Time Control. *Applied Sciences*. 2022; 12(23):12337. <https://doi.org/10.3390/app122312337>
5. Стюпкін 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>
6. Стюпкін А.В. Мультиагентна система розпізнавання неорієнтованих графів. *Інформаційні технології та суспільство*. Київ, 2024. Вип. 3(14). 38–43 с.
7. Stopkin A.V. Finite graph exploration by a mobile agent. *Mathematical Modeling and Computing*. 2025 Vol. 12, No. 1, pp. 75–82.
8. Shannon C.E. *Presentation of a maze-solving machine*. Cybernetics Trans, of the 8th Conf. of the JosiahMacy Jr. Found / Editor: H.Foerster. 1951. pp. 173–180.
9. Dopp K. *Automaten in labirinth*. Elektronische Informationsverarbeitung und Kybernetik. 1971. V. 7, № 2. pp. 79–94.
10. Dudek G., Jenkin M., Milios E., Wilkes D. *Map validation in a graphlike world*. Proceedings of the 13th International Joint Conference on Artificial Intelligence (Chambery, France, August 1993). San Fransisco: Morgan Kaufmann Publishers Inc., 1993. pp. 1648–1653.
11. Сапунов С.В., Сенченко А.С. Лингвистическое представление графов с помеченными вершинами. *Доповіді Національної академії наук України*, 2019. № 11. С. 17–24.
12. Dudek G., Jenkin M., Milios E., Wilkes D. *A taxonomy for swarm robots*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Yokohama, Japan, 1993. pp. 441–447.
13. Dudek G., Jenkin M., Milios E., Wilkes D. *Topological exploration with multiple robots*. Proc. 7th International Symposium on Robotics with Application (ISORA). Anchorage, Alaska, USA, 1998.
14. Грунский И.С., Татаринев Е.А. Распознавание конечного графа блуждающим по нему агентом. *Вестник ДонНУ*, 2009, Вып.1. С. 492–497.
15. Грунский И.С., Стёпкин А.В. Распознавание конечного графа коллективом агентов. *Труды ИПММ НАН Украины*, 2009. Т.19. С. 43–52.
16. Стёпкин А.В. Распознавание конечных графов тремя агентами / А.В.Стёпкин // Искусственный интеллект. 2011. № 2. С. 84–93.

17. Stepkin A. Using a Collective of Agents for Exploration of Undirected Graphs. *Cybernetics and Systems Analysis*. 2015. V.51, № 2. pp. 223–233. <https://doi.org/10.1007/s10559-015-9715-z>
18. Cormen T., Leiserson Ch., Rivest R., Stein C. *Introduction to Algorithms*. Cambridge, 2009. 1292 p.
19. Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1974). *The Design and Analysis of Computer Algorithms*. Reading, Mass.: Addison-Wesley.

References

1. Goth G. (2009). Exploring new frontiers. *Communications of the ACM*, 52(11), 21–23.
2. 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>
3. 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>
4. Zhang T, Ma X, Li H, Wang Z, Xie S, Luo J. (2022). Ordered-Bipartite Consensus of Multi-Agent Systems under Finite Time Control. *Applied Sciences*, 12(23):12337. <https://doi.org/10.3390/app122312337>
5. 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>
6. Stepkin, A. V. (2024). Multi-agent system for recognizing undirected graphs. *Information Technologies and Society*, 3(14), 38–43.
7. Stopkin, A. V. (2025). Finite graph exploration by a mobile agent. *Mathematical Modeling and Computing*, 12(1), 75–82.
8. Shannon, C. E. (1951). Presentation of a maze-solving machine. In H. Foerster (Ed.), *Cybernetics Trans, of the 8th Conf. of the Josiah Macy Jr. Foundation*. 173–180.
9. Dopp, K. (1971). Automaten in Labyrinthen. *Elektronische Informationsverarbeitung und Kybernetik*, 7(2), 79–94.
10. Dudek, G., Jenkin, M., Milios, E., & Wilkes, D. (1993). Map validation in a graph-like world. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence* (Chambery, France, August 1993). San Francisco: Morgan Kaufmann Publishers Inc., 1648–1653.
11. Sapunov, S. V., & Senchenko, A. S. (2019). Linguistic representation of graphs with labeled nodes. *Reports of the National Academy of Sciences of Ukraine*, 11, 17–24.
12. Dudek, G., Jenkin, M., Milios, E., & Wilkes, D. (1993). A taxonomy for swarm robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Yokohama, Japan, 441–447.
13. Dudek, G., Jenkin, M., Milios, E., & Wilkes, D. (1998). Topological exploration with multiple robots. In *Proceedings of the 7th International Symposium on Robotics with Application (ISORA)*, Anchorage, Alaska, USA.
14. Grunsky, I. S., & Tatarinov, E. A. (2009). Recognition of a finite graph by a wandering agent. *Bulletin of Donetsk National University*, 1, 492–497.
15. Grunsky, I. S., & Stepkin, A. V. (2009). Exploration of a finite graph by a team of agents. *Proceedings of the Institute of Applied Mathematics and Mechanics of the National Academy of Sciences of Ukraine*, 19, 43–52.
16. Stepkin, A. V. (2011). Exploration of finite graphs by three agents. *Artificial Intelligence*, 2, 84–93.
17. Stepkin, A. (2015). Using a collective of agents for exploration of undirected graphs. *Cybernetics and Systems Analysis*, 51(2), 223–233. <https://doi.org/10.1007/s10559-015-9715-z>
18. Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2009). *Introduction to algorithms*. Cambridge: MIT Press, 1292 p.
19. Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1974). *The design and analysis of computer algorithms*. Reading, Mass.: Addison-Wesley.