**УДК 519.1:519.7:007.5**
**DOI https://doi.org/10.32689/maup.it.2024.3.5**

***Андрій СТЬОПКІН***
*кандидат фізико-математичних наук, доцент, доцент кафедри математики та інформатики ДВНЗ «Донбаський державний педагогічний університет», stepkin.andrej@gmail.com*
***ORCID:** 0000-0002-6130-9920*

## МУЛЬТИАГЕНТНА СИСТЕМА РОЗПІЗНАВАННЯ НЕОРІЄНТОВАНИХ ГРАФІВ

***Анотація.*** *Робота присвячена проблемі розпізнавання простих неорієнтованих графів мобільними агентами.*

***Метою роботи*** *є побудова нового ефективного алгоритму розпізнавання скінчених неорієнтованих графів без петель та кратних ребер мультиагентною системою. В статті запропоновано наступну* ***методологію*** *до досягнення поставленої мети: використати мультиагентну систему, що складається з трьох агентів двох різних типів. Перший тип – це агенти-дослідники, які рухаються по графу, можуть зчитувати мітки на елементах графу й змінювати забарвлення цих елементів. Також ці агенти можуть обмінюватися повідомленнями з агентом другого типу. Агенти-дослідники мають скінчену пам'ять та для розпізнавання графу використовують по дві фарби різного кольору кожен (всього три фарби різного кольору). Другий тип – це агент-експериментатор – нерухомий агент, що знаходиться поза межами графу, в пам'яті якого на кожному кроці фіксується результат функціонування агентів-дослідників. На основі отриманої інформації агент-експериментатор в своїй пам'яті поступово вибудовує представлення досліджуваного графа списком ребер і списком вершин. У статті детально розглянуто режими роботи агентів-дослідників із зазначенням пріоритетності активації цих режимів в процесі роботи. Також наведено алгоритм роботи агента-експериментатора з детальним описом процедур обробки отриманих повідомлень, на підставі яких і відбувається розпізнавання досліджуваного графа. Також в статті проведено аналіз часової, ємнісної, комунікаційної складності побудованого алгоритму та проаналізовано кількість переходів по ребрах, які необхідно виконати агентам-дослідникам для повного розпізнавання графа.*

***Науковою новизною*** *є отримання більш ефективного алгоритму розпізнавання графів, який дозволяє використовувати агентів-дослідників зі скінченою пам'яттю та дає можливість в подальшому масштабувати розглянуту мультиагентну систему до $k$ агентів.*

***Висновки.*** *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 виконують агенти-дослідники оцінюється як $O(n)$ де $n$ – кількість вершин досліджуваного графа. Робота запропонованого алгоритму розпізнавання ґрунтується на методі обходу графа в глибину.*

***Ключові слова:*** *розпізнавання графів, прості скінчені графи складність алгоритму, обхід в глибину, колектив агентів.*

## Andrii STOPKIN. MULTI-AGENT SYSTEM FOR NON-ORIENTED GRAPHS EXPLORATION

***Abstract.*** *The work is devoted to the problem of simple undirected graphs exploration by mobile agents.*

***The purpose*** *of this work is to develop a new effective algorithm for exploring undirected graphs without loops and multiple edges by a multi-agent system. The article proposes the next* ***methodology*** *to achieve the goal: to use a multi-agent system that consists of three agents of two different types. The first type is agents-researchers, that move along the graph, can read the labels on the graph elements and color these elements. Also, these agents can exchange messages with the second type of agent. Agents-researchers have a finite memory and use two different colors each (total of three different colors) to a graph exploration. The second type is an agent-experimenter is a stationary agent located outside the graph, in whose memory the result of the functioning of the agents-researchers is recorded at each step. On the basis of the received information, the agent-experimenter in his memory gradually builds a map of the investigated graph. In the article in detail examines the modes of operation of agents-researchers with an indication of the priority of activation of these modes. Also given the algorithm of the agent-experimenter with a detailed description of the procedures for processing the received messages, on the basis of which the graph is explored. Also, the article analyzes the time, space, and communication complexity of the algorithm and analyzes the number of transitions along the edges that must be performed by agents-researchers for complete the graph exploration.*

***A scientific novelty*** *is the development of a more efficient graph exploration algorithm, which allows the use of agents-researchers with finite memory and makes it possible to further scale the considered multi-agent system up to $k$ agents.*

***Conclusions.*** *Thus, the paper proposes a new graph exploration algorithm that has quadratic (from the number of nodes of the graph) time, space and communication complexities. The number of edge transitions performed by agents-researchers is estimated as O(n), where n is the number of nodes of the investigated graph. The algorithm is based on depth-first traversal method.*

***Key words:*** *graph recognition, simple finite graphs, algorithm complexity, depth traversal, collective of agents.*

**Вступ. Постановка проблеми** в загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями. В наш час у світі існує величезна кількість різноманітних недосліджених середовищ [1]. Причому задачі з розпізнавання досить різноманітні, починаючи з дослідження невеликих середовищ за допомогою наноботів [13] і закінчуючи дослідженням масштабних поверхонь віддалених планет, за допомогою планетоходів. На наш погляд, це є однією з основних причин активного розвитку такого напряму математичної кібернетики як теорія дискретних динамічних систем. Дослідження середовища – це свого роду дискретна система, представлена як модель взаємодії керуючої та керованої систем, взаємодія яких найчастіше представляється як процес переміщення керуючого автомата графом керованої системи. Що, зрештою, і призвело до інтенсивного розвитку дослідження поведінки автоматів в лабіринтах [9, 16]. Також в наш час досить активно вивчаються різноманітні мультиагентні системи. В залежності від цілей, проблеми керування мультиагентними системами можна умовно розділити на три основні види: проблема керування групою агентів [5, 14], проблема формації групи агентів [6, 11] та проблеми консенсусу [2, 18]. Всі види активно досліджуються в наш час.

Враховуючи вищесказане, можна зробити висновок, що дослідження питання розпізнавання графів за допомогою колективу агентів є досить актуальним. Що, в свою чергу, робить актуальною задачу проведення систематичного дослідження експериментів з розпізнавання графів колективом агентів, що блукають по ньому. Тобто створення маршрутів руху агентів по невідомому графу, розмітки його елементів, збору та обробки локальної інформації про граф та способів побудови мапи графа за цією інформацією, з точністю до позначок на елементах графа. А також завдання, пов'язані з оптимізацією витрат ресурсів та витрат часу на розпізнавання графів. Зрозуміло, що при розпізнаванні графа декількома агентами, що блукають по ньому, основною проблемою є проблема ефективності їх взаємодії, з метою зменшення витрат часу і пам'яті на розпізнавання. Потрібно розробити такі алгоритми переміщення, у яких блукаючі агенти не заважають один одному і не дублюють роботу один одного. Таким чином, **метою роботи** є створення ефективного методу і побудова відповідного алгоритму розпізнавання (побудови мапи) неорієнтованих простих графів за допомогою колективу агентів, а також дослідження часової, ємнісної, комунікаційної складностей побудованого алгоритму та кількості переходів по ребрах, які виконують агенти-дослідники для повного розпізнавання графа. У роботі розглядається колектив із трьох агентів: два агента-дослідника блукають по графу, перефарбовують його елементи та передають інформацію про свої дії агенту-експериментатору. Взаємодія агентів-дослідників здійснюється за рахунок фарбування елементів графа. Розглянемо алгоритм вирішення нашої проблеми у разі, коли два агенти-дослідники $A$ і $B$ одночасно пересуваються по невідомому скінченому неорієнтованому простому графу та обмінюються інформацією з агентом-експериментатором, який і будує мапу досліджуваного графу в своїй пам'яті у вигляді списків ребер та списків вершин. У роботі запропоновано алгоритм побудови маршрутів агентами-дослідниками, що дозволяють агенту-експериментатору точно розпізнати граф. Кожному агенту-досліднику для роботи необхідно дві фарби різного кольору: у агента $A$ це червона та чорна ($r$ і $b$), у агента $B$– жовта и чорна ($y$ і $b$). Тобто агентам необхідно всього три фарби різного кольору. Алгоритм заснований на методі обходу графа в глибину [3].

**Аналіз останніх досліджень і публікацій.**

Початком досліджень проблеми обходу графів прийнято вважати роботу К. Шеннона [15]. Активне ж вивчення особливостей функіонування автоматів в лабіринтах починається з дослідження К. Деппа [7], який вивчав поведінку автоматів при роботі в шахових лабіринтах. Також вивченням даного питання займався Г. Дудек [8, 10] в роботах якого проводилися дослідження властивостей невідомого середовища при різній взаємодії автомата з операційним середовищем та при різній апріорній інформації про нього. В тому числі й за умови роботи мультиагентних систем [10]. В подальшому вивчення особливостей роботи мультиагентних систем знайшло свій розвиток в низці праць [4, 10, 12, 16, 17].

**Виклад основного матеріалу дослідження.**

Нехай $G=(V,E)$ – зв'язний неорієнтований скінчений граф без петель і кратних ребер, де $V$ – множина вершин, $E$ – множина ребер (двохелементних підмножин $(v,u)$, де $v,u \in V$). Трійку $((v,u),u)$ будемо називати інцидентором (точкою з'єднання) ребра $(v,u)$ і вершини $u$. Послідовність $u_1,u_2,...,u_k$ попарно суміжних вершин графа $G$ називається шляхом довжини $k$. Околом $Q(v)$ вершини $v$, будемо називати множину елементів графа, що складається з самої вершини $v$, всіх вершин $u$ суміжних з $v$ всіх ребер $(v,u)$ та всіх інциденторів $((v,u),v),((v,u),u)$. Потужність множин вершин $V$ та ребер $E$ позначимо через $n$ та $m$ відповідно. Зрозуміло, що $m \le \frac{n(n-1)}{2}$ елементів. Ізоморфізмом графа $G$ і графа $H$ назвемо таку бієкцію $\varphi: V_G \to V_H$, 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 та розфарбування їх елементів. Зворотним ребром для агента-дослідника називатимемо біле ребро, дальня вершина якого забарвлена в «свій» колір. Під перешийком розуміємо ребро, що з'єднує вершини, що належать областям роботи різних агентів-дослідників. Зазначимо, що колектив агентів, який ми розглядаємо складається з трьох агентів: два агенти-дослідники $A$ і $B$ та один агент-експериментатор. Агенти дослідники можуть рухатись по графу, змінювати забарвлення елементів графа, записувати номери в вершини графа та передавати повідомлення про свої дії агенту експериментатору, який аналізуючі отримані повідомлелння, будує мапу графа (у вигляді списків вершин та списків ребер) в своїй пам'яті. Агенти-дослідники мають скінчену пам'ять. На початку роботи вони поміщаються в довільні неспівпадаючі вершини графа $G$ та одразу нумерують їх і передають номери агенту-експериментатору, який одразу додає їх в множину вершин $V_H$. Перебуваючи у вершині $v$, агент-дослідник може сприймати мітки всіх елементів околу $Q(v)$ та номери суміжних із нею вершин. Спираючись на цю інформацію він визначає, по якому ребру буде далі переміщатися, і як фарбуватиме елементи графа. Агент-експериментатор передає, приймає та ідентифікує повідомлення, отримані від агента-дослідника, має скінчену на кожному кроці, але необмежено зростаючу внутрішню пам'ять. Розглянемо режими роботи агентів-дослідників. При описанні режимів, у дужках будемо зазначати повідомлення, які відправляють агенти-дослідники агенту-експериментатору, працюючи в цих режимах (MESSAGE_A; MESSAGE_B). Агент-експериментатор, в свою чергу, обробляє отримане повідомлення та відправляє агенту досліднику дані, необхідні для завершення ходу.

*Звичайний режим роботи*. Агент-дослідник обирає з околу вершини, в якій він знаходиться, довільну білу вершину і переходить до неї. Таким чином агент-дослідник рухається вперед по білих вершинах, фарбуючи вершини, ребра, що їх з'єднують та дальні інцидентори в «свій» колір (FORWARD_A; FORWARD_B), далі отримавши від агента-експериментатора номер, записує його у вершину. Якщо в околі вершини, в якій стоїть агент-дослідник не знайдеться білої вершини, для подальшого руху вперед, то агент-дослідник $A$ відправляє агенту-експериментатору повідомлення ASSIGN_A, на що витрачає один хід, далі повертається назад, забарвлюючи пройдені вершини, ребра та ближні інцидентори в чорний колір (BACK_A). Агент-дослідник $B$, не виявивши білої вершини для подальшого руху вперед, відразу починає рух назад, забарвлюючи пройдені вершини, ребра та ближні інцидентори у чорний колір (BACK_B). Повернувшись у початкову вершину, агент-дослідник завершує роботу (STOP_A; STOP_B). Таким чином будується два дерева методом обходу графа в глибину.

*Режим розпізнавання зворотних ребер*. Якщо під час руху вперед було виявлено зворотне ребро, то агент-дослідник сканує окіл вершини в якій він знаходиться, покроково зчитує номери всіх суміжних вершин, інцидентних зворотним ребрам, фарбуючи ближній інцидентор в чорний колір, і відправляє номер агенту-експериментатору (INV_A$(x)$; INV_B $(k)$), де $x,k$ – номери, записані агентами $A$ і $B$, відповідно, у вершинах свого шляху.

*Режим розпізнавання перешийків*. Цей режим дещо відрізняється для кожного з агентів-дослідників. Якщо, при русі вперед, агент $A$ виявив у вершині перешийки обидва інцидентори яких білі, і в жодній з дальніх вершин цих перешийків немає агента $B$ (або агент $B$ перебуває у одній з цих вершин, але вже розпізнав всі, раніше виявлені ним, перешийки у поточну для себе вершину, або агент $B$ виконує повернення назад своїм шляхом в звичайному режимі роботи), то агент $A$ покроково відправляє агенту-експериментатору всі номери дальніх вершин перешийків (ISTHM_A$(x)$, де $x$ – номер, записаний агентом $B$, у вершині свого шляху), фарбуючи їх ближні інцидентори в чорний колір. Якщо агент $B$ знаходиться в одній з дальніх вершин перешийків і ще не розпізнавав інцидентні їй перешийки або не повертається назад по своєму шляху, то агент $A$ відправляє агенту-експериментатору номери всіх дальніх вершин, виявлених перешийків, фарбуючи ближній інцидентор в чорний колір, крім номера вершини, в якій знаходиться агент $B$. Зауважимо, що агент $A$, дізнається про знаходження агента $B$ в суміжній вершині в результаті сканування околу на наявність перешийків, але про те чи можна розпізнавати перешийок в дальній вершині якого знаходиться $B$, агент $A$ дізнається зі значення змінної $mr\_A$, значення якої запитується у агента-експериментатора. Якщо ж, при русі вперед, перешийки з білими інциденторами виявив агент $B$ і в жодній з дальніх вершин цих перешийків немає агента $A$ (або $A$ знаходиться в одній з цих вершин, але виконує повернення назад по своєму шляху), то $B$ покроково передає агенту-експериментатору номери всіх дальніх вершин виявлених перешийків (ISTHM_B$(k)$, де $k$ – номер, записаний агентом $A$, у вершинах свого шляху). Якщо ж агент $A$ знаходиться в одній з дальніх вершин перешийків і не виконує повернення назад своїм шляхом, то $B$ не виконує жодних дій до виходу $A$ з околу вершини, в якій знаходиться $B$. Агент $B$, дізнається про перебування $A$ в суміжній вершині в результаті сканування околу на наявність перешийків, але про те чи можна розпізнавати перешийки за наявності в одній з дальніх вершин агента $A$, агент $B$ дізнається зі значення змінної $mr\_B$, що запитується у агента-експериментатора.

*При одночасному попаданні двох агентів-дослідників в одну білу вершину* кожен агент-дослідник забарвлює вершину наполовину, і вона стає червоно-жовтою. Агент $B$, на наступному кроці, відступає назад своїм шляхом, видаляючи мітки, залишені ним на попередньому кроці (видаляється фарба з ребра і ближнього інцидентора) (COME_BACK_B), і переключається в звичайний режим роботи. Агент $A$ бачить різнокольорову вершину як свою, але при розпізнаванні забарвлює у чорний колір всю вершину.

При попаданні агентів-дослідників у ситуацію, коли у вершині можливий вибір відразу кількох режимів роботи, першим буде обрано режим розпізнавання перешийків, за ним режим розпізнавання зворотних ребер і нарешті звичайний режим роботи. Потрапляння двох агентів дослідників в одну білу вершину тут не розглядається, оскільки така ситуація призведе до змін у роботі виключно агента $B$

і в цей момент інші режими роботи для нього будуть недоступні. Виконуючи обхід графа, агенти $A$ і $B$ створюють відповідно червоний та жовтий шляхи. Розглянемо принцип побудови агентами шляху «свого» кольору. При русі в білу вершину червоний (жовтий) шлях подовжується, при русі назад своїм шляхом – коротшає. Якщо агент-дослідник повернувся у вершину, з якої почав обхід графа, а в її околі не було білих вершин, то він забарвлює цю вершину в чорний колір та зупиняється. Алгоритм закінчує роботу, коли червоний та жовтий шляхи стають порожніми, а всі вершини чорними. Виконуючи обхід графа $G$ агенти створюють нумерацію відвіданих вершин. Вперше відвідавши вершину агент $A$ забарвлює їх у червоний колір (агент $B$ – у жовтий колір), записує в пам'ять вершини відповідний номер (отриманий від агента-експериментатора), рівний значенню змінної $ct\_A$ ($ct\_B$ для агента $B$). Розпізнавання графа $G$ відбувається на основі створеної агентами-дослідниками нумерації шляхом побудови йому ізоморфного графа $H$.

Розглянемо алгоритм роботи агента-експериментатора:

*Вхід*: списки повідомлень $M$ та $N$ від агентів-дослідників.

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

*Дані* : $V_H, E_H$ – списки вершин і ребер графа $H$, ізоморфного графу $G$.

$ct\_A$, $ct\_B$ – лічильники, що використовуються для визначення числа відвіданих вершин агентами $A$ і $B$ відповідно. $STOP\_A, STOP\_B$ – змінні, що використовуються агентами $A$ і $B$ відповідно, для сигналізації агенту-експериментатору про завершення розпізнавання «своєї» області відповідним агентом-дослідником. $mr\_A$ – змінна, яка може набувати двох значень: «1» або «0». Значення «1» дозволяє агенту $A$ розпізнавати перешийки, у будь-якому випадку. Значення «0» забороняє агенту $A$ розпізнавати ті перешийки (якщо вони існують), у дальній вершині яких знаодиться агент $B$. $mr\_B$ – змінна, яка може набувати двох значень «1» або «0». Значення «1» дозволяє агенту $B$ розпізнавати перешийки у будь-якому випадку. Значення «0» забороняє агенту $B$ розпізнавання перешийків, якщо у дальній вершині одного з них знаходиться агент $A$. $r(1), r(2), ..., r(t)$ – список номерів вершин червоного шляху, де $t$ – довжина цього списку. $y(1), y(2), ..., y(p)$ список номерів вершин жовтого шляху, де $p$ – довжина цього списку. *Mes* – змінна в якій зберігається поточне повідомлення.

1. $ct\_A := 1, ct\_B := 1, M := \varnothing, N := \varnothing, E_H := \varnothing, STOP\_A := 0, STOP\_B := 0,$
$mr\_A := 0, mr\_B := 0, t := 1, p := 1, r(t) := ct\_A, y(p) := ct\_B,\ V_H := \{A[1], B[1]\}$;
2. *while* $(STOP\_A = 0) or (STOP\_B = 0)$ *do*
3. *if* $M \neq \varnothing$ *then do*
4. прочитати у *Mes* повідомлення та видалити його з черги $M$;
5. *LIST_PROCES_A*();
6. *end do*;
7. *if* $N \neq \varnothing$ *then do*
8. прочитати у *Mes* повідомлення та видалити його з черги $N$;
9. *LIST_PROCES_B*();
10. *end do*;
11. *end do*;
12. друк $V_H, E_H$.

Розглянемо процедури, які використовуються в алгоритмі.

*LIST_PROCES_A*():
1. *if* $Mes = "ISTHM\_A(x)"$ *then* $ISTHM\_A(x)$;
2. *if* $Mes = "INV\_A(x)"$ *then* $INV\_A(x)$;
3. *if* $Mes = "FORWARD\_A"$ *then* *FORWARD_A*();
4. *if* $Mes = "ASSIGN\_A"$ *then* *ASSIGN_A*();
5. *if* $Mes = "BACK\_A"$ *then* *BACK_A*();
6. *if* $Mes = "STOP\_A"$ *then* *STOP_A*().

$ISTHM\_A(x)$ : *if* $(B[x], A[r(t)]) \notin E_H$ *then* $E_H := E_H \cup \{(A[r(t)], B[x])\}$.

$INV\_A(x)$ : $E_H := E_H \cup \{(A[r(t)], A[x])\}$.

*FORWARD_A*(): $ct\_A := ct\_A + 1$; $t := t + 1$; $r(t) := ct\_A$; $V_H := V_H \cup \{A[ct\_A]\}$;
$E_H := E_H \cup \{(A[r(t-1)], A[r(t)])\}$; $mr\_B := 0$.

*ASSIGN_A*() : $mr\_B := 1$.

*BACK_A*(): зі списку $r(1), ..., r(t)$ видаляється елемент $r(t)$; $t := t - 1$.

*STOP_A*(): $STOP\_A := 1$.

Процедури роботи зі списком повідомлень від агента $B$, які не розглянуті нижче, аналогічні до процедур роботи зі списком повідомлень від агента $A$.

$LIST\_PROCES\_B()$:
1. $if\ Mes = "COME\_BACK\_B"\ then\ COME\_BACK\_B\ ()$;
2. $if\ Mes = "ISTHM\_B(k)"\ then\ ISTHM\_B(k)$;
3. $if\ Mes = "INV\_B(k)"\ then\ INV\_B(k)$;
4. $if\ Mes = "FORWARD\_B"\ then\ FORWARD\_B()$;
5. $if\ Mes = "BACK\_B"\ then\ BACK\_B\ ()$;
6. $if\ Mes = "STOP\_B"\ then\ STOP\_B()$.

$COME\_BACK\_B\ ()$: $E_H := E_H \setminus \{(y(p-1), y(p))\}$; $V_H := V_H \setminus \{B[ct\_B]\}$; $ct\_B := ct\_B - 1$; $p := p - 1$; $y(p) := ct\_B$.

$ISTHM\_B(k)$: $if\ (A[k], B[y(p)]) \notin E_H\ then\ E_H := E_H \cup \{(B[y(p)], A[k])\}$; $mr\_A := 1$.

$FORWARD\_B()$: $ct\_B := ct\_B + 1$; $p := p + 1$; $y(p) := ct\_B$; $V_H := V_H \cup \{B[ct\_B]\}$; $E_H := E_H \cup \{(B[y(p-1)], B[y(p)])\}$; $mr\_A := 0$.

$BACK\_B\ ()$: зі списку $y(1),...,y(p)$ видаляється $y(p)$; $p := p - 1$; $mr\_A := 1$.

Розглянемо властивості наведеного алгоритму розпізнавання.

**Теорема 1.** *Три агенти, виконавши алгоритм розпізнавання на графі $G$, розпізнають досліджуваний граф з точністю до ізоморфізму.*

*Доведення.* Процедури $FORWARD\_A()$ і $FORWARD\_B()$ виконуються агентом-експериментатором, під час відвідування агентами-дослідниками білих вершин досліджуваного графа $G$. Цими процедурами створюється по одній новій вершині графа $H$. При одночасному попаданні агентів $A$ і $B$ в одну білу вершину процедурами $FORWARD\_A()$ і $FORWARD\_B()$ буде створено дві нові вершини графа $H$. Для того щоб не допустити подібне дублювання вершин, на наступному кроці агент $B$, процедурою $COME\_BACK\_B\ ()$, видалить вершину, додану ним на попередньому кроці. Таким чином, виконання описаного алгоритму індукує відображення $\varphi: V_G \to V_H$. Причому $\varphi(v) = t$ (коли вершина $v$ пофарбована в червоний колір і $t = ct\_A$) і $\varphi(s) = p$ (коли вершина $s$ пофарбована в жовтий колір і $p = ct\_B$). Зазначене відображення $\varphi$ є біекцією, оскільки у зв'язному графі $G$ всі вершини можна досягти з початкових вершин. При виконанні процедури $FORWARD\_A()$ і $FORWARD\_B()$ агент-експериментатор розпізнає ребро $(v,u)$, що належить дереву і так нумерує вершину $u$, що ребру $(v,u)$ однозначно відповідає ребро $(\varphi(v), \varphi(u))$ графа $H$. При виконанні процедур $INV\_A(x)$ або $INV\_B(k)$ агент-експериментатор розпізнає зворотні ребра $(v,u)$ графа $G$ і ставить їм в однозначну відповідність ребра $(\varphi(v), \varphi(u))$ графа $H$. При виконанні $ISTHM\_A(x)$ або $INV\_B(k)$ агент-експериментатор розпізнає відповідно перешийки $(v,u)$ або зворотні ребра $(v,u)$ графа $G$ і ставить їм в однозначну відповідність ребра $(\varphi(v), \varphi(u))$ графа $H$. Отже, $\varphi$ є ізоморфізмом графа $G$ на граф $H$. *Що й треба було довести.*

**Теорема 2.** *Часова, ємнісна та комунікаційна складності алгоритму розпізнавання дорівнюють $O(n^2)$, а число переходів по ребрам, що здійснюють агенти дослідники оцінюється як $O(n)$. При цьому алгоритм використовує 3 фарби.*

*Доведення.* Підрахуємо часову, ємнісну та комунікаційну складності алгоритму. З опису випливає, що на кожному кроці алгоритму червоний (жовтий) шлях – це простий шлях, що сполучає початкову вершину $v$ ($s$ – для агента $B$) з номером $\varphi(v) = 1$ $(\varphi(s) = 1)$ з вершиною $u$ $(z)$ з номером $\varphi(u) = ct\_A$ $(\varphi(z) = ct\_B)$. Отже, загальна довжина червоного та жовтого шляху не перевищує $n$.

При одноразовому виконанні процедур зі звичайного режиму роботи агент-дослідник проходить одне ребро, або ж, не виконуючи переміщень, відправляє одне повідомлення на що також витрачається один хід. При одноразовому виконанні процедур з режиму розпізнавання зворотних ребер агенти-дослідники розпізнають не більше $n-2$ зворотних ребер, на що витрачають витрачають не більше $n-1$ кроків (зчитування номерів дальніх вершин зворотних ребер та один хід на відправку повідомлення). При одноразовому виконанні процедури з режиму розпізнавання перешийків агенти-дослідники розпізнають не більше $n-2$ перешийки, на що так само йде не більше $n-1$ кроків (зчитування номерів дальніх вершин перешийків та один хід на відправку повідомлення). При одночасному попаданні двох агентів-дослідників в одну білу вершину агент $A$ не змінює режим роботи, а агент $B$ витрачає один хід на повернення в свою область. При підрахунку часової складності алгоритму вважатимемо, що ініціалізація алгоритму, аналіз околу $Q(v)$ робочої вершини та вибір однієї з можливих процедур займають певну постійну кількість одиниць часу. Також будемо вважати, що вибір ребер, прохід по них агента-дослідника та обробка повідомлень відправлених на даному етапі здійснюється за одну одиницю часу. Тоді часова складність алгоритму визначається наступним чином. Процедури зі звичайного режиму роботи виконуються не більше ніж $3 \times (n-2)$ раз, загальний час їх виконання оцінюється як $O(n)$. Процедури з режиму розпізнавання зворотних ребер виконуються не більше $n \times (n-1)$ раз, тобто загальний час їх виконання оцінюється як $O(n^2)$. Процедури з режиму розпізнавання перешийків виконуються не більше $n\times(n-1)$ раз, тобто загальний час їх виконання оцінюється як $O(n^2)$. Час простою агентів в очікуванні оцінюється як $O(n^2)$. Отже, сумарна часова складність $T(n)$ алгоритму задовольняє наступному співвідношенню: $T(n)=O(n^2)$.

Таким чином, верхня оцінка числа переходів по ребрах, що здійснюються агентом-дослідником оцінюється зверху як $\mathrm{O}(\mathrm{n})$.

Ємнісна складність $S(n)$ алгоритму визначається потужностями списків $V_H, E_H, r(1)...r(t), y(1)...y(p)$, які визначаються величинами $O(n), O(n^2)$, $O(n), O(n)$ відповідно. Отже: $S(n)=O(n^2)$.

Комунікаційна складність $K(n)$ алгоритму визначається кількістю повідомлень, якими необхідно обмінятися агентам, для розпізнавання графа. Працюючи в звичайному режимі роботи, агенти-дослідники відправляють по одному повідомленню на кожному кроці, тобто передана інформація у цьому режимі оцінюється як $O(n)$. Агент-експериментатор при цьому передасть обсяг інформації, що також оцінюється як $O(n)$. Загальна кількість переданих повідомлень при роботі агентів в режимі розпізнавання зворотних ребер та в режимі розпізнавання перешийків оцінюється як $2\times O(n^2)$. Отже, сумарна комунікаційна складність $K(n)$ алгоритму задовольняє наступному співвідношенню: $K(n)=O(n^2)$. *Що й треба було довести.*

**Висновки**

У роботі запропоновано новий алгоритм розпізнавання скінчених неорієнтованих графів часової складності, ємнісної та комунікаційної складностей рівними $O(n^2)$ та верхньою оцінкою числа переходів по ребрам, що здійснюють агенти-дослідники рівною $O(n)$. Агенти-дослідники мають скінчену пам'ять і використовують по дві фарби кожен (всього три фарби). Результати даного дослідження планується масштабувати на більшу кількість агентів для отримання більш ефективних алгоритмів розпізнавання графів.

**Список використаних джерел:**


1. Albers S., Henzinger M. R. Exploring unknown environments. *SIAM Journal on Computing*. 2000. 29 (4). P. 1164–1188.
2. 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
3. Cormen Ò., Leiserson Ch., Rivest R., Stein C. Introduction to Algorithms Cambridge, 2009. 1292 p.
4. Das S., Flocchini 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.
5. 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
6. 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
7. Dopp K. Automaten in labirinthen. *Electronische Informationsverarbeitung und Kybernetik*. 1971. V.7, № 2. P. 79–94.
8. Dudek G., Jenkin M. Computational principles of mobile robotics. *Cambridge Univ. press*. 2000. 280 p.
9. Dudek G., Jenkin M., Milios E., Wilkes D. Map validation in a graphlike world. *Proceedings of the 13th International Joint Conference on Artifical Intelligence.* San Fransisco: Morgan Kaufmann Publishers Inc., 1993. P. 1648–1653.
10. Dudek G., Jenkin M., Milios E., Wilkes D. Topological exploration with multiple robots. Robotics with Application (ISORA)*: Proc. 7th International Symposium*. Alaska, 1998
11. Feng Xiao, Long Wang, Jie Chen, Yanping Gao. Finite-time formation control for multi-agent systems. *Automatica.* V 45, Issue 11. 2009. P. 2605–2611. https://doi.org/10.1016/j.automatica.2009.07.012
12. 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
13. Piña-García C., Rechy Ericka, Garcia-Vega Virginia. Using an Alternative Model in a Complex Environment for Nanorobotics Navigation. *Proceedings of the 16th International Conference on Computing.* Mexico City, Mexico, 2007. URL: http://magno-congreso.cic.ipn.mx/cd-2007/IEEE/paper_103.pdf
14. Selden M., Zhou J., Campos F., Lambert N., Drew D. and Pister K. S. J. BotNet: A Simulator for Studying the Effects of Accurate Communication Models on Multi-Agent and Swarm Control. *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, Cambridge, United Kingdom, 2021, pp. 101–109, doi: 10.1109/MRS50823.2021.9620611.
15. Shannon C.E. Presentation of a maze-solving machine. *Cybernetics Trans, of the 8 th Conf. of the JosiahMacy Jr. Found* / Editor: H. Foerster. 1951. P. 173–180.
16. Stepkin A. Using a Collective of Agents for Exploration of Undirected Graphs. *Cybernetics and Systems Analysis.* V.51, 2. 2015 223–233. https://doi.org/10.1007/s10559-015-9715-z
17. Zhang C. Parallelizing Depth-First Search for Robotic Graph Exploration. Harvard College, Cambridge, Massachusetts. 2010.
18. 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/app122312337ц