

Стьопкін А.В.

ДВНЗ «Донбаський державний педагогічний університет»

РОЗПІЗНАВАННЯ НЕОРІЄНТОВАНИХ ГРАФІВ МУЛЬТИАГЕНТНОЮ СИСТЕМОЮ

Дане дослідження присвячено вирішенню проблеми розпізнавання скінчених зв'язних неорієнтованих графів без петель та кратних ребер мультиагентними системами. Метою дослідження є розробка нового ефективного алгоритму розпізнавання графів мультиагентною системою. В статті використовується наступна методологія досягнення поставленої мети: використати мультиагентну систему, що складається з двох агентів двох різних типів. Перший агент – це агент-дослідник, який може пересуватися по графу, зчитувати та змінювати мітки на відвіданих елементах графу та обмінюватись повідомленнями з другим агентом. Другий агент – це агент-експериментатор – що знаходиться за межами досліджуваного графу, та займається лише побудовою мапи досліджуваного графу на основі повідомлень, отриманих від агента-дослідника. Агент-дослідник має скінчену пам'ять, яка не залежить від розмірності досліджуваного графа. Агент-експериментатор, в свою чергу, має пам'ять, яка залежить від розмірності графа, внаслідок того, що мапа графу зберігається в його пам'яті. В роботі наведено режими функціонування агента-дослідника із зазначенням пріоритетності активації цих режимів в процесі роботи. Також наведено алгоритм функціонування агента-експериментатора з описом процедур обробки отриманих від агента-дослідника повідомлень. В роботі проведено аналіз часової, ємнісної, комунікаційної складностей побудованого алгоритму та кількість переходів по ребрах, які необхідно виконати агенту-досліднику для повного розпізнавання досліджуваного графа. Науковою новизною є отримання ефективного алгоритму розпізнавання скінчених зв'язних неорієнтованих графів без петель та кратних ребер, який дозволяє використовувати для розпізнавання дві фарби різного кольору та один камінь. Таким чином, в роботі запропоновано новий алгоритм розпізнавання графів, який має квадратичні (від числа вершин досліджуваного графа) часову, ємнісну та комунікаційну складності. Кількість переходів по ребрах, які необхідно виконати агенту-досліднику для повного розпізнавання досліджуваного графа оцінюється зверху як $O(n^2)$, де n – кількість вершин досліджуваного графа. Робота запропонованого алгоритму ґрунтується на методі обходу графа в глибину.

Ключові слова: розпізнавання графів, пості графи, складність алгоритму, обхід в глибину, мультиагентна система.

Постановка проблеми. В наш час досить стрімко розвивається робототехніка, вона набуває все більшої популярності серед молоді та стає все більш затребуваною в різноманітних напрямках діяльності людини. Що зробило її однією з провідних галузей кібернетики. Її витoki пов'язані з прагненням людства полегшити виконання важких або небезпечних для людини завдань, а іноді й повністю замінити людину в таких видах діяльності.

З огляду на те, що залишається велика кількість недосліджених середовищ [1] через їхню недоступність або небезпеку для людини, стає очевидною актуальність задачі їх дослідження за допомогою роботів [2–4]. Зрозуміло, що використовуючи при цьому не одного, а декількох роботів [5–8] підвищується ймовірність успішного виконання поставленого завдання, зокрема у випадках виходу з ладу окремих агентів під час

роботи. Використання декількох агентів також може позитивно вплинути на часову чи ємнісну складність побудованих алгоритмів.

Відома значна кількість наукових праць присвячених дослідженню графів за допомогою одного агента, в яких отримано численні результати щодо можливості такого розпізнавання та його складності. Натомість проблема дослідження графів мультиагентною системою, залишається недостатньо вивченою. Саме тому актуальними є дослідження мультиагентних систем, пов'язані з систематичним проведенням експериментів та їх аналізом. До таких задач належать побудова маршрутів переміщення агентів по невідомому графу, маркування його елементів, збирання та обробка локальної інформації про граф, а також побудова мапи графа за цією інформацією, зокрема й з урахуванням нанесених міток. Крім того, важливими залишаються задачі

оптимізації використання ресурсів, витрат часу, навантаження на канали зв'язку, тощо.

Аналіз останніх досліджень і публікацій. Початок досліджень у цьому науковому напрямі зазвичай пов'язують із роботою С. Shannon [9], у якій розглядалася задача пошуку автоматом заданої цілі в лабіринті. Активні дослідження поведінки автоматів у лабіринтах розпочалися у 1971 році після публікації роботи К. Dorr [10], де описувався обхід шахових лабіринтів скінченими автоматами. Згодом умови задачі розширювалися і автомату необхідно вже було відвідати всі вершини та ребра досліджуваного графа (лабіринту). У подальшому, в роботі [11], було проведено дослідження властивостей невідомого середовища залежно від способу взаємодії автомата з середовищем, а також за різної апіорної інформації про це середовище.

Насправді, дослідження з аналізу графів включають низку окремих завдань. Наприклад, задача самолокалізації (визначення вершини графа, в якій агент знаходиться на початку дослідження), задача контролю карти (перевірка ізоморфізму між досліджуваним графом і еталонною картою), задача повного розпізнавання графа (побудова мапи досліджуваного графа).

Для вирішення останньої задачі існує декілька підходів у відповідності до яких розроблено низку алгоритмів обходу графів мобільними агентами та способів маркування елементів графа цими агентами за допомогою різних фарб або камінців, які дозволяють розпізнавати різноманітні графи з точністю до ізоморфізму.

Так, у роботі [12] запропоновано метод, за яким агент під час руху маркує лише інцидентори пройдених ребер. Для дослідження ребер агент проходить по ним, а потім повертається до початкової вершини найкоротшим шляхом. При цьому він запам'ятовує мітки пройдених інциденторів, які однозначно визначають вершину, до якої агент дійшов під час руху цим ребром. У роботі [13] агенти обмінюються інформацією про виконані завдання з базовою станцією через спеціальну мережу, що має обмеження на обсяг переданих даних. Інакше кажучи, для координації дій між агентами попередньо формується мережа, яка відповідає вказаним умовам. Обмін даними з базовою станцією відбувається у заздалегідь визначених точках. У статті [5] зроблено спробу розробити загальні концепції та вимоги до задачі розпізнавання графа мультиагентними системами. Запропоноване представлення мультиагентної системи орієнтоване на визначення

загальних умов завершення процесу розпізнавання графа та забезпечення його завершення за скінченний час. У роботі запропоновано модифікацію матриці інцидентності для організації інформаційного обміну між агентами, а також наведено загальний алгоритм колективної роботи агентів під час дослідження графа. Алгоритм базується на запропонованій узагальненій структурі та модифікованій матриці інцидентності. У роботі [7] розглядається нескінченний в усі сторони граф квадратної решітки. Усі його вершини позначено позначками з деякої відомої множини, але конкретна функція розмітки невідома. Завданням автомата є визначення двох двійок протилежних напрямків на укладанні вищезгаданого графа на цілочисельну решітку, або розпізнавання розмітки. Доведено, що поодинокий автомат не може розпізнати мінімальну детерміновану розмітку нескінченного графа квадратної решітки, але автомат з одним каменем може. Запропоновано новий тип експериментів з графами, призначений для розпізнавання їх розмітки та запропоновано метод побудови та проведення розпізнавального експерименту для нескінченного позначеного графа квадратної решітки.

Постановка завдання. Проаналізувавши дослідження з роботи мультиагентних систем, можна прийти до висновку, що в них мало уваги приділено вивченню процесу обміну інформацією агентами. Що робить актуальними проведення експериментів по розпізнаванню графів мультиагентними системами та оптимізацію витрат часу, ресурсів та навантаження на канал зв'язку між агентами.

Таким чином, метою даного дослідження є створення ефективного алгоритму розпізнавання скінчених зв'язних простих неорієнтованих графів мультиагентною системою, ефективність якого або перевищить раніше розроблені алгоритми, або зменшить кількість використовуваних в них ресурсів [10, 14–17]. Тобто важливою частиною нашої роботи є дослідження часової, ємнісної й комунікаційної складностей побудованого алгоритму та кількості переходів по ребрах, які необхідно виконати агенту-досліднику для повного розпізнавання досліджуваного графа.

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 $L = V \cup E \cup I$ назовемо множиною всіх елементів графа G . Функцією розфарбування графа G назовемо сюр'єктивне відображення $\mu: L \rightarrow \{w, r, b\}$, де w – білий колір, r – червоний, а b – чорний. Пару (G, μ) будемо називати розфарбованим графом. Послідовність $u_1, u_2, \dots, u_k$ попарно суміжних вершин графа G назовемо 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

назовемо таку бієкцію $\phi: V_G \rightarrow V_H$, що $(v, u) \in E_G$ тоді й тільки тоді, коли $(\phi(v), \phi(u)) \in E_H$. Таким чином, ізоморфні граfi рівні з точністю до позначення вершин і розфарбування їх елементів.

В статті розглядається мультиагентна система, що складається з двох агентів: один агент-дослідник і один агент-експериментатор. Агент-дослідник може переміщуватися графом і змінювати забарвлення елементів графа. Агент-експериментатор відповідає за побудову мапи графа в своїй пам'яті на основі повідомлень, отриманих від агента-дослідника.

На початку роботи агент-дослідник розміщується в довільній вершині графа G та фарбує її в свій колір. Агент-експериментатор в цей момент додає нову вершину в множину V_H в своїй пам'яті. Агент-дослідник пересувається по графу з вершини v у вершину u по ребру (v, u) , може змінювати забарвлення вершин v, u , ребер (v, u) , інциденторів $((v, u), v), ((v, u), u)$. Перебуваючи у вершині v , агент-дослідник може сприймати мітки всіх елементів околу $Q(v)$ та виходячи з цієї інформації визначає, яким ребром буде далі переміщатися та як фарбуватиме елементи графа. Агент-експериментатор, в свою чергу, передає, приймає та ідентифікує повідомлення, отримані від агента-дослідника та фіксує в своїй пам'яті результат функціонування агента-дослідника на кожному кроці. На основі цієї інформації агент-експериментатор будує мапу досліджуваного графа G , спочатку невідомого агентам, списками ребер і вершин.

Зауважимо, що запропонований алгоритм має декілька особливостей: по-перше, граф G агентам не відомий; по-друге, запропонований метод

розпізнавання графа ґрунтується на стратегії пошуку в глибину; по-третє, при обході графа G агенти створюють неявну нумерацію пройдених вершин. На основі побудованої нумерації й відбувається розпізнавання графа G шляхом побудови графа H ізоморфного G . У процесі обходу агент-дослідник будує неявне дерево пошуку в глибину всі ребра якого поділяються на ті, що забарвлюються при першому проходженні в червоний колір та належать самому дереву й ті, що не належать дереву і забарвлюються при першому проходженні в чорний колір – зворотні ребра. Вершина, у якій всі інцидентні їй ребра розпізнані, забарвлюється в чорний колір. Алгоритм закінчує роботу, коли червоний шлях стає порожнім, а всі вершини чорними.

У роботі агента-дослідника є два режими:

Звичайний режим. Агент-дослідник рухається вперед по білих вершинах, забарвлюючи вершини, ребра, що з'єднують ці вершини та дальні інцидентори в червоний колір. Якщо в околі вершини в якій знаходиться агент-дослідник немає білих ребер чи вершин, тобто відсутні можливі шляхи переміщення, то агент-дослідник повертається назад, забарвлюючи пройдені вершини, ребра та ближні інцидентори у чорний колір. Повернувшись до початкової вершини, агент-дослідник завершує роботу. На кожному кроці агент-дослідник обмінюється даними з агентом-експериментатором.

Режим розпізнавання зворотних ребер. Якщо при русі вперед у вершині v було виявлено зворотне ребро, то агент-дослідник припиняє роботу у звичайному режимі та перемикається у режим розпізнавання зворотних ребер. Агент-дослідник залишає в вершині камінь. Після цього він починає рухатись назад своїм шляхом, до виявлення вершини інцидентної зворотному ребру (тут йдеться мова про біле ребро, у якого дальня вершина пофарбована в червоний колір і в ній знаходиться камінь) та переходить по цьому ребру, фарбуючи його в чорний колір. Далі повертається до попередньої вершини та продовжує рух назад. На цьому етапі можливі два випадки. Випадок, коли агент-дослідник знайде зворотне ребро. У цьому випадку агент-дослідник повторює процедуру розпізнавання зворотного ребра. Та випадок, коли агент-дослідник досягнув початкової вершини, що буде означати, що ним розпізнано всі помічені зворотні ребра. Після чого агент повертається в кінець червоного шляху, де знаходиться камінь, підбирає камінь та продовжує роботу в звичайному режимі.

Алгоритм роботи агента-експериментатора:

Вхід: список список повідомлень M від агента-дослідника.

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

Дані: V_H , E_H – списки вершин и ребер графа H , ізоморфного графу G . ct – лічильник кількості відвіданих вершин графа G агентом-дослідником. M – список повідомлень від агента-дослідника. i – лічильник, що використовується для з'ясування номерів дальніх вершин зворотних ребер після того як встановлено камінь у їх спільну вершину. $STOP$ – змінна, що використовується агентом-дослідником, для повідомлення агенту-експериментатору, про завершення обходу графа. $r(1) \dots r(t)$ – список номерів вершин червоного шляху, де t – довжина цього списку. Mes – змінна, в якій зберігається поточне повідомлення.

Алгоритм роботи агента-експериментатора:

1. $ct := 1, M := \emptyset, i := 0, E_H := \emptyset, STOP := 0, t := 1, r(t) := ct, V_H := \{1\}, Mes := ""$;
2. while $STOP = 0$ do
3. if $M \neq \emptyset$,
4. then do
5. зчитуємо в Mes повідомлення і видаляємо його з черги M ;
6. LIST_PROCESSING();
7. end do;
8. end do;
9. друк V_H, E_H .

LIST_PROCESSING():

1. if $Mes := "FORWARD"$; then FORWARD();
 2. if $Mes := "BACK"$ then BACK();
 3. if $Mes := "RETURN"$ then RETURN();
 4. if $Mes := "FORWARD_IE"$ then FORWARD_IE();
 5. if $Mes := "RECOGNIZED_IE"$ then RECOGNIZED_IE();
 6. if $Mes := "STOP"$ then STOP();
- FORWARD(): $ct := ct + 1$; $t := t + 1$; $r(t) := ct$;
 $V_H := V_H \cup \{ct\}$; $E_H := E_H \cup \{r(t-1), r(t)\}$;
 BACK(): Зі списку $r(1) \dots r(t)$ видаляється елемент $r(t)$; $t := t - 1$;
 RETURN(): $i := i + 1$;
 FORWARD_IE(): $E_H := E_H \cup \{r(t), r(t-i)\}$;
 RECOGNIZED_IE(): $i := 0$;
 STOP(): $STOP := 1$;

На початку роботи алгоритму розпізнавання, при $n \geq 3$, як мінімум два рази, коли агент-дослідник відвідує білі вершини досліджуваного графа G , виконується процедура FORWARD(v). При цьому процедурою FORWARD() створю-

ється нова вершина графа H . Таким чином, процес виконання даного алгоритму індукуює відображення $\phi: V_G \rightarrow V_H$ вершин графа G в вершини графа H . Причому рівність $\phi(v) = ct$ встановлюється, коли вершина v фарбується в червоний колір.

Зазначене відображення природнім чином встановлює неявну нумерацію вершин графу G . Більш того, відображення ϕ є бієкцією, оскільки розглядаються виключно зв'язні графи, а в таких графах всі вершини досяжні з будь-яких початкових вершин. Тому всі вершини досягаються агентом-дослідником і при першому ж відвідуванні вершини, агенти надають їй неявний номер.

З опису алгоритму видно, що агент-дослідник проходить абсолютно всі ребра графу G , оскільки після виконання алгоритму всі ребра графу зафарбовані в чорний колір.

Виконуючи процедуру FORWARD() агент-експериментатор розпізнає деревне ребро (v, u) і так нумерує вершину u , що ребру (v, u) графа G однозначно відповідає ребро $(\phi(v), \phi(u))$ графа H . Виконуючи процедуру FORWARD_IE(), агент-експериментатор розпізнає зворотне ребро (v, u) графа G і ставить йому в однозначну відповідність ребро $(\phi(v), \phi(u))$ графа H . А так як всі ребра графа поділяються виключно на деревні та зворотні, то можна зробити висновок, що відображення ϕ є ізоморфізмом графа G на граф H .

Теорема. Два агенти, виконавши алгоритм розпізнавання графа, розпізнають будь-який граф з точністю до ізоморфізму.

Розглянемо докладніше властивості червоного шляху. З опису алгоритму слідує, що на кожному кроці алгоритму червоний шлях – це простий шлях, що бере початок у вершині v з номером $\phi(v) = 1$, що закінчується у вершині u з номером $\phi(u) = ct$. Отже, довжина червоного шляху не перевищує n . Щоразу при виконанні процедур FORWARD(v) та BACK(v) агент-дослідник проходить одне ребро. При одноразовому виконанні процедури з режиму розпізнавання зворотних ребер агент-дослідник розпізнає не більше $n - 2$ зворотних ребер в процесі чого: витрачає один хід на встановлення каменя у вершину, по два рази (прохід назад та повернення в останню червону вершину) проходить не більше $n - 1$ ребер червоного шляху, а також по два рази проходить не більше $n - 2$ зворотних ребер при їх фарбуванні.

При підрахунку часової складності алгоритму вважатимемо, що ініціалізація алгоритму і вибір однієї з можливих процедур займають деяку

постійну кількість одиниць часу. Також будемо вважати, що прохід по ребру агента-дослідника та обробка команди агентом-експериментатором, отриманої на даному етапі від агента-дослідника, здійснюється за час, що дорівнює деякій константі. Прийнемо, що час пересилання одного повідомлення дорівнює константі, яка не перевищує часу проходження по ребру агента-дослідника. Причому загальний час, що витрачається на аналіз околу $Q(v)$ вершини v та вибір необхідних ребер оцінюється зверху як $O(n^2)$.

Тоді часова складність алгоритму визначається наступними співвідношеннями:

1. Процедури $FORWARD(v)$ і $BACK(v)$ виконуються не більше ніж $2 \times (n-1)$ разів та загальний час їх виконання оцінюється як $O(n)$.

2. Час, що витрачається на розпізнавання зворотних ребер оцінюється як $(n-2) \times (2 \times (n-1) + 2 \times (n-2))$ разів, тобто по суті $O(n) \times O(n)$ разів, а отже загальний час виконання процедури розпізнавання зворотних ребер оцінюється як $O(n^2)$.

Таким чином, верхня оцінка числа переходів по ребрах, що здійснюються агентом-дослідником оцінюється зверху як $O(n) + O(n^2) = O(n^2)$, а сумарна часова складність $T(n)$ алгоритму задовольняє співвідношенню: $T(n) = O(n^2)$.

Ємнісна складність $S(n)$ алгоритму визначається сумарною складністю списків E_n , V_n та $r(1), \dots, r(t)$ складність яких визначається величинами $O(n^2)$, $O(n)$, $O(n)$ відповідно.

А отже верхня оцінка ємнісної складності алгоритму задовольняє співвідношенню: $S(n) = O(n^2)$.

При підрахунку комунікаційної складності алгоритму, з урахуванням того, що агент-дослідник може відправляти агенту-експериментатору лише одне з шести можливих повідомлень ($FORWARD, BACK, RETURN, FORWARD_IE, RECOGNIZED_IE, STOP$), вважатимемо, що одне повідомлення має обсяг, рівний одиниці пам'яті.

Так як при роботі в звичайному режимі, на кожному кроці каналом зв'язку передається

одне повідомлення ($FORWARD, BACK, STOP$), то загальний обсяг переданої інформації в цьому режимі буде оцінюватись зверху як $2 \times O(n)$. При роботі в режимі розпізнавання зворотних ребер, агент-дослідник при русі назад по червоному шляху, покроково відправляє одне сповіщення ($RETURN$), тобто загальний обсяг переданої інформації в цьому режимі також буде оцінюватись зверху як $n \times O(n)$. При першому переході по зворотних ребрах відправляється одне повідомлення ($FORWARD_IE$) тобто загальний обсяг переданої інформації в цьому режимі також буде оцінюватись зверху як $n \times O(n)$. При переході назад по зворотному ребру ніяких повідомлень не надсилається. Також буде надіслано одне повідомлення ($FORWARD_IE$), при поверненні в вершину, в якій знаходиться камінь, тобто обсяг переданої інформації в цьому режимі буде оцінюватись зверху як $n \times O(1)$.

Таким чином загальний обсяг інформації, що передається агентами оцінюється як $K(n) = 2 \times O(n) + n \times (O(n) + O(n) + O(1)) = O(n^2)$.

Теорема. Часова, ємнісна, комунікаційна складності алгоритму розпізнавання та кількість переходів по ребрах, що здійснюються агентом-дослідником, оцінюються зверху як $O(n^2)$, де n – кількість вершин в графі. Для розпізнавання достатньо двох фарб різного кольору.

Висновки. Дослідження присвячено проблемі розпізнавання графів мультиагентною системою. Запропоновано новий алгоритм розпізнавання скінчених зв'язних неорієнтованих графів без петель та кратних ребер, квадратичної часової, ємнісної та комунікаційної складностей. Верхня оцінка числа переходів по ребрах, що здійснює агент-дослідник для повного розпізнавання графа дорівнює $O(n^2)$. Агент-дослідник має скінчену пам'ять, яка не залежить від розмірності досліджуваного графа. Агент-експериментатор, в свою чергу, має пам'ять, яка залежить від розмірності графа, внаслідок того, що мапа графу зберігається в його пам'яті. Для виконання алгоритму розпізнавання необхідно дві фарби різного кольору і один камінь.

Список літератури:

1. Goth G. Exploring new frontiers. *Communications of the ACM*. 2009. 52 (11). P. 21–23.
2. Sapunov S.V. Experiments on Recognition of Infinite Grid Graph Labelling. *Праці ІПММ НАН України*, 2021. 35 (1), 67–78. URL: <https://doi.org/10.37069/1683-4720-2021-35-6>
3. 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.
4. Stopkin A. V. Finite graph exploration by a mobile agent. *Mathematical Modeling and Computing*. 2025. Vol. 12, No. 1, P. 75–82.

5. Nagavarapu S.C., Vachhani L., Sinha A. et al. Generalizing Multi-agent Graph Exploration Techniques. *International Journal of Control, Automation and Systems*. 2020. URL: <https://doi.org/10.1007/s12555-019-0067-8>
6. Sapunov S.V. Minimal Deterministic Traversable Vertex Labelling of Infinite Square Grid Graph. Праці ІПММ НАН України, 2020. 34, 118–132. URL: <https://doi.org/10.37069/1683-4720-2020-34-12>
7. 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, is. 6, P. 899–906. URL: <https://doi.org/10.1007/s10559-024-00727-x>
8. Стопкін 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.
9. Shannon C.E. Presentation of a maze-solving machine. *Cybernetics Trans, of the 8 th Conf. of the Josiah Macy Jr. Found* / Editor: H. Foerster. 1951. P. 173–180.
10. Dopp K. Automaten in labirinth. *Electronische Informationsverarbeitung und Kybernetik*. 1971. V.7, № 2. P. 79–94.
11. Dudek G., Jenkin M. Computational principles of mobile robotics. Cambridge Univ. press. 2000. 280 p.
12. Dudek G., Jenkin M., Milios E., Wilkes D. Map validation in a graphlike world. *Proceedings of the 13th International Joint Conference on Artificial Intelligence*. San Fransisco: Morgan Kaufmann Publishers Inc., 1993. P. 1648–1653.
13. 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. URL: <https://doi.org/10.1007/s10514-017-9652-y>
14. Stepkin A. Using a Collective of Agents for Exploration of Undirected Graphs. *Cybernetics and Systems Analysis*. 2015. V.51, № 2. P. 223–233. URL: <https://doi.org/10.1007/s10559-015-9715-z>
15. Стопкін А. В. Мультиагентна система розпізнавання неорієнтованих графів. Інформаційні технології та суспільство. Київ, 2024. Вип. 3 (14). 38–43 с.
16. Стопкін А.В. Мультиагентна система розпізнавання графів. Штучний інтелект, 2025, №2, P. 84–93. URL: <https://doi.org/10.15407/jai2025.01.027>
17. Dudek G., Jenkin M., Milios E., Wilkes D. (1998) Topological exploration with multiple robots. *Robotics with Application (ISORA) : Proc. 7th International Symposium*. Alaska.

Stopkin A.V. UNDIRECTED GRAPH EXPLORATION USING A MULTI-AGENT SYSTEM

This study is devoted to solving the problem of graph exploration by multi-agent systems in the class of finite, connected, undirected graphs without loops and multiple edges. The goal of the research is to develop a new efficient algorithm for graph exploration by a multi-agent system.

The proposed methodology involves the use of a multi-agent system consisting of two agents of different types. The first is an agent-researcher, which can move through the graph, read and modify labels on visited elements of the graph, and exchange messages with the second agent. The second is an agent-experimenter, which is located outside the graph and is solely responsible for constructing a map of the explored graph based on the messages received from the agent-researcher.

The agent-researcher is equipped with finite memory, independent of the size of the explored graph. The agent-experimenter, in turn, has memory that depends on the size of the graph, since the map is stored in its memory.

The paper outlines the operating modes of the agent-researcher, indicating the priority for their activation during exploration. It also presents the functioning algorithm of the agent-experimenter, including procedures for processing the messages received from the agent-researcher.

The paper analyzes the time, space, and communication complexities of the proposed algorithm, as well as the number of edge transitions that must be performed by the agent-researcher to completely explore the graph.

The scientific novelty lies in obtaining an efficient algorithm for the exploration of finite, connected, undirected graphs without loops and multiple edges. The algorithm uses two distinct colors and one pebble to exploration.

Thus, the paper proposes a new graph exploration algorithm with quadratic time, space, and communication complexities (with respect to the number of nodes in the graph). The upper bound on the number of edge transitions required by the agent-researcher to fully explore the graph is estimated as $O(n^2)$, where n is the number of nodes in the explored graph. The operation of the proposed algorithm is based on the depth-first search (DFS) strategy.

Key words: graph exploration, finite graphs, algorithm complexity, depth-first search, multi-agent system.