

Стьопкін А.В.

ДВНЗ «Донбаський державний педагогічний університет»

РОЗПІЗНАВАННЯ ПРОСТИХ ГРАФІВ КОЛЕКТИВОМ АГЕНТІВ

В роботі досліджується проблема розпізнавання скінчених зв'язних неорієнтованих графів без петель та кратних ребер колективом агентів. Метою роботи є побудова нового ефективного алгоритму розпізнавання графів колективом агентів. В статті запропоновано наступну методологію до досягнення поставленої мети: використати колектив агентів, що складається з двох агентів двох різних типів. Перший тип – це агент-дослідник, який може рухатись по графу, обмінюючись повідомленнями з другим агентом, а також може зчитувати та змінювати мітки на елементах графа. Другий тип агентів – це агент-експериментатор – нерухомий агент, що знаходиться поза межами досліджуваного графу, в пам'яті якого на кожному кроці фіксується результат функціонування агента-дослідника. На основі отриманої інформації в пам'яті агента-експериментатора поступово вибудовується представлення досліджуваного графа списком ребер і списком вершин. Агенти мають скінчену на кожному кроці, але необмежено зростаючу внутрішню пам'ять. В роботі детально розглянуто режими функціонування агента-дослідника із зазначенням пріоритетності активації цих режимів в процесі дослідження графа. Також розглянуто алгоритм функціонування агента-експериментатора з детальним описом процедури обробки отриманих повідомлень, за допомогою яких і відбувається розпізнавання досліджуваного графа. Також в роботі проаналізовано часову, ємнісну, комунікаційну складності побудованого алгоритму та кількість переходів по ребрах, які необхідно виконати агенту-досліднику для повного розпізнавання досліджуваного графа. Науковою новизною є отримання ефективного методу та алгоритму розпізнавання скінчених неорієнтованих графів без петель та кратних ребер, який дозволяє використовувати для розпізнавання лише одну фарбу та дає можливість в подальшому масштабувати розглянутий колектив до k агентів. Таким чином, в роботі запропоновано новий метод та алгоритм розпізнавання графів, який має квадратичні (від числа вершин досліджуваного графа) часову та ємнісну складності, комунікаційна складність алгоритму оцінюється зверху як $O(n^2 \times \log(n))$, де n – кількість вершин досліджуваного графа. Кількість переходів по ребрах, які необхідно виконати агенту-досліднику для повного розпізнавання досліджуваного графа оцінюється зверху як $O(n^2)$. Робота запропонованого алгоритму розпізнавання ґрунтується на методі обходу графа в глибину.

Ключові слова: розпізнавання графів, неорієнтовані графи складність алгоритму, обхід в глибину, колектив агентів.

Постановка проблеми. В наш час однією з важливих проблем комп'ютерних наук, вирішенню якої приділяється особлива увага є проблема взаємодії керуючої та керованої систем. Прикладом такої взаємодії є взаємодія керуючого автомата, його агента та операційного середовища даного агента. Наприклад, в роботах [1,2] взаємодія таких систем представлена як процес переміщення агентів по графу. Зрозуміло, що якщо відсутня повна модель операційного середовища, то є неможливим цілеспрямований рух по ній мобільних агентів. В питанні моделювання операційного середовища, визначено ряд підходів, одним з яких є топологічний [3]. При такому підході агенту, що рухається графом, доступна тільки інформація щодо зв'язків між різноманітними областями середовища та недоступна метрична інформація про дане середовище. Тобто топологічна модель представляє

собою граф з відміченими вершинами, ребрами та інциденторами.

Серед задач по дослідженню середовищ прийнято виділяти три основні задачі: задача самолокалізації агента; задача контролю відповідності моделі середовища самому середовищу; задача побудови мапи середовища. Вирішенню задачі побудови мапи середовища мобільними агентами присвячено досить велику кількість робіт [4-8]. Відомий ряд алгоритмів розпізнавання з використанням різних інструментів мобільних агентів, різноманітних колективів агентів чи робота агентів за наявною апріорною інформацією про досліджуваний граф, тощо. Дана робота присвячена створенню нового ефективного методу та побудові відповідного алгоритму розпізнавання скінчених неорієнтованих простих графів за допомогою колективу агентів, а також дослідженню часової, ємнісної й комунікаційної складностей

побудованого алгоритму та кількості переходів по ребрах, які виконують агенти-дослідники для повного розпізнавання досліджуваного графа.

Аналіз останніх досліджень і публікацій. Задача розпізнавання невідомих середовищ залишається актуальною і в наш час, про що свідчать сучасні дослідження в цьому напрямку [2,9]. Але в наш час більш детально вивчається робота різноманітних мультиагентних систем [10-15] над розпізнаванням невідомого середовища та вивчаються проблеми, які виникають при роботі таких систем. В залежності від поставлених цілей, проблеми керування мультиагентними системами можна умовно розділити на такі види: проблема керування мультиагентною системою [16], проблема формування мультиагентної системи [17] та проблема консенсусу мультиагентної системи [18].

Постановка завдання. Якщо проаналізувати останні дослідження, то можна виявити, що в них замало уваги приділяється дослідженню процесу обміну інформацією між агентами. Це робить актуальним проведення експериментів по розпізнаванню графів колективом агентів, а також завдання, спрямовані на оптимізацію витрат ресурсів, часу, та навантаження на канал зв'язку між агентами.

Таким чином, метою нашого дослідження є створення ефективного алгоритму розпізнавання скінчених зв'язних простих неорієнтованих графів колективом агентів, ефективність якого або перевищить раніше розроблені алгоритми, або зменшить кількість використовуваних в них ресурсів [2,8,10,11]. Тобто дослідження часової, емпіричної й комунікаційної складностей побудованого алгоритму та кількості переходів по ребрах, які необхідно виконати агенту-досліднику для розпізнавання графа також є важливою частиною нашого дослідження.

Виклад основного матеріалу.

В роботі розглядаються зв'язні неорієнтовані прості скінчені графи $G = (V, E)$, де V – множина вершин графа, а E – множина ребер (двоелементних підмножин (u, v) , де $u, v \in V$) графа. Трійки $((u, v), v)$ називатимемо інциденторами (точкою з'єднання) ребра (u, v) та вершини v . Через I позначимо множину всіх інциденторів графа. Множину $L = V \cup E \cup I$ назвемо множиною всіх елементів графа G . Сюр'єктивне відображення $\mu: L \rightarrow \{w, b\}$, де w – білий колір, а b – чорний, назвемо функцією розфарбування графа G . Пару (G, μ) будемо називати розфарбованим графом. Послідовність $u_1, u_2, \dots, u_k$ попарно суміжних вершин графа G назвемо шляхом довжини k . Околом

$Q(v)$ вершини v будемо називати множину елементів графа, яка складається з вершини v , всіх вершин u , суміжних з v , всіх ребер (v, u) та всіх інциденторів $((v, u), v)$ та $((v, u), u)$. Потужності множини вершин V та множини ребер E позначимо через n та m відповідно. Зрозуміло, що $m \leq \frac{n(n-1)}{2}$. Ізоморфізмом графа G та графа H назвемо таку бієкцію $\phi: V_G \rightarrow V_H$, що $(v, u) \in E_G$ тоді й тільки тоді, коли $(\phi(v), \phi(u)) \in E_H$.

Агент-дослідник характеризується наступними властивостями. Він може рухатися по графу з вершини v у вершину u по ребру (v, u) , що з'єднує ці вершини. При цьому він може фарбувати вершини, ребра та інцидентори, через які здійснив перехід. Коли агент-дослідник знаходиться у вершині v , він сприймає мітки всіх елементів околу $Q(v)$ і на основі цієї інформації визначає в якому режимі буде далі працювати. Агент-дослідник має скінчену на кожному кроці та необмежено зростаючу внутрішню пам'ять, яка залежить від розмірності досліджуваного графа.

Агент-експериментатор характеризується наступними властивостями. Це нерухомий агент, що знаходиться за межами графа. Він приймає та ідентифікує повідомлення від агента-дослідника й на основі цих повідомлень будує представлення досліджуваного графа в своїй пам'яті у вигляді списків ребер та списків вершин. Агент-експериментатор має скінчену на кожному кроці та необмежено зростаючу внутрішню пам'ять, яка залежить від розмірності досліджуваного графа.

Стратегія роботи агента-дослідника полягає в наступному: агент-дослідник рухається вглибину, поки це можливо, після чого повертається назад по своєму шляху в пошуках ще не відвіданих вершин та ребер, якщо він знаходить такі вершини, то знову починає рух вглибину і так відбувається поки агент не повернеться в стартову вершину, в околі якої вже не буде нерозпізнаних вершин та ребер. Після цього агент-дослідник закінчує роботу.

Таким чином, в процесі роботи агент-дослідник будує неявне дерево пошуку вглибину та відносно цього дерева всі ребра поділяються на деревні, тобто такі, що належать побудованому дереву та зворотні, які не належать дереву.

Розберемо більш детально алгоритми роботи агентів. Вся процедура розпізнавання графа складається з двох принципово різних типів алгоритмів: алгоритму обходу досліджуваного графа та обчислювального алгоритму побудови представлення графа у вигляді списків ребер та вершин. Зазначимо, що спільна робота цих алгоритмів

відбувається в наступній послідовності: агент-дослідник виконує крок та відправляє повідомлення агенту-експериментатору, після чого агент-експериментатор оброблює повідомлення, агент-дослідник робить крок і т.д.

До початку роботи агентів всі елементи графа білого кольору. Під час роботи по розпізнаванню графа, його елементи можуть бути пофарбовані мобільним агентом в чорний колір. Для підвищення зручності читання алгоритму, при описі дій, які виконуються агентом-дослідником ми будемо зазначати в дужках повідомлення, які він відправляє агенту-експериментатору у форматі «Message_A».

На початку роботи алгоритму розпізнавання графа, агент-дослідник розміщується в довільній вершині досліджуваного графа та фарбує її в чорний колір, а агент-експериментатор одразу додає першу вершину в список вершин у своїй пам'яті. Далі агент-дослідник рухається по білим вершинам, фарбуючи ці вершини, ребра та дальні інцидентори в чорний колір та покроково відправляючи агенту-експериментатору повідомлення «Forward_A», на основі яких він додає ці вершини і ребра до множин V і E відповідно.

Якщо під час руху вперед агент-дослідник зустрів зворотне ребро (біле ребро з білими ближнім та дальнім інциденторами, яке веде в чорну вершину), то агент-дослідник покроково фарбує в чорний колір ближні інцидентори всіх зворотних ребер, інцидентних поточній вершині, при цьому відправляє агенту-експериментатору повідомлення «Marked_IE», завдяки якому агент-експериментатор буде вести підрахунок помічених зворотних ребер. Завершивши фарбування інциденторів, агент-дослідник починає рух назад по своєму шляху, покроково відправляючи агенту-експериментатору повідомлення «Back_BE» завдяки якому агент-експериментатор буде вести підрахунок зроблених агентом-дослідником кроків назад. Такий рух назад продовжується до виявлення агентом-дослідником вершини інцидентної поміченому та ще не розпізаному зворотному ребру (біле ребро, що веде в чорну вершину з чорним дальнім інцидентором). Коли таке ребро буде виявлено, агент-дослідник запитує в агента-експериментатора кількість помічених для розпізнавання зворотних ребер, і якщо вона більше одного, то агент-дослідник здійснює перехід по цьому ребру, фарбуючи ближній інцидентор в чорний колір та відправляє агенту-експериментатору повідомлення «Forward_BE», завдяки якому агент-експериментатор додасть

в список ребер у своїй пам'яті це зворотне ребро, а лічильник помічених зворотних ребер зменшить на одиницю. На наступному кроці агент-дослідник 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 помічене для розпізнавання зворотне ребро (лічильник кількості помічених для розпізнавання зворотних ребер дорівнює одиниці) агент-дослідник здійснить перехід по цьому ребру, фарбуючи ближній інцидентор та ребро в чорний колір та відправить агенту-експериментатору повідомлення «Forward_BE». Після чого відправляє агенту-експериментатору повідомлення «Recognized_BE», завдяки якому агент-експериментатор обнуляє лічильник зроблених назад кроків і завершує роботу по розпізнаванню зворотних ребер. Слід зауважити, що лічильник помічених для розпізнавання зворотних ребер в цей момент вже дорівнює нулю, внаслідок його поступового зменшення при розпізнаванні зворотних ребер.

Після закінчення роботи з розпізнавання зворотних ребер агент-дослідник продовжує рух вперед по білим вершинам. У випадку коли при русі вперед агент-дослідник потрапляє у вершину, в околі якої відсутні зворотні ребра та нерозпізані вершини, він починає рух назад по своєму шляху в пошуках нерозпізнаних вершин, фарбуючи при цьому в чорний колір дальні інцидентори ребер. На кожному кроці при цьому агенту-експериментатору відправляється повідомлення «Back_A». При знаходженні нерозпізнаних вершин, агент-дослідник знову починає рух вперед для розпізнавання невідвіданих вершин та ребер. Якщо ж при русі назад, агент-дослідник потрапляє в стартову вершину, а в її околі відсутні нерозпізані вершини, то це означає, що весь граф розпізнано і агент-дослідник завершує свою роботу, відправивши агенту-експериментатору повідомлення «Stop». В цей момент в пам'яті агента експериментатора вже сформовано представлення досліджуваного графа у вигляді списку ребер та вершин.

Алгоритм роботи агента-дослідника:

1. $\mu(v) := b$;
2. *if* $\exists (v,u) \in Q(v) \mid (\mu(v,u) = w) \text{ and } (\mu(u) = \mu(v) = b)$ *then do*
3. $EXPL_IE(v)$;
4. *else if* $\exists (v,u) \in Q(v) \mid (\mu(v,u) = w) \text{ and } (\mu(u) = w)$ *then do*
5. $FORWARD(v)$;

```

6. go to 2;
7. end do;
8. else if  $\exists (v,u) \in Q(v) | (\mu((v,u),v)=b) \text{ and } (\mu((v,u),u)=w)$  then do
9. BACK(v);
10. go to 3;
11. end do;
12. else STOP(v);
EXPL IE(v):
1. while  $\exists (v,u) \in Q(v) | (\mu(v,u)=w) \text{ and } (\mu(v)=\mu(u)=b)$  do
2. агент обирає  $(v,u) \in Q(v) | (\mu(v,u)=w) \text{ and } (\mu(v)=\mu(u)=b)$  та фарбує  $(\mu((v,u),v)) := b$ ;
3. агент записує в список LM повідомлення:
Marked IE;
4. end do;
5. if  $\exists (v,u) \in Q(v) | (\mu((v,u),v)=b) \text{ and } (\mu(v,u)=b) \text{ and } (\mu((v,u),u)=w)$  do
6. агент обирає  $(v,u) \in Q(v) | (\mu((v,u),v)=r) \text{ and } (\mu(v,u)=r) \text{ and } (\mu((v,u),u)=w)$  та переходить по ньому в вершину u;
7.  $v := u$ ;
8. агент записує в список LM повідомлення:
Back_BE;
9. end do;
10. if  $\exists (v,u) \in Q(v) | (\mu((v,u),v)=w) \text{ and } (\mu(v,u)=w) \text{ and } (\mu((v,u),u)=b)$  then do
11. запит NBE;
12. if  $NBE > 1$  then do
13. агент обирає  $(v,u) \in Q(v) | (\mu((v,u),v)=w) \text{ and } (\mu(v,u)=w) \text{ and } (\mu((v,u),u)=b)$  та переходить по ньому в вершину u, фарбуючи  $(\mu((v,u),v)) := b$ ;
14.  $v := u$ ;
15. агент записує в список LM повідомлення:
Forward_BE;
16. агент обирає  $(v,u) \in Q(v) | (\mu((v,u),v)=b) \text{ and } (\mu(v,u)=w) \text{ and } (\mu((v,u),u)=b)$  та переходить по ньому в вершину u, фарбуючи  $(\mu(v,u)) := b$ ;
17. go to 5 даної процедури;
18. end do;
19. else do
20. агент обирає  $(v,u) \in Q(v) | (\mu((v,u),v)=w) \text{ and } (\mu(v,u)=w) \text{ and } (\mu((v,u),u)=b)$  та переходить по ньому в вершину u, фарбуючи  $(\mu((v,u),v)) := b$  та  $(\mu(v,u)) := b$ ;
21.  $v := u$ ;
22. агент записує в список LM повідомлення:
Forward_BE;
23. агент записує в список LM повідомлення:
Recognized_BE;
24. end do;
25. end do;

```

FORWARD(v):

1. Агент обирає $(v,u) \in Q(v) | (\mu(v,u)=w) \text{ and } (\mu(u)=w)$ та переходить по ньому в вершину u, фарбуючи $\mu(v,u) := b, \mu((v,u),u) := b, \mu(u) := b$;

2. $v := u$;

3. агент записує в LM повідомлення: Forward; BACK(v):

1. Агент обирає $(v,u) \in Q(v) | (\mu((v,u),v)=b) \text{ and } (\mu((v,u),u)=w)$ 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 LM від агента-дослідника.

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

Дані: V_H, E_H списки вершин і ребер графа H. ct – лічильник числа відвіданих агентом-дослідником вершин графа G. i – лічильник, що використовується при визначенні номерів других вершин помічених зворотних ребер. STOP – змінна, що використовується агентом-дослідником для повідомлення агента-експериментатора, про завершення розпізнавання графа. $work(1), work(2), \dots, work(t)$ – список номерів вершин робочого шляху, де t – довжина цього списку. NBE – змінна, що використовується для підрахунку кількості помічених для розпізнавання зворотних ребер. Mes – повідомлення, яке оброблюється в даний момент.

1. $LM := \emptyset$; $ct := 1$, $i := 0$, $E_H := \emptyset$, $t := 1$, $work(t) := ct$, $V_H := \{1\}$, STOP := 0;

3. while $LM \neq \emptyset$ do

4. if $LM \neq \emptyset$ then do

5. читаємо повідомлення в Mes;

6. $LM := LM \setminus \{Mes\}$;

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 = "Marked IE" then Marked_IE();

4. if Mes = "Forward_BE" then Forward_BE();

5. if Mes = "Back_BE" then Back_BE();

6. if Mes = "Recognized_BE" then Recognized_BE();

7. if Mes = "Stop" then Stop().

$Forward(): \quad ct := ct + 1; \quad t := t + 1;$
 $work(t) := ct; \quad V_H := V_H \cup \{ct\};$
 $E_H := E_H \cup \{(work(t-1), work(t))\};$
 $Back(): \quad delete \ work(t); \quad t := t - 1;$
 $Stop(): \quad STOP := 1;$
 $Marked_IE: \quad NBE := NBE + 1; \quad Back_BE: \quad i := i + 1;$
 $Forward_BE: \quad NBE := NBE - 1;$
 $E_H := E_H \cup \{(work(t), work(t-i))\};$
 $Recognized_BE: \quad i := 0;$

Розглянемо більш детально властивості алгоритму розпізнавання. При виконанні зазначеного алгоритму, 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 одноразового виконання цієї процедури, в пам'ять агента-експериментатора додається одна нова вершина графа H . Таким чином, при виконанні алгоритму створюється неявна нумерація $\phi: V_G \rightarrow V_H$, де рівність $\phi(v) = ct$ встановлюється, коли вершина v фарбується в чорний колір. Зазначена нумерація є бієкцією, оскільки в зв'язному графі G всі вершини досяжні зі стартової вершини, а отже будуть відвідані агентом-дослідником. Оскільки алгоритм базується на методі обходу графа в глибину, то всі ребра досліджуваного графа поділяються на ребра, що належать дереву та зворотні ребра. При виконанні процедури $FORWARD(v)$ агент-дослідник також розпізнає одне ребро дерева (v, u) і так нумерує вершину u , що ребру (v, u) графа G однозначно відповідає ребро $(\phi(v), \phi(u))$ графа H . При виконанні процедури $EXPL_IE(v)$, агент розпізнає зворотні ребра (v, u) графа G та ставить їм в однозначну відповідність ребра $(\phi(v), \phi(u))$ графа H . Таким чином, відображення ϕ є ізоморфізмом графа G на граф H . Враховуючи сказане має місце наступна теорема.

Теорема. Виконавши алгоритм розпізнавання на графі G , агент розпізнає досліджуваний граф з точністю до ізоморфізму.

Підрахуємо часову, смісну та комунікаційну складності запропонованого алгоритму. Також визначимо верхню оцінку числа переходів по ребрах, які необхідно здійснити агенту-досліднику для повного розпізнавання графа G .

При підрахунку часової складності побудованого алгоритму розпізнавання будемо вважати, що перехід агента-дослідника з однієї вершини в іншу займає час, що дорівнює деякій константі. Очевидно, що загальний час аналізу околу робочої вершини $Q(v)$ та вибір необхідних ребер оцінюється зверху як $O(n^2)$. Також вважатимемо,

що обробка одного повідомлення агентом-експериментатором відбувається не довше переходу агента-дослідника з однієї вершини в іншу.

Під час опису алгоритму, ми зазначали, що у агента-дослідника є робочий шлях – це шлях від стартової вершини агента-дослідника до його поточної робочої вершини. Тобто на кожному кроці алгоритму робочий шлях – це простий шлях, від стартової вершини v з номером $\phi(v) = 1$ до вершини u з номером $\phi(u) = ct$. Таким чином, можна зробити висновок, що довжина робочого шляху не перевищує n . На виконання процедур $FORWARD(v)$ та $BACK(v)$ агент-дослідник витрачає один крок, проходячи при цьому одне ребро. Виконуючи процедуру $EXPL_IE(v)$ по розпізнаванню зворотних ребер з однієї вершини, агент-дослідник проходить назад не більше $n-1$ ребер робочого шляху та розпізнає не більше $n-2$ ребер.

Враховуючи сказане, співвідношення, що визначають часову складність алгоритму будуть виглядати 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 $(n-1) + 2 \times (n-2)$ часу, тобто загальний час виконання процедури оцінюється як $((n-1) + 2 \times (n-2)) \times (n-2)$, 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 $(n-1) + (n-1) + (((n-1) + 2 \times (n-2)) \times (n-2))$, тобто верхня оцінка числа переходів задовольняє співвідношенню: $M(n) = O(n^2)$.

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

Смісна 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 $S(n) = O(n^2)$.

На кожному кроці алгоритму агент-дослідник може відправити агенту-експериментатору не більше одного повідомлення, і так як повідомлення не містить ніякої інформації про елементи графа, то його об'єм можна вважати рівним кон-

станті. В той же час агент-дослідник при розпізнаванні зворотних ребер запитує кількість помічених для розпізнавання ребер. Тобто повідомлення будуть містити змінну, розмір якої залежить від розмірності графа. Таким чином, об'єм переданої інформації оцінюється як $O(n^2 \times \log(n))$. Отже, $K(n) = O(n^2 \times \log(n))$. Враховуючи сказане, має місце наступна теорема.

Теорема. Часова та ємнісна складності алгоритму розпізнавання дорівнюють $O(n^2)$, комунікаційна складність складає $O(n^2 \times \log(n))$ верхня оцінка числа переходів по ребрах, що здійснює агент-дослідник оцінюється як $O(n^2)$. При цьому алгоритм використовує одну фарбу.

Висновки. В дослідженні розглядається задача розпізнавання графів колективом агентів. Запропоновано новий алгоритм розпізнавання простих неорієнтованих скінчених графів, квадратичної часової та ємнісної складностей, комунікаційна складність алгоритму при цьому складає $O(n^2 \times \log(n))$, а верхня оцінка числа переходів по ребрах, що здійснюється агентом-дослідником дорівнює $O(n^2)$. Агенти мають скінчену на кожному кроці та необмежено зростаючу внутрішню пам'ять, об'єм якої залежить від розмірності досліджуваного графа. Для виконання алгоритму розпізнавання колективу агентів достатньо однієї фарби.

Список літератури:

1. Dudek G., Jenkin M., Milios E., Wilkes D. Topological exploration with multiple robots. Robotics with Application (ISORA) : Proc. 7th International Symposium. Alaska. 1998.
2. Stopkin A. V. Finite graph exploration by a mobile agent. *Mathematical Modeling and Computing*. Vol. 12, No. 1, pp. 75–82. 2025.
3. Kuipers B. The spatial semantic hierarchy. *Artificial Intelligence*. 2000. V.119, № 1–2. P. 191–233.
4. Fraigniaud P., Jecincas D., Peer G., Pelc A., Peleg D. Graph Exploration by a finite automaton. *Proc. 29 th Internac. Symp. On Math. Foundation of Computer Science (MFCS)*, LNCS 3153, 2004. p. 451–462.
5. Das S., Flochini P., Kutten S., Nayak A., Santoro N. Map construction of unknown graphs by multiple agents. *Theoretical Computer Science*. 2007. V.385, 1–3. P. 34–48.
6. Wang H., Jenkin M., Dymond P. *It can be beneficial to be 'lazy' when exploring graph-like worlds with multiple robots*. In Proceedings of the IASTED International Conference on Advances in Computer Science and Engineering (ACSE). 2009. P. 55–60.
7. Nagavaranu 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>
8. 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>
9. Gongye, X., Wang, Y., Wen, Y., Nie, P., & Lin, P. A simple detection and generation algorithm for simple circuits in directed graph based on depth-first traversal. *Evolutionary Intelligence*, 2022. 1-10.
10. 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.
11. Стьопкін А. В. Мультиагентна система розпізнавання неорієнтованих графів. *Інформаційні технології та суспільство*. Київ, 2024. Вип. 3 (14). 38–43 с.
12. Nagavaranu, S. C., Vachhani, L., Sinha, A., & Buriuly, S. Generalizing multi-agent graph exploration techniques. *International Journal of Control, Automation and Systems*, 2021. 19(1), 491–504.
13. 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>
14. 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>
15. Sapunov S.V. Directional Movement of a Collective of Compassless Automata on a Square Lattice of Width 2. *Cybernetics and Systems Analysis*, 2024. 60(6), 899–906. <https://doi.org/10.1007/s10559-024-00727-x>
16. 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>
17. 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>
18. 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>

Stopkin A.V. SIMPLE GRAPHS EXPLORATION BY A COLLECTIVE OF AGENTS

The paper addresses the problem of exploring finite, connected, undirected graphs without loops and multiple edges by a collective of agents. The goal of the work is to construct a new, efficient graph exploration algorithm for a collective of agents. The article proposes the following methodology to achieve the goal: using a collective of agents consisting of two agents of different types. The first type is the agent-researcher, who can move through the graph, exchange messages with the second agent, and can read and modify labels on graph elements. The second type of agent is the agent-experimenter – a stationary agent located outside the explored graph, in whose memory the results of the agent-researcher's actions are recorded at each step. Based on the received information, the agent-experimenter gradually builds a representation of the explored graph, consisting of a list of edges and a list of nodes. The agents have finite memory at each step but an unbounded growing internal memory. The paper discusses in detail the operational modes of the agent-researcher, indicating the priority of activating these modes during the graph exploration process. It also describes the algorithm for the agent-experimenter, including the detailed procedure for processing the received messages, which is how the studied graph is explored. The paper also analyzes the time, space, and communication complexities of the proposed algorithm, as well as the number of edge transitions the agent-researcher must make to fully explore the studied graph. The scientific novelty lies in the development of an effective method and algorithm for exploring finite undirected graphs without loops and multiple edges, which allows the use of only one color for exploration and provides the possibility of scaling the proposed collective to k agents. Thus, the paper proposes a new method and algorithm for graph exploration, which has quadratic (in terms of the number of nodes of the studied graph) time and space complexities, while the communication complexity of the algorithm is upper-bounded as $O(n^2 \times \log(n))$, where n is the number of nodes of the studied graph. The number of edge transitions the agent-researcher must make to fully explore the studied graph is also upper-bounded as $O(n^2)$. The proposed graph exploration algorithm is based on the depth-first search method.

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