

Інститут прикладної математики і механіки НАН України  
Кафедра теоретичної кібернетики  
Київського національного університету імені Тараса Шевченка

**АКТУАЛЬНІ ПРОБЛЕМИ ТЕОРІЇ КЕРУЮЧИХ СИСТЕМ  
У КОМП'ЮТЕРНИХ НАУКАХ**

**ACTUAL PROBLEMS OF CONTROL-SYSTEM THEORY  
IN COMPUTER SCIENCES**

**АПТКС'2021**

**Праці конференції  
Proceeding**

**21-24 грудня 2021 року  
Слов'янськ**

**УДК: 004+519**

Друкується за рішенням Вченої ради  
Інституту прикладної математики і механіки НАН України  
(протокол № 8 від 28.12.2021 р.)

**Актуальні проблеми теорії керуючих систем у комп'ютерних науках:** праці науково-практичної конференції, м. Слов'янськ, 21-24 грудня 2021 р. / за заг. ред.: І.І. Скрипніка, Ю.В. Крака, О.С. Сенченка та ін. Слов'янськ: Видавництво Б.І. Маторіна, 2021. 128 с.

У збірнику представлено найкращі доповіді науково-практичної конференції «Актуальні проблеми теорії керуючих систем у комп'ютерних науках» (АПТКС'2021), яка проходила в Слов'янську з 21 по 24 грудня 2021 р.

**УДК: 004+519**  
**©ІПММ НАНУ, 2021**

Науково-практична конференція  
**«АКТУАЛЬНІ ПРОБЛЕМИ ТЕОРІЇ КЕРУЮЧИХ СИСТЕМ  
У КОМП'ЮТЕРНИХ НАУКАХ»**

(Україна, Слов'янськ, 21-24 грудня 2021 року)

Інститут прикладної математики і механіки НАН України, Кафедра теоретичної кібернетики Київського національного університету імені Тараса Шевченка з 21 по 24 грудня 2021 року проводили науково-практичну конференцію «Актуальні проблеми теорії керуючих систем у комп'ютерних науках» (АПТКС'2021).

**НАУКОВІ НАПРЯМИ КОНФЕРЕНЦІЇ**

- ✓ Теорія графів, теорія керуючих систем та теорія автоматів
- ✓ Теорія програмування, валідація, верифікація і тестування програмних систем
- ✓ Захист та кодування інформації
- ✓ Інформаційні технології в освіті
- ✓ Бази даних, Web-програмування та електронна комерція

**ПРОГРАМНИЙ КОМІТЕТ**

- ◇ ч-к. НАН України, д. ф-м. н., директор Інституту прикладної математики і механіки НАН України **І.І. Скрипник, голова;**
- ◇ ч-к. НАН України, д. ф-м. н., зав. кафедри теоретичної кібернетики Київського національного університету імені Тараса Шевченка, п. н. с. Інституту кібернетики імені В.М. Глушкова НАН України **Ю.В. Крак, заступник голови;**
- ◇ ч-к. НАН України, д. ф-м. н., зав. відділу № 235 Інституту кібернетики імені В.М. Глушкова НАН України **А.М. Гупал;**
- ◇ д. ф-м. н., проф. кафедри інформатики Національного університету «Києво-Могилянська Академія» **М.М. Глибовець;**
- ◇ д. т. н., зав. відділенням другої вищої та післядипломної освіти Інституту прикладного системного аналізу НТУУ «Київський політехнічний інститут імені Ігоря Сікорського» **Ю.П. Зайченко;**
- ◇ д. т. н., проф. кафедри інформаційних систем та мереж Національного університету «Львівська політехніка» **В.В. Пасічник;**
- ◇ к. ф-м. н., с. н. с. відділу теорії керуючих систем Інституту прикладної математики і механіки НАН України **О.С. Сенченко;**
- ◇ д. т. н., проф. кафедри проектного навчання в інформаційних технологіях Інституту штучного інтелекту та робототехніки Державного університету «Одеська політехніка» **С.А. Устенко;**
- ◇ д. т. н., зав. кафедри комп'ютерних систем, мереж і кібербезпеки Національного аерокосмічного університету імені М.Є. Жуковського «Харківський авіаційний інститут» **В.С. Харченко;**
- ◇ д. ф-м. н., проректор з науково-педагогічної роботи Донбаського державного педагогічного університету **С.О. Чайченко;**
- ◇ д. т. н., зав. кафедри комп'ютерних інформаційних технологій Дніпровського національного університету залізничного транспорту імені академіка В. Лазаряна **В.І. Шинкаренко.**

**ОРГАНІЗАЦІЙНИЙ КОМІТЕТ**

- ◇ О.В. Несмелова, голова
- ◇ С.В. Сапунов, заст. голови
- ◇ О.С. Сенченко, секретар

З 21 по 24 грудня 2021 року в Слов'янську проходила науково-практична конференція «Актуальні проблеми теорії керуючих систем у комп'ютерних науках» (АПТКС'2021). До участі у конференції надійшло понад 40 праць з різних напрямків комп'ютерних наук та теорії керуючих систем. З цих робіт програмним комітетом конференції було рекомендовано до друку в цьому збірнику 24 роботи. Автори цих робіт (загалом 47 авторів) представляють провідні наукові установи та заклади вищої освіти України: Київський національний університет імені Тараса Шевченка (13 авторів), Інститут прикладної математики і механіки НАН України (7 авторів), Інститут кібернетики імені В.М. Глушкова НАН України (5 авторів), Донбаську державну машинобудівну академію (5 авторів), Харківський національний університет радіоелектроніки (3 автори), Національний університет «Києво-Могилянська Академія» (2 автори), Національний авіаційний університет (2 автори), Вінницький державний університет імені Михайла Коцюбинського (2 автори), Хмельницький національний університет (2 автори), Національну металургійну академію України (2 автори), та ін. Серед авторів доповідей 1 член-кореспондент НАН України, 11 докторів наук та 20 кандидатів наук.

У зв'язку з запровадженням обмежувальних протиепідемічних заходів з метою запобігання поширенню COVID-19, доповіді конференції проводилися в дистанційному режимі на платформі Zoom, у кожен з чотирьох робочих днів конференції в обговоренні доповідей приймали участь від 20 до 40 учасників конференції. Учасники конференції відмічали достатньо високий рівень доповідей, а також цікаве їх обговорення; від більшості учасників було одержано попередню згоду на участь у нашій конференції й у наступному 2022 році. Організаційний комітет конференції бажає міцного здоров'я та творчих успіхів всім учасникам конференції та обіцяє докласти всіх зусиль для того, щоб наступна конференція у 2022 році також пройшла на високому науковому та організаційному рівні, а якщо дозволить епідеміологічна ситуація, пов'язана з розповсюдженням COVID-19, то і в традиційному очному форматі.

З повагою, організаційний комітет конференції

## ЗМІСТ

|                                                                                                                                                                      |     |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| <b>Баранюкова І.С., Жоголева Н.В.</b> Алгоритм визначення параметрів мережі нелінійних осциляторів.....                                                              | 6   |
| <b>Буренко О.І., Камінська Л.П.</b> Організація дистанційного навчання в закладі освіти засобами інструментів Google.....                                            | 11  |
| <b>Грудкіна Н.С., Васильєва Л.В., Єрмакова К.Ю.</b> Розробка Web-сервісу оцінки рівня конкурентноспроможності страхових послуг.....                                  | 16  |
| <b>Гулаєва Н.М., Глибовець М.М.</b> Генетичні алгоритми і чисельні методи оптимізації.....                                                                           | 20  |
| <b>Денісова О.О.</b> Особливості організації платформ клієнтських даних.....                                                                                         | 27  |
| <b>Духновська К.К.</b> Пошук закономірностей для класів текстових документів.....                                                                                    | 32  |
| <b>Іллічева Л.М., Авдєєва Т.В.</b> До моделювання відсоткових ставок дифузійними рівняннями.....                                                                     | 35  |
| <b>Канарська І.С.</b> Алгоритми реалізації теоретико-множинних операцій у табличних алгебрах.....                                                                    | 39  |
| <b>Ключко О.В., Березов Д.О.</b> Інноваційні навички як компетентнісний шлях розкриття потенціалу освітньої робототехніки.....                                       | 45  |
| <b>Коротчиков О.Д., Селівьорстова Т.В.</b> Методологія розробки та програмна реалізація підсистеми для ранжування рекламного контенту .....                          | 50  |
| <b>Крак Ю.В., Пашко А.О., Стеля О.Б.</b> Послідовний аналіз ЕКГ-сигналів.....                                                                                        | 54  |
| <b>Кривий С.Л.</b> Про процедуру розв'язуваності для нечіткої пропозиційної логіки....                                                                               | 59  |
| <b>Кузнєцов В.О., Кондратюк С.С., Крак Ю.В., Петрович В.М., Самойлов О.М.</b> До розробки людино-машинних інтерфейсів на основі жестової інформації.....             | 66  |
| <b>Кузнєцов В.О., Крак Ю.В., Петрович В.М., Самойлов О.М., Куляс А.І.</b> Оцінка надійності подання знань із використанням оптимального шляху між властивостями..... | 68  |
| <b>Манзюк Е.А., Крак Ю.В., Бармак О.В.</b> Про розробку методів машинного навчання на основі довіри штучному інтелекту.....                                          | 70  |
| <b>Марченко О.О., Насіров Е.М.</b> Блочно-діагональний підхід до невід'єсної факторизації розріджених лінгвістичних матриць надвеликої розмірності.....              | 72  |
| <b>Нікітенко О.М., Дейнеко Ж.В., Грищенко Т.Б.</b> Дослідження переваг застосування графіків у системі $IT_{EX}$ при оформленні наукових праць.....                  | 79  |
| <b>Парамонова К.О., Шевченко Н.Ю.</b> Модуль генерації елементів стегографічної системи для захисту інформації при передачі даних відкритими каналами зв'язку.....   | 85  |
| <b>Сапунов С.В., Полшенцев Б.В.</b> Розпізнавання розмітки вершин позначеного графа квадратної решітки.....                                                          | 90  |
| <b>Семенюта М.Ф.</b> Про граціозну та різницеву квадратну розмітку графів.....                                                                                       | 99  |
| <b>Сенченко О.С.</b> Властивості теоретико-множинних операцій у табличних алгебрах.....                                                                              | 102 |
| <b>Сенченко О.С., Середа О.А., Притула М.І.</b> Визначальна пара для детермінованих графів.....                                                                      | 112 |
| <b>Хусаїнов Д.Я., Шатирко А.В., Бичков О.С., Шакотько Т.І.</b> Стійкість та збіжність в моделях керуючих нейродинамічних систем.....                                 | 121 |
| <b>Яременко С.В., Крак Ю.В.</b> Алгоритми створення інтерактивного тренажера для навчання та тренування навичок стрільби.....                                        | 127 |

**Баранюкова І.С., Жоголева Н.В.**

<sup>1</sup> аспірант відділу прикладної механіки Інституту прикладної математики і механіки НАНУ, Слов'янськ

<sup>2</sup> к. ф-м. н., науковий співробітник відділу прикладної механіки Інституту прикладної математики і механіки НАНУ, Слов'янськ

*e-mail:* <sup>1</sup> irina63m28@gmail.com, <sup>2</sup> zhogoleva.nadia@gmail.com

## **АЛГОРИТМ ВИЗНАЧЕННЯ ПАРАМЕТРІВ МЕРЕЖІ НЕЛІНІЙНИХ ОСЦИЛЯТОРІВ**

Розглянуто задачу спостереження стану та ідентифікації параметрів коливальної системи, що складається із зв'язаних осциляторів Ван дер Поля. Використовується метод інваріантних відношень для оцінки параметрів. Такий підхід базується на динамічному розширенні вихідної системи та синтезі відповідних інваріантних співвідношень, з яких невідомі можуть бути виражені як функція відомих величин на траєкторіях розширеної системи під час спостережуваного руху. На першому кроці вирішуються задача ідентифікації для одного автономного генератора Ван дер Поля, далі отримані результати поширюються на систему пов'язаних осциляторів. Результати моделювання підтверджують ефективність запропонованої схеми розробки нелінійного спостерігача та ідентифікатора для мережі осциляторів.

**Ключові слова:** ідентифікація, нелінійні коливання, пов'язані осцилятори ван дер Поля, інваріантні співвідношення, асимптотичні оцінки.

## **Вступ**

У багатьох прикладних додатках для опису складних нелінійних коливальних процесів використовується модель, яка складається з одного або декількох пов'язаних між собою осциляторів Ван дер Поля або деяких їх модифікацій [1]. В даній роботі побудовано нелінійний ідентифікатор, який формує асимптотичні оцінки невідомих для одного автономного осцилятора, а далі отриманий результат поширено на систему пов'язаних осциляторів. Використано метод інваріантних співвідношень [2], який призначено для пошуку частинних розв'язків (залежностей між змінними) в задачах динаміки твердого тіла з нерухомою точкою.

## Визначення характеристик осцилятора ван дер Поля. Синтез додаткових співвідношень

Розглянемо рівняння Ван дер Поля у вигляді:

$$\begin{aligned}\dot{x}_1 &= x_2, \\ \dot{x}_2 &= \mu(1 - x_1^2)x_2 - \omega^2 x_1.\end{aligned}\tag{1}$$

Метою є ідентифікація параметрів  $\mu$  і  $\omega$  за відомою інформацією про рух. Такою інформацією є вихід – функція часу  $y(t) = (x_1(t), x_2(t))$ , а також ті величини, які можуть бути отримані з використанням значень виходу. Далі відомим будемо вважати будь-яке рішення задачі Коші  $\zeta(t, \zeta_0)$

$$\dot{\zeta} = U(\zeta, x_1(t), x_2(t)), \quad \zeta(0) = \zeta_0 \in R^p,\tag{2}$$

для будь-якої системи диференціальних рівнянь розмірності  $p \geq 1$ , праві частини  $U(\zeta, x_1(t), x_2(t))$  якої задовольняють достатнім умовам теорему існування і єдиності рішень для  $t \in [0, \infty)$

**Задача 1.** Знайти асимптотично точні оцінки параметрів  $\mu$  і  $\omega$  системи (1) за відомими значеннями виходу  $y(t) = (x_1(t), x_2(t))$ .

Для розв'язку використовуємо метод синтезу інваріантних співвідношень спеціального виду [1]. Ідея методу полягає в динамічному розширенні вихідної системи диференціальних рівнянь (1) додатковими рівняннями (2), де  $p$  дорівнює 2 – числу невідомих (константи  $\mu$  і  $\omega$ ). При цьому праві частини допоміжної системи  $U(\zeta, x_1, x_2)$  підбираються таким чином, щоб отримана розширена система диференціальних рівнянь (1), (2) допускала на своїх рішеннях інваріантні співвідношення

$$G_i(x_1(t), x_2(t), \zeta(t), \mu, \omega) = 0, \quad i = 1, 2.\tag{3}$$

Співвідношення виду (3) будемо шукати у вигляді

$$\begin{aligned}G_1 &= \mu - \zeta_1(t) - \Phi(x_1(t), x_2(t)) = 0, \\ G_2 &= \omega^2 - \zeta_2(t) - \Psi(x_1(t), x_2(t)) = 0,\end{aligned}\tag{4}$$

де  $\zeta_i(t)$ ,  $i = 1, 2$  – розв'язки допоміжного диференціального рівняння (2), а функції  $\Phi(x_1, x_2)$ ,  $\Psi(x_1, x_2)$ ,  $U(\zeta, x_1, x_2)$  безперервно диференціюються по своїм аргументам. Якщо ці функції будуть обрані так, що співвідношення (4) виявляться інваріантними на даному рішенні системи (1), (2), то тоді невідомі параметри  $\mu$  і  $\omega$  можуть бути знайдені безпосередньо з цих рівностей.

**Твердження 1.** Для будь-яких диференційованих за своїми аргументами функцій  $\Phi(x_1, x_2)$ ,  $\Psi(x_1, x_2)$  можна підібрати праву частину в системі диференціальних рівнянь (2) так, що рівності (4) будуть виконуватись тотожно на деяких рішеннях розширеної системи диференціальних рівнянь (1), (2).

**Твердження 2.** Вибір вільних функцій  $\Phi(x_1, x_2)$ ,  $\Psi(x_1, x_2)$ , у вигляді

$$\Phi(x_1, x_2) = (1 - x_1^2) \frac{x_2^2}{2}, \quad \Psi(x_1, x_2) = -x_1 x_2 \quad (5)$$

гарантує асимптотичну стійкість тривіального рішення системи

$$\begin{aligned} \dot{\varepsilon}_1 &= x_2(1 - x_1^2)[-x_2(1 - x_1^2)\varepsilon_1 + x_1\varepsilon_2], \\ \dot{\varepsilon}_2 &= -x_1[-x_2(1 - x_1^2)\varepsilon_1 + x_1\varepsilon_2]. \end{aligned} \quad (6)$$

Таким чином, співвідношення (4) і допоміжна система диференціальних рівнянь формує для системи (1) асимптотичний ідентифікатор.

**Твердження 3.** Для будь-якого нетривіального рішення  $x_1(t)$ ,  $x_2(t)$  системи (1) і будь-якого початкового значення  $\zeta_1(0)$ ,  $\zeta_2(0)$  в задачі Коші для допоміжної системи диференціальних рівнянь

$$\begin{aligned} \dot{\zeta}_1 &= x_1 x_2^3 + x_1 x_2^2 \{-x_1(\zeta_2 - x_1 x_2) + [\zeta_1 + (1 - x_1^2) \frac{x_2^2}{2}](1 - x_1^2)x_2\}, \\ \dot{\zeta}_2 &= x_2^2 + x_1 \{-x_1(\zeta_2 - x_1 x_2) + [\zeta_1 + (1 - x_1^2) \frac{x_2^2}{2}](1 - x_1^2)x_2\}. \end{aligned} \quad (7)$$

формули

$$\hat{\mu} = \zeta_1(t) + (1 - x_1^2(t)) \frac{x_2^2(t)}{2}, \quad \hat{\omega}^2 = \zeta_2(t) - x_1(t)x_2(t) \quad (8)$$

визначають асимптотичні оцінки параметрів  $\mu$  і  $\omega^2$  відповідно.

## Система пов'язаних осциляторів

Розглянемо систему, що складається з  $n$  пов'язаних між собою неідентичних осциляторів ван дер Поля

$$\begin{aligned} \dot{x}_{i1} &= x_{i2}, \\ \dot{x}_{i2} &= \mu_i(1 - x_{i1}^2)x_{i2} - \omega_i^2 x_{i1} + F_i(t, x_{i1}, \dots, x_{n2}), i = \overline{1, n}. \end{aligned} \quad (9)$$

Тут змінні  $x_{i1}$  означають зміщення від власного положення рівноваги  $i$ -го осцилятора,  $x_{i2}$  – швидкість відповідного зміщення, функції

$F_i(t, x_{11}, \dots, x_{n2})$  формалізують зовнішній вплив і вплив зв'язків в мережі на динаміку  $i$ -го осцилятора,  $i = \overline{1, n}$ . У задачах наближеного моделювання коливань складних об'єктів за допомогою моделей виду (9) ці функції підбираються з різних міркувань і можуть не мати фізичного або механічного сенсу, описуючи, наприклад, односторонні зв'язки.

**Задача 2.** Знайти асимптотично точні оцінки параметрів  $\mu_i, \omega_i^2, i = \overline{1, n}$  системи (9) за відомими значеннями фазового вектору  $x(t) = (x_{11}(t), x_{12}(t), \dots, x_{n2}(t))$ .

**Твердження 4.** Для будь-якого нетривіального рішення  $x(t)$  системи (9) і будь-якого початкового значення  $\zeta(0)$  в задачі Коші для допоміжної системи диференціальних рівнянь

$$\begin{aligned}\dot{\zeta}_{i1} &= x_{i1}x_{i2}^3 + x_{i1}x_{i2}^2\{-x_{i1}(\zeta_{i2} - x_{i1}x_{i2}) + [\zeta_{i1} + (1 - x_{i1}^2)\frac{x_{i2}^2}{2}](1 - x_{i1}^2)x_{i2} + F_i\}, \\ \dot{\zeta}_{i2} &= x_{i2}^2 + x_{i1}\{-x_{i1}(\zeta_{i2} - x_{i1}x_{i2}) + [\zeta_{i1} + (1 - x_{i1}^2)\frac{x_{i2}^2}{2}](1 - x_{i1}^2)x_{i2} + F_i\}.\end{aligned}\quad (10)$$

формули

$$\hat{\mu}_i = \zeta_{i1}(t) + (1 - x_{i1}^2(t))\frac{x_{i2}^2(t)}{2}, \hat{\omega}_i^2 = \zeta_{i2}(t) - x_{i1}(t)x_{i2}(t) \quad (11)$$

визначають асимптотичні оцінки параметрів  $\mu_i, \omega_i^2, i = \overline{1, n}$ .

## Числове моделювання

Числове моделювання здійснювалося для ланцюжка з трьох послідовно з'єднаних неідентичних осциляторів ван дер Поля

$$\begin{aligned}\dot{x}_{11} &= x_{12}, \\ \dot{x}_{12} &= \mu_1(1 - x_{11}^2)x_{12} - \omega_1^2x_{11} + F_1, \\ \dot{x}_{21} &= x_{22}, \\ \dot{x}_{22} &= \mu_2(1 - x_{21}^2)x_{22} - \omega_2^2x_{21} + F_2, \\ \dot{x}_{31} &= x_{32}, \\ \dot{x}_{32} &= \mu_3(1 - x_{31}^2)x_{32} - \omega_3^2x_{31} + F_3,\end{aligned}\quad (12)$$

зв'язаних за допомогою пружних і дисипативних зв'язків:

$$\begin{aligned}F_1 &= \nu(x_{11} - x_{21}) + k(x_{12} - x_{22}), \\ F_2 &= \nu(x_{21} - x_{11}) + \nu(x_{21} - x_{31}) + k(x_{22} - x_{12}) + k(x_{22} - x_{32}), \\ F_3 &= \nu(x_{31} - x_{21}) + k(x_{32} - x_{22}),\end{aligned}\quad (13)$$

де  $\nu$  – коефіцієнт жорсткості,  $k$  – коефіцієнт дисипативного зв'язку. Вихід системи – вектор  $y(t) = (x_1(t), x_2(t), x_3(t), x_4(t), x_5(t), x_6(t))$  отриманий в результаті числового розв'язку системи диференціальних рівнянь (11) у якій параметри  $\mu_1, \mu_2, \mu_3 = 0.0, 2.0, 5.0$ , частоти  $\omega_1^2, \omega_2^2, \omega_3^2 = 4.0, 9.0, 1.0$ , а параметри:  $\nu = 1.2, k = 0.3$ . В даному випадку:  $\zeta_{i1}(0) = -2.0; \zeta_{i2}(0) = -2.0, i = 1, 2, 3$ .

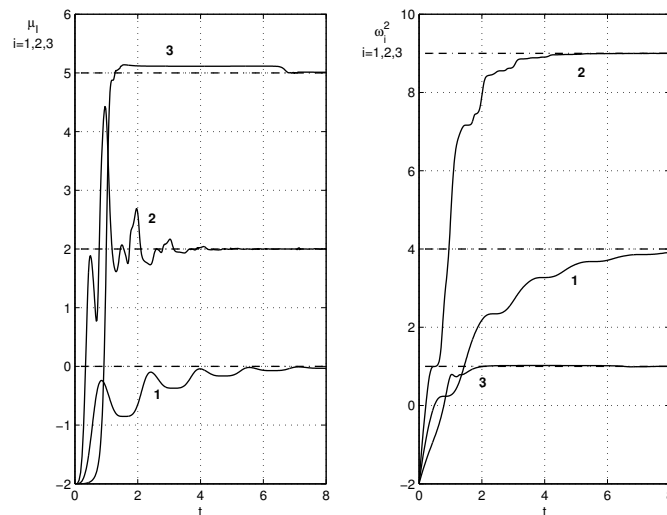


Рис. 1: Графіки функцій, що прямують до значень параметрів  $\mu$  (а) та  $\omega$  (б)

На рис 1.а під номерами  $i = 1, 2, 3$  зображено графіки функцій  $\zeta_{i1}(t) + (1 - x_{i1}^2(t)) \frac{x_{i2}^2(t)}{2}$ , які, відповідно до твердження 4, асимптотично прямують до шуканих значень параметрів  $\mu_1 = 0.0, \mu_2 = 2.0, \mu_3 = 5.0$ . На рис. 1.б, відповідно, зображено графіки функцій  $\zeta_{i2}(t) - x_{i1}(t)x_{i2}(t)$ , які сходяться до шуканих значень  $\omega_1^2 = 4.0, \omega_2^2 = 9.0, \omega_3^2 = 1.0$ . Результати моделювання підтверджують працездатність запропонованої схеми розв'язання задачі ідентифікації.

## Список використаних джерел

- [1] Кузнецов А.П., Селиверстова Е.С., Трубецков Д.И., Тюрюкина Л.В. Феномен уравнения ван дер Поля. *Известия вузов. Прикладная нелинейная динамика*. 2014. Т. 22, № 4. С. 3–42.
- [2] Харламов П.В. Об инвариантных соотношениях системы дифференциальных уравнений. *Механика твердого тела*. 1974. № 6. С. 15–24.
- [3] Жоголева Н.В., Щербак В.Ф. Синтез дополнительных соотношений в обратных задачах управления *Труды ИПММ НАН Украины*. 2015. Т. 29, С. 69–76.

**Буренко О.І., Камінська Л.П.**

<sup>1</sup> методист відділу цифрової освіти Донецького обласного інституту післядипломної педагогічної освіти, Краматорськ

<sup>2</sup> магістрант факультету інформаційно-комп'ютерних технологій Державного університету «Житомирська політехніка»

*e-mail:* <sup>1</sup> alex@ippo.dn.ua, <sup>2</sup> l.kaminskaya@icloud.com

## **ОРГАНІЗАЦІЯ ДИСТАНЦІЙНОГО НАВЧАННЯ В ЗАКЛАДІ ОСВІТИ ЗАСОБАМИ ІНСТРУМЕНТІВ GOOGLE**

У роботі розглянуто структуру курсів підвищення кваліфікації для вчителів загально-освітніх шкіл «Організація дистанційного навчання в закладі освіти засобами інструментів Google». Показано успішне застосування набутих на курсах умінь та навичок у педагогічному процесі.

**Ключові слова:** *курси підвищення кваліфікації, сервіси Google, дистанційна освіта.*

### **Вступ**

Донецький обласний інститут післядипломної педагогічної освіти впродовж багатьох років успішно проводить курси підвищення кваліфікації для вчителів за різними напрямками. Нажаль, для багатьох педагогів у більшості випадків курси підвищення кваліфікації сприймаються як не дуже корисний, але обов'язковий захід, необхідний для підтвердження педагогічної категорії чи її підвищення, або іншої матеріальної чи кар'єрної мотивації. Починаючи з весни 2020 року, в зв'язку з протиепідеміологічними заходами проти Covid-19, значна частина навчальних заходів повністю або частково перейшла на дистанційне навчання. Більшість вчителів які викладають дисципліни, що не пов'язані з інформаційно-комп'ютерними технологіями (особливо вчителі похилого віку), виявились не готовими для проведення занять у дистанційному форматі. У зв'язку з цим, у Донецькому обласному інституті було розроблено курси «Організація дистанційного навчання в закладі освіти засобами інструментів Google», які викликали підвищений інтерес серед вчителів Донецької області. Далі ми коротко розповімо про ці курси.

## Передумови розроблення курсів

Розвиток і широке застосування інформаційно-комунікаційних технологій (ІКТ) є глобальною тенденцією науково-технічного прогресу останніх десятиліть, що призвела до значних змін у багатьох сферах людської діяльності, особливо в системі освіти. Комп'ютерні технології не лише змінюють сформований навчально-виховний процес, але і роблять істотний вплив на професійну діяльність педагогів. Запровадження ІКТ в освіті є обов'язковою передумовою формування інформаційного суспільства в широкому розумінні.

Вироблення ключових компетентностей у процесі застосування ІКТ передбачає формування в учителя здатності орієнтуватися в інформаційному просторі, навчити його отримувати необхідну інформацію й оперувати нею відповідно до власних потреб і вимог сучасного високотехнологічного інформаційного суспільства.

У часи пандемії нового значення набуло дистанційне навчання, яке покликане сформувати в учасників освітнього процесу навички віртуальної колаборації, тобто вміння працювати продуктивно та активно взаємодіяти у віртуальній команді в спеціалізованому середовищі на основі ІКТ. Дистанційне навчання надає доступ до освітніх ресурсів через використання сучасних засобів електронних комунікацій, дає змогу навчати без відриву від основного виду діяльності, надавати освітні послуги особам із фізичними вадами, мешканцям віддалених районів, обдарованим учням тощо. Ці переваги успішно використовуються в Донецькій області, адже в цьому регіоні існує велика потреба в організації якісного процесу навчання з віддаленим доступом для учнів з тимчасово окупованих територій та забезпеченні реалізації їх права як громадян України на здобуття освіти.

Вимушене дистанційне навчання під час пандемії Covid-19 стало викликом для всіх учасників освітнього процесу: учителів, учнів та батьків. Організувати якісне навчання з використанням цифрових технологій, надихати й мотивувати учнів, давати раду технічним проблемам виявилось зовсім не просто. Наша держава впроваджувала дистанційне навчання та розробляла рекомендації для роботи в віддаленому режимі [1] паралельно з іншими європейськими країнами.

Зважаючи на це, у Донецькому обласному інституті післядипломної педагогічної освіти відділом дистанційного навчання було створено курси «Організація дистанційного навчання в закладі освіти засобами інструментів Google». Ці курси були впроваджені педагогами, які вже мають сертифікати тренерів Google.

Інструменти та сервіси Google є дуже поширеними серед учнів, ба-

тків та вчителів [2], вони підтримуються більшістю сучасних пристроїв, є зручними у використанні, один акаунт надає можливість працювати з електронною поштою, хмарним середовищем, спільно працювати з документами онлайн, створювати презентації та гугл-форми, розміщувати власне відео на Youtube, створювати сайти та блоги тощо. Цими курсами зацікавилися близько 2500 педагогів області, за 2021 рік в Донецькому обласному інституті було організовано понад 80 курсів підвищення кваліфікації за цією тематикою. Викладачі інституту відзначали надзвичайно високу мотивацію серед вчителів шкіл у засвоєнні знань, умінь та навичок, що формувалися на цих курсах. Програма підвищення кваліфікації розроблена на 30-годинний курс в очному або дистанційному форматі (на вибір).

## **Структура курсів та їх подальше використання**

Наразі в середовищі Google Workspace (G Suite for Education) [3] є 14 основних сервісів (рис. 1) та 62 додаткових. Перші є основною ланкою середовища та повністю задовольняють вимогам до дистанційного навчання та документообігу в закладі освіти. Зважаючи на це, а також на певні обмеження, пов'язані з кількістю навчальних годин, відведених проведення курсів, для розгляду в цих курсах було обрано такі теми:

1. Освітнє середовище Google Workspace.
2. Робота з Google-документами та диском. Спільний доступ.
3. Робота з Google-календарем. Організація синхронного навчання за допомогою Google-meet. Створення Google-форм.
4. Формування інфомедійної грамотності та навичок критичного сприймання інформації в учасників освітнього процесу як чинник розвитку ключових та професійних компетентностей.
5. Створення гугл-сайту.
6. Організація змішаного навчання за допомогою Google-класу.
7. Робота з сервісом Youtube. Ведення власного блогу.

Курси підвищення кваліфікації є корпоративними – проводяться для викладачів певного закладу в одному середовищі. Учителі одночасно вчаться, набувають навичок роботи в середовищі, створюють спільні документи. Такий підхід полегшує перехід всього закладу освіти на електронний документообіг засобами Google Workspace, прискорює впровадження сервісів у роботу, допомагає подолати психологічний бар'єр у педагогів при впровадженні нових інструментів та технологій в роботу закладу.



Рис. 1: Основні сервіси середовища Google Workspace.

Здобувачі отримують свідоцтва про підвищення кваліфікації, які не тільки надають можливість атестуватися в закладі освіти, а і згідно з наказом МОН № 466 від 25.04.13 року «Про затвердження Положення про дистанційне навчання» вести уроки в дистанційному форматі.

Цікавим моментом є те, що після навчання на курсах вчителі організовують дистанційне навчання у своєму закладі на основі освітнього середовища Google Workspace – набір безпечних додатків для співпраці й ефективної роботи в компанії будь-якого розміру, де пропонуються персональні електронні адреси для вчителів та учнів й такі інструменти спільної роботи, як Gmail, Календар, Meet, Chat, Диск, Документи, Таблиці, Презентації, Форми, Сайти тощо. У цьому контексті показовим є приклад Слов'янського педагогічного ліцею Слов'янської міської ради Донецької області, педагоги якого успішно пройшли наші курси, а потім, під керівництвом директора закладу Бабенко Оксани Анатоліївни та з технічним супроводом вчителя інформатики Іваненка Юрія Івановича, було створено освітнє середовище Google Workspace – сучасне освітнє середовище для організації дистанційного та змішаного навчання, ведення документообігу, календарного планування, проведення онлайн уроків тощо (рис 2).

Використання інструменту Meet середовища Google Workspace дало змогу проводити «захищені» онлайн уроки, бо, на відміну від безкоштовної версії сервісу Google Meet, заходити на заняття можуть лише учні, що мають корпоративний акаунт. Значно розширилися можливості сервісу Клас, де з'явилася змога запрошувати батьків учня для відстежування діяльності школярів на платформі. Слід відзначити й необмежений Google-Диск, який надається користувачам у межах цього тарифного плану, бо під час дистанційного навчання більшість учнів використовує мобільні пристрої та надсилає виконані роботи у графічному вигляді, що швидко призводить до заповнення стандартного Диска обсягом 15 ГБ. Але, окрім

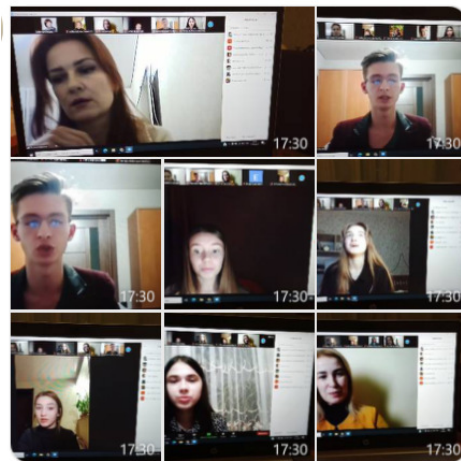


Рис. 2: Дистанційне навчання у Слов'янському педагогічному ліцеї.

надання суто технічних переваг, Google Workspace має ще й виховний аспект, оскільки отримання учнями корпоративного ліцейського акаунту ідентифікує їх як учнів певного закладу освіти та викликає почуття гордості за свій заклад.

## Висновки

Отже, Донецьким обласним інститутом післядипломної педагогічної освіти було розроблено та успішно впроваджено курси підвищення кваліфікації «Організація дистанційного навчання в закладі освіти засобами інструментів Google». Ці курси виявились дуже вчасними та корисними для організації дистанційного та змішаного навчання. Дистанційне навчання також виявилось корисним і для багатьох учнів, оскільки така форма навчання дозволила суттєво оптимізувати їх продуктивний час.

## Список використаних джерел

- [1] Лотоцька А.В., Пасічник О.В. Організація дистанційного навчання в школі : методичні рекомендації. 2020. URL: <https://nus.org.ua/questions/organizatsiya-dystantsijky-v-shkolah-rozrasnyuyemo-osnovne-z-rekomendatsij-mon/>
- [2] Гладун М.А., Сабліна М.А. Сучасні онлайн інструменти інтерактивного навчання як технологія співробітництва. *Open educational environment of modern University*. 2018. № 4. С. 33–43.
- [3] Сервіси та інструменти Google. URL: <https://sites.google.com/site/istoriakompanii/-google>

**Грудкіна Н.С., Васильєва Л.В., Єрмакова К.Ю.**

<sup>1</sup> д. т. н., доцент кафедри математики та моделювання Донбаської державної машинобудівної академії, Краматорськ

<sup>2</sup> к. т. н., доцент кафедри комп'ютерних інформаційних технологій Донбаської державної машинобудівної академії, Краматорськ

<sup>3</sup> магістрант факультету автоматизації машинобудування та інформаційних технологій Донбаської державної машинобудівної академії, Краматорськ

*e-mail:* <sup>1</sup> vm.grudkina@ukr.net, <sup>2</sup> vasileva.dgma@gmail.com, <sup>3</sup> kit@dgma.donetsk.ua

## **РОЗРОБКА WEB-СЕРВІСУ ОЦІНКИ РІВНЯ КОНКУРЕНТОСПРОМОЖНОСТІ СТРАХОВИХ ПОСЛУГ**

Розроблено Web-сервіс оцінки рівня конкурентоспроможності страхових послуг, що дозволяє визначити поточний рівень системи реалізації страхових послуг компанією. Це забезпечить своєчасність виявлення основних її проблем та можливість корегування побудови ефективної моделі системи реалізації страхових послуг, що сприятиме підвищенню ефективності страхової діяльності в цілому.

**Ключові слова:** *Web-сервіс, страхові послуги, конкурентоспроможність.*

### **Вступ**

Український страховий ринок в сучасних реаліях значно поступається ринкам розвинених країн, де збори середньої страхової компанії можуть перевищувати збори страхових премій українського страхового ринку в цілому. Аналізу проблем оцінки конкурентоспроможності страхових компаній присвячені роботи багатьох вчених, в тому числі з виокремленням проблем, характерних для перехідної економіки. Однак на даний час актуальним є не стільки визначення місця та ролі страхової галузі у підвищенні рівня економіки нашої країни, скільки забезпечення реалізації використання комплексного аналізу факторів підвищення конкурентоспроможності страхових послуг [1 – 3], що полягає у побудові ефективної моделі системи їх реалізації у вигляді програмних модулів саме із оцінки рівня конкурентоспроможності страхової послуги [3 – 5].

З огляду на поставлену мету вивчено особливості, сутність та механізм реалізації страхових послуг та конкурентної політики на національному ринку страхових послуг. Проведений аналіз найуживаніших математичних методів обробки експертної інформації дозволив визначитися із математичним інструментарієм оцінки рівня конкурентоспроможності страхової послуги.

## Функціонал розробленого Web-сервісу

Побудовано концептуальну модель програмного модуля для визначення рівня конкурентоспроможності страхової послуги на основі уніфікованої мови моделювання UML [5]. Розроблено Web-сервіс із визначення рівня конкурентоспроможності страхової послуги, що дозволяє визначити поточний рівень системи реалізації страхових послуг компанією [6]. Відповідно до виявлених основних чинників впливу на конкурентоспроможність страхової послуги, оцінку рівня конкурентоспроможності досліджуваної страхової послуги проводили за показниками: швидкість та простота отримання послуги, міра довіри до страхової компанії, привабливість страхових тарифів, оптимальність страхової послуги (співвідношення вартість – якість), обсяг страхового покриття, розмір франшизи, кількість основних та додаткових послуг, ступінь урахування спектра можливих ризиків, простота норм страхового договору та орієнтація на захист інтересів клієнта та швидкість та справедливність врегулювання страхових випадків. Структура додатка визначення рівня конкурентоспроможності страхової послуги наступна:

1. Розділ «Мій кабінет».
2. Розділ «Новий розрахунок»:
  - підрозділ «Графічне порівняння показників»;
  - підрозділ «Оцінка узгодженості експертів»;
  - підрозділ «Оцінка послуги»;
  - підрозділ «Оцінка послуги конкурента».
3. Розділ «Налаштування облікового запису».

Розділ «Мій кабінет» є головним екраном програмного модуля та демонструє всі виконані розрахунки даним користувачем. Розрахунки завантажуються з бази даних на ім'я користувача із можливістю переглянути звіт щодо виконаного розрахунку (рис. 1). Побудована діаграма класів дає уявлення статичної структури моделі системи та призначена для відображення взаємозв'язків між окремими сутностями предметної області з описами їхньої внутрішньої структури і типів відносин.

Розділ «Новий розрахунок» включає підрозділи «Оцінка узгодженості експертів», «Оцінка послуги», «Оцінка послуги конкурента» та «Графічне порівняння показників» та стає доступним лише після авторизації.

Підрозділ «Графічне порівняння показників» (рис. 2) призначений для візуалізації значень показників, що використовуються для оцінки рівня конкурентоспроможності страхової послуги як досліджуваної компанії, так і компанії-конкурента. Це дозволить мати наочну загальну картину

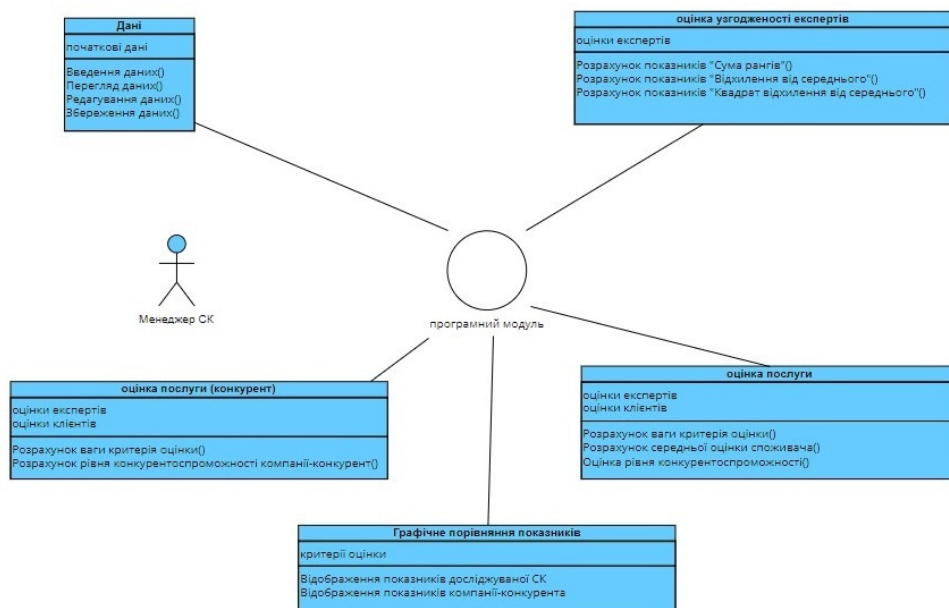


Рис. 1: Діаграма класів.

отриманого результату за кожним аналізованим показником та відслідкувати динаміку зміни його значень з метою порівняння та полегшення аналізу, що проводить менеджер страхової компанії. Діаграми будуються автоматично після здійснення розрахунків щодо оцінки рівня конкурентоспроможності страхової послуги за допомогою графічного порівняння показників у вкладці «Графічне порівняння показників».



Рис. 2: Розділ «Графічне порівняння показників».

У підрозділі «Оцінка узгодженості експертів» здійснюється оцінка узгодженості думок експертів, отримана ступінь узгодженості вказує на

якість результуючої оцінки рівня конкурентоспроможності досліджуваної страхової послуги.

## Висновки

Таким чином, розроблений Web-сервіс оцінки рівня конкурентоспроможності страхових послуг дозволяє визначити поточний рівень системи реалізації страхових послуг компанією, що забезпечує своєчасність виявлення проблем та можливість корегування побудови ефективної моделі системи реалізації страхових послуг, що сприятиме підвищенню ефективності страхової діяльності в цілому.

## Список використаних джерел

- [1] Козьменко О.В. Страхування і страховий ринок: термінологія, законодавство і динаміка розвитку. Суми: Ділові перспективи, 2006.
- [2] Городніченко Ю.В. Тенденції та перспективи розвитку страхового ринку України. *Економіка та суспільство*. 2017. № 10. С. 569–573.
- [3] Золотарьова О.В. Ключові тенденції та пріоритети розвитку ринку страхових послуг в Україні. *Економіка та суспільство*. 2017. № 11. С. 413–420.
- [4] Статистика страхового ринка України. URL: <https://forinsurer.com/stat> (дата звернення: 15.12.2021).
- [5] Боггс М. UML и Rational Rose. Москва: Лори, 2010.
- [6] Грудкіна Н.С., Сташкевич І.І. Розробка WEB сервісу для визначення рівня конкурентоспроможності страхових послуг. *Інформатика, управління та штучний інтелект* : тези восьмої міжнародної науково-технічної конференції (м. Харків, 16 – 19 лист. 2021 р.). Харків, 2021. С. 29.

<sup>1</sup> к. ф-м. н., доцент кафедри інформатики Національного університету «Києво-Могилянська Академія», Київ

<sup>2</sup> д. ф-м. н., професор кафедри інформатики Національного університету «Києво-Могилянська Академія», Київ

e-mail: <sup>1</sup> ngulayeva@yahoo.com, <sup>2</sup> glib@ukma.edu.ua

## ГЕНЕТИЧНІ АЛГОРИТМИ І ЧИСЕЛЬНІ МЕТОДИ ОПТИМІЗАЦІЇ

Описано клас задач, які можуть бути розв'язані за допомогою генетичних алгоритмів, та показано спосіб зведення заданої оптимізаційної задачі до задачі, розв'язуваної за допомогою генетичного алгоритму. Показано, що генетичні алгоритми розв'язання екстремальних задач є стохастичними чисельними методами прямого пошуку.

**Ключові слова:** задачі безумовної та умовної оптимізації, чисельні методи, генетичні алгоритми.

Генетичні алгоритми, в основу яких покладено ідею еволюції живої природи, широко використовуються для розв'язування різноманітних задач оптимізації; при цьому єдине обмеження, яке накладається на цільову функцію – обчислюваність в кожній точці пошукового простору [1]. Серед основних властивостей генетичних алгоритмів відзначимо такі: понятійний апарат запозичений з біології; кожен допустимий розв'язок задачі кодується у вигляді хромосоми та представлений особиною популяції; пошук розв'язку здійснюється одночасно цілою популяцією особин, причому для генерації нового розв'язку використовуються аналоги генетичних операторів. В загальному випадку генетичні алгоритми не можуть бути застосовними до задач, для яких множина  $\arg \min_{x \in X} f(x) = \emptyset$ .

Генетичні алгоритми природно розглядати як чисельні методи прямого пошуку. Наприклад, представимо набором  $s$  генетичний алгоритм турнірного витиснення з гаусовою мутацією, запропонований в [2] для розв'язання задачі мінімізації функції  $f(x)$  на множині  $X$ :  $f(x) \rightarrow \min, x \in X$  або  $\min_{x \in X} f(x)$  в постановці знаходження всіх точок локальних мінімумів та значення функції в цих точках:  $X = X_1 \times X_2 \times \dots \times X_n$ ,  $X_i = [a_i, b_i] \subseteq R \quad \forall i = \overline{1, n}$  (як варіант, може розглядатися задача максимізації). Наведемо схему цього алгоритму для розв'язування задач в постановці знаходження хоча б однієї точки  $x^* = \arg \min_{x \in X} f(x)$ :

1. Кодування в дійсних числах: кожен ген хромосоми

$\vec{X} = (x_1, x_2, \dots, x_n) \in R^n$  є дійсною змінною  $x_i = [a_i, b_i] \subseteq R$ .

2. Генерація початкової популяції за рівномірним розподілом.
3. Обчислення коефіцієнта пристосованості всіх особин популяції; при цьому функція здоров'я збігається з цільовою функцією.
4. Породження  $L$  нащадків кожною особою шляхом застосування оператора гаусової мутації до кожного гена особи:  $x_i = x_i + N_i(0, \sigma)$ , де  $N_i(0, \sigma)$  – стандартна нормально розподілена випадкова величина.
5. Обчислення коефіцієнта пристосованості отриманих нащадків.
6. Формування нової популяції за допомогою турнірного відбору; група для кожного турніру формується з батька та усіх його нащадків.
7. Якщо кількість ітерацій дорівнює  $G$ , завершення роботи алгоритму (розв'язком задачі є найкраща особина популяції); інакше перехід на крок 4.

Описаний алгоритм можна подати як паралельне виконання  $N$  екземплярів алгоритмів, кожен з яких описується за такою схемою  $c$  (розглядається задача мінімізації).

$\Phi$  – клас функцій, визначених на  $R^n$  та обчислюваних в кожній точці;

$G_1(\Phi) = \xi$ , де  $\xi$  – вектор випадкових дійсних чисел;

$G_{k+1}(\Phi, \omega_k) =$

$\arg \min \left\{ f(x^k), \min_{i=1, \dots, L} \{ f(x_1^k + N_{1,i}^k(0, \sigma), \dots, x_n^k + N_{n,i}^k(0, \sigma)) \} \right\}, k \geq 1;$

$H_k(\Phi, \omega_k) = \begin{cases} 0, & k = G \\ 1, & k < G \end{cases};$

$E_k(\Phi, \omega_k) = f_k^* = \min_{1 \leq l \leq N} f(x^{k,l})$ , де  $x^{k,l}$  – точка випробування  $l$ -го генетичного алгоритму на  $k$ -ій ітерації.

Описаний генетичний алгоритм легко трансліюється в схему  $c$  завдяки таким своїм особливостям: відсутність оператора кросинговеру; локальний відбір особин в наступну популяцію (використовуються турнірні змагання між батьками та їх власними нащадками). Для того щоб подати довільний генетичний алгоритм у вигляді схеми  $c$ , необхідно провести модифікацію цієї схеми.

Розглянемо спочатку генетичні алгоритми розв'язання задач безумовної оптимізації:  $f(x) \rightarrow \min, x \in X = R^n$  (1).

Для розв'язування задачі (1) за допомогою генетичного алгоритму необхідно визначити:

1) відображення (бієктивне)  $\mu : X \rightarrow S$ , яке зіставляє кожному допустимому розв'язку задачі  $x \in X = R^n$  код  $s \in S$ ; очевидно, зворотне

відображення  $\mu^{-1} : S \rightarrow X$  за кодом  $s \in S$  відновлюватиме розв'язок  $x \in X$ . Множина  $S$  може бути, наприклад, множиною бінарних рядків заданої довжини або навіть збігатись з множиною  $X$ ;

2) функцію здоров'я (приспосованості)  $F(s)$ ,  $s \in S$ , яка має задовольняти таким вимогам:

- адекватність задачі:  $\forall s_1, s_2, x_1 = \mu^{-1}(s_1), x_2 = \mu^{-1}(s_2)$  виконується  $F(s_1) < F(s_2) \Leftrightarrow f(x_1) < f(x_2)$ , де  $s_1, s_2$  – закодовані значення розв'язків  $x_1, x_2$ ;  $F$  – функція приспосованості,  $f$  – цільова функція оптимізаційної задачі;
- невисока складність обчислення;
- різноманітний рельєф.

Часто покладають  $F(s) \equiv f(x)$ , де  $s$  – хромосома, що кодує значення розв'язку  $x(s = \mu(x))$ . Іноді, залежно від параметрів генетичного алгоритму, роблять незначні перетворення цільової функції. Наприклад, для можливості використання ймовірнісного відбору роблять таке перетворення цільової функції, яке дозволяє звести задачу мінімізації цільової функції до задачі максимізації функції приспосованості.

Розв'язування задачі (1) зводиться до розв'язування за допомогою генетичного алгоритму задачі  $F(s) \rightarrow \min, s \in S$  (2).

Так, нехай  $s^*$  – розв'язок задачі (2) у постановці, коли потрібно знайти хоча б одну точку  $x^* = \arg \min_{x \in X} f(x)$ . Тоді розв'язком задачі (1) буде  $x^* = \mu^{-1}(s^*)$ .

Важливою характеристикою генетичного алгоритму є його паралелізм: пошук розв'язків здійснюється одночасно популяцією особин. Іншими словами, на кожній ітерації алгоритму аналізується множина точок випробування (кількість таких точок дорівнює розміру популяції та є сталою). В зв'язку з цим для опису генетичного алгоритму як прямого чисельного методу, модифікуємо схему  $c$  так, щоб на кожному кроці розглядалась не одна, а  $N$  точок випробування.

1. Обрати множину точок першого випробування (в термінах генетичного алгоритму – згенерувати початкову популяцію):  $\langle s_1^1, s_2^1, \dots, s_N^1 \rangle$ ,  $s_i^1 = G_1(\Phi) = \xi_i$ , де  $\xi_i$  – випадкова величина (залежно від множини  $S$  це може бути випадкове дійсне число, послідовність довжини  $n$  випадкових дійсних чисел, випадкова двійкова послідовність довжини  $n$  та ін.).
2. Нехай обрано множину точок  $k$ -го випробування  $\langle s_1^k, s_2^k, \dots, s_N^k \rangle$ ,  $k \geq 1$ . Після обчислення значення функції приспосованості в кожній з цих точок  $y_i^k = F(s_i^k)$  отримаємо пошукову (апостеріорну) інформацію про функцію  $F$ :  $\omega_k = \{(s_1^k, y_1^k), (s_2^k, y_2^k), \dots, (s_N^k, y_N^k)\}$ .

Звернемо увагу, що апостеріорна інформація про функцію  $F$  містить тільки інформацію, отриману на поточному кроці роботи алгоритму (на відміну від оригінальної схеми  $s$ , в якій запам'ятовується вся інформація, отримана від початку роботи алгоритму).

3. Визначити поточну оцінку екстремуму:  $e_k = E_k(\Phi, \omega_k) = \min_{1 \leq i \leq N} y_i^k$ .
4. Для обчислення множини точок наступних випробувань до поточної множини точок  $\langle s_1^k, s_2^k, \dots, s_N^k \rangle$  застосовують послідовність операторів: відбір в батьківський пул  $\langle p_1^k, p_2^k, \dots, p_N^k \rangle = G_{k+1}^{SEL}(\langle s_1^k, s_2^k, \dots, s_N^k \rangle, \omega_k)$ , кросинговер  $\langle c_1^k, c_2^k, \dots, c_N^k \rangle = G_{k+1}^{CROS}(\langle p_1^k, p_2^k, \dots, p_N^k \rangle)$ , мутація  $\langle s_1^{k+1}, s_2^{k+1}, \dots, s_N^{k+1} \rangle = G_{k+1}^{MUT}(\langle c_1^k, c_2^k, \dots, c_N^k \rangle)$ . Іншими словами,  $G_{k+1}(\Phi, \omega_k)$  являє собою послідовність спеціальних генетичних операторів. Відбір відкидає деякі «невдалі» розв'язки, та натомість, дублює «вдалі». Кросинговер здійснює обмін частинами розв'язків, а мутація фактично є локальним перетворенням розв'язку (збуренням).
5. Визначити величину  $h_k = H_k(\Phi, \omega_k) \in \{0, 1\}$ . Якщо  $h_k = 1$ , то збільшити номер кроку  $K$  на одиницю та перейти на п. 2. Інакше ( $h_k = 0$ ) зупинити обчислення. Розв'язок задачі має оцінку  $e_k$ . В більшості генетичних алгоритмів  $H_k(\Phi, \omega_k) = 0$ , якщо  $k = G$ , де  $G$  – максимальна кількість ітерацій (поколінь), або ідентифіковано збіжність генетичного алгоритму (наприклад,  $\left| e_k - \frac{\sum_{i=1}^N y_i^k}{N} \right| < \varepsilon$ ).

Зауважимо, що кількість різних класів генетичних алгоритмів є практично необмеженою: залежно від способу уточнення генетичних операторів можна будувати нові класи генетичних алгоритмів. Наведена схема є загальним описом генетичного алгоритму з генераційним типом репродукції; опис алгоритму зі стійким типом репродукції вимагає незначної модифікації.

Підсумовуючи сказане та враховуючи стохастичність операторів  $G_{k+1}^{SEL}$ ,  $G_{k+1}^{CROS}$ ,  $G_{k+1}^{MUT}$ , можна говорити про генетичний алгоритм як про стохастичний чисельний метод прямого пошуку.

## Генетичні алгоритми в задачах умовної оптимізації

Досі ми обговорювали генетичні алгоритми розв'язання задач безумовної оптимізації (1). В задачах умовної оптимізації допустима множина

$X \subset R^n$ . Загальні методи врахування обмежень в задачах математичного програмування детально описані в [3]: методи штрафних функцій, множників Лагранжа, параметризації цільової функції та ін.

Для врахування обмежень на допустимі розв'язки в генетичних алгоритмах застосовують такі підходи.

А) Вибір такого способу кодування розв'язків, що всі значення з множини  $S$  декодуються в допустимі значення:  $x = \mu^{-1}(s) \in X \subset R^n$ . Зазначимо, що на практиці розробити такий спосіб кодування не завжди вдається.

Б) Розробка спеціальних генетичних операторів (кросинговер, мутація), застосування яких гарантує породження допустимих розв'язків. Тобто, після застосування генетичного оператора породжений ним розв'язок буде допустимим (нова точка випробування задовольняє обмеженням на розв'язок).

В) Застосування модифікації широко використовуваного в інших чисельних методах оптимізації методу штрафних функцій. В цьому випадку цільова функція з задачі (2) (функція здоров'я) замінюється функцією  $F_p(s) = F(s) + \gamma P(s)$ ,  $s \in S$ .

Тут  $F_p(s)$  – нова цільова функція (функція здоров'я в генетичному алгоритмі);  $P(s)$  – функція штрафу,  $P(s) = \begin{cases} 0, & x = \mu^{-1}(s) \in X \\ > 0, & x = \mu^{-1}(s) \notin X \end{cases}$ ;  $\gamma$  – коефіцієнт штрафу. Така заміна цільової функції дозволяє погіршити оцінку якості розв'язку (оцінку точки випробування), в якому порушуються задані умовою обмеження.

Відмінність методу штрафних функцій в генетичних алгоритмах полягає в тому, що значення коефіцієнтів штрафу  $\gamma$  підбираються експертним або експериментальним шляхом і способи підбору цих коефіцієнтів чітко не регламентуються.

## Висновки

Питанню висвітлення алгоритмів оптимізації присвячено багато робіт [4,5,6]. Генетичні алгоритми відносяться до евристичних (взагалі кажучи, до метаевристичних) методів. Хоча досить давно ведуться роботи, присвячені теоретичному аналізу генетичних алгоритмів, розроблені теорії – теорема схем, гіпотеза будівельних блоків, залучення теорії марківських ланцюгів або проведення аналогій з методом відпалу – не є загально-визнаними теоріями обґрунтування збіжності генетичних алгоритмів [1]. Поширеною практикою є експериментальний аналіз ефективності гене-

тичних алгоритмів: численні прогони алгоритму на однакових вхідних даних з обчисленням усереднених значень критеріїв оцінювання якості отриманих розв'язків.

Досить часто як окремий клас розглядають алгоритми, породжені ідеєю імітації природних явищ (роєві, еволюційні алгоритми). Хоча ідея виділення в окремий клас алгоритмів, на створення яких авторів надихнуло спостереження за навколишнім середовищем, може здатись цілком природною, така класифікація не розкриває принципів роботи самих алгоритмів, їхньої суті.

В роботі [2] пропонується проводити класифікацію методів комбінаторної оптимізації за вісьмома характеристиками: принцип прийняття розв'язку, складність структури, тип траєкторії, що формується алгоритмом, та ін.. Ця класифікація є ґрунтовною, але дає надвелику кількість різних класів. Так, за такої класифікації генетичні алгоритми є стохастичними метаевристичними траєкторно-розривними популяційними алгоритмами без зміни ландшафту пошуку, без пам'яті, в загальному випадку без адаптації, задачно-орієнтованими.

Загалом, на думку авторів, наразі стан досліджень методів розв'язання оптимізаційних задач характеризується надзвичайно великою кількістю розроблених методів оптимізації та постійно зростає, причому достатньо поширеною є практика надання нових назв раніше відомим поняттям або незначним модифікаціям раніше розроблених методів. Такий стан справ пояснюється надзвичайною розгалуженістю різних напрямів досліджень і вузькою спеціалізацією дослідників. Необхідний перегляд та переосмислення історії виникнення, розробки й модифікацій різних методів оптимізації для узагальнення і систематизації цих методів. Автори сподіваються, що наведене представлення генетичного алгоритму як чисельного методу прямого пошуку може стати одним з перших кроків на цьому шляху.

## Список використаних джерел

- [1] Глибовець М.М., Гулаєва Н.М. Еволюційні алгоритми. Підручник. Київ: НаУКМА, 2013.
- [2] Шило В.П., Глибовець М. М., Гулаєва Н. М., Нікіщихіна К. В. Генетичні алгоритми турнірного витиснення з гаусовою мутацією. *Кібернетика та системний аналіз*. 2020. Т. 56, №2. С. 75–88.
- [3] Городецкий С.Ю., Гришагин В.А. Нелинейное программирование и многоэкстремальная оптимизация: учебное пособие. Нижний Новгород: Изд-во Нижегородского госун-та, 2007.

- [4] Сергиенко И.В., Шило В.П. Проблемы дискретной оптимизации: сложные задачи, основные подходы к их решению. *Кибернетика и системный анализ*. 2006. Т. 42, №4. С. 3–25.
- [5] Сергиенко И.В., Гуляницкий Л.Ф., Сиренко С.И. Классификация прикладных методов комбинаторной оптимизации. *Кибернетика и системный анализ*. 2009. Т. 45, №5. С. 71–83.
- [6] Scheibenpflug A., Wagner S., Pitzer E., Burlacu B., Affenzeller M. On the analysis, classification and prediction of metaheuristic algorithm behavior for combinatorial optimization problems. *Proceedings of the 24th European Modeling and Simulation Symposium (EMSS 2012)*. 2012. P. 368–372.

## ОСОБЛИВОСТІ ОРГАНІЗАЦІЇ ПЛАТФОРМ КЛІЄНТСЬКИХ ДАНИХ

Наведено структуру бази клієнтських даних на спеціалізованій платформі у контексті підтримки цифрової маркетингової стратегії. Визначено переваги і проблеми використання такої платформи, а також специфічні функції системи керування даними.

**Ключові слова:** платформа даних клієнтів, база даних, маркетинг.

### Вступ

Системи управління взаємовідносинами з клієнтами (CRM) стали невідмінним засобом організації бізнес-процесів навколо інтересів споживачів [1]. CRM охоплює всі аспекти взаємодії відділів продажів, маркетингу, технічної підтримки та сервісного обслуговування з потенційними та існуючими клієнтами, але цього стає недостатньо за умов швидкого розвитку цифрової економіки. Нові бізнес-моделі, а також зміна поведінки споживачів і процесів прийняття ними рішень щодо покупки, зумовили появу платформ управління даними, використання яких обмежено анонімністю зібраних даних, оскільки сучасна стратегія маркетингу передбачає цифрове розпізнавання та профілювання споживачів [2]. Акумуляція даних з різних джерел та створення уніфікованої бази даних, на основі якої можна сформувати єдиний комплексний профіль для кожного споживача, потрібні для аналізу та підтримки маркетингових рішень, які можливі на основі спеціалізованих платформ клієнтських даних.

### Структура бази клієнтських даних

За результатами дослідження інформаційних потреб та практики цифрового маркетингу [2 – 4] було визначено основні категорії клієнтських даних, що їх слід нагромаджувати на платформі.

#### 1. Профілі клієнтів:

- контактна інформація і демографічні характеристики – стать, вік, сімейний статус, освіта, професія, національність, етнічна належність, релігійні переконання тощо;
- психологічний портрет – темперамент, характер, емоційність, комунікабельність, волюві якості, інтелектуальність, самооцінка, самовладання та інші характеристики за ознаками-індикаторами;
- характеристики особистості – цілі, цінності, мотивації, проблеми, емоції та звички;
- соціально-економічні характеристики – рівень доходів, місце роботи, вид діяльності та посада, умови проживання;
- стиль життя – дані, необхідні для визначення моделі споживання, оцінки купівельної спроможності, виявлені переваги (улюблені локації, харчові уподобання, вибір розваг, відпочинку, техніки тощо), поточні і потенційні інтереси;
- соціальна поведінка – самоідентифікація, належність до соціальних груп;
- оточення клієнта – характеристики географічних, кліматичних, економічних, політичних, соціальних, культурних та інших умов й особливостей, в яких проживає та приймає рішення клієнт;
- асоційовані особи – зазначення типів зв'язків (родинних, дружніх, ділових тощо) між теперішніми клієнтами; нагромадження даних про нових осіб як потенційних клієнтів у майбутньому;
- офлайн-поведінка у ролі споживача – маршрути переміщень, включно з туристичними поїздками, точки одержання комерційних послуг (магазини, заклади харчування, готелі тощо), час або періодичність витрат, їх обсяги та структура, середній чек.

Щодо бізнес-клієнтів, крім контактної, має нагромаджуватись:

- бізнес-інформація – розмір, вид діяльності, кількість співробітників, регіон діяльності, річний оборот, бізнес-потреби тощо;
- інформація щодо процесу прийняття рішень – особи або групи осіб, які приймають участь у процесі підготовки і прийняття рішення, ступінь залучення і ролі, формальні і неформальні взаємозв'язки та відносини між особами, ступінь і характер їх впливу на остаточне рішення та на інших відповідальних осіб; процедури і правила вироблення, узгодження та прийняття рішень; оцінки тривалості відповідних процесів.

2. Дані клієнтів з CRM-систем – роль і статус клієнта, етап воронки продаж, на якій перебуває клієнт, канали його залучення, а також історія взаємодії з ним: угоди і звернення; історія виконання задач щодо клієнта; склад, обсяги і причини покупок; попередні маркетингові пропозиції клієнтові та його реакції на них. Також мають нагромаджуватись дані стосовно товарів і послуг, що були предметом інтересу або покупки клієнта.
3. Онлайн-поведінка клієнта – зміст його запитів; коло інтересів за відвідуванням сайтів, часом перебування на сторінці і кліками; використання додатків і вибір функцій; пристрої та операційна система, з якої відбувається доступ; середня кількість переглянутих за один сеанс сторінок, коефіцієнти конверсії, купівлі, завершення; кількість і зміст транзакцій, витрати в онлайн-режимі. Важливим аспектом для подальшого аналізу є дані щодо сприйняття та реакцій на цифрову рекламу.
4. «Голос клієнта» – дані, що відображають користувацький досвід, ставлення клієнтів та їхнє загальне враження від компанії, її продуктів, процесів передпродажного і післяпродажного обслуговування, рекламної кампанії та інших точок та аспектів взаємодії. Вони формуються за результатами опитувань споживачів щодо їхніх вимог, потреб, переваг, пріоритетів, проблем, бажаних змін у продукті або сервісі; оцінок задоволеності клієнтів; висновків та інсайтів за підсумками дискусій фокусних груп тощо.
5. Інформація сервісних центрів і служб підтримки клієнтів – технічні звіти; протоколи коллтрекінгу; скарги клієнтів та їх повідомлення про проблеми або труднощі; питання, що виникають у споживачів, зокрема потенційних; причини позитивного або негативного ставлення до продукту або компанії; висновки та інсайти за результатами польових досліджень; відгуки користувачів на реакції та дії компанії у відповідь на критику, схвальні відзиви або повідомлення про проблему.
6. Користувацький контент – контент, що його створюють лідери думок, агенти впливу, амбасадори і прокурори бренду щодо товару або компанії, а також відгуки і дописи пересічних користувачів.
7. Цифровий поголос – повідомлення, коментарі та інші неформальні комунікації між двома або більше особами у соціальних медіа, соціальних мережах, зокрема, ті, що набувають поширення. Такі дані збираються у розрізі приводів виникнення реакцій та осіб, які спричинили розголос; задіяних способів привертання уваги; кана-

лів поширення повідомлень і думок; груп, співтовариств і спільнот, де відбувається обговорення.

8. Аналітична інформація щодо типових клієнтів, сегментів клієнтів, цільових аудиторій, як результат оброблення зібраних і нагромаджених даних.

## **Переваги і проблеми організації платформ клієнтських даних**

Нагромадження структурованих і неструктурованих даних з різних джерел, їх великий обсяг, а також оброблення під час вирішення численних маркетингових задач підвищують вимоги до системи керування даними. Водночас інтеграція клієнтських даних на єдиній платформі забезпечує переваги:

- ненадмірність даних – платформа забезпечує доступ до даних всім прикладним системам, тому виключається дублювання даних;
- високорівневий доступ – функції керування даними передаються платформі;
- незалежність даних – структуру даних платформи не зумовлено функціоналом існуючих прикладних систем, реалізується концепція маркетингу, керованого даними;
- реалізація непередбачених запитів – інтерфейс платформи має підтримувати конструювання нових запитів, не передбачених у прикладних системах;
- ефективне виконання службових функцій – контроль транзакцій, протоколювання та відновлення даних для підтримки їх цілісності, організація і розмежування доступу до даних численних засобів і користувачів, тощо;
- розширені можливості з аналізу даних – нагромадження великих даних надає можливість маркетологам одержати більш повний і точний портрет цільової аудиторії, сфокусуватись на її потребах та особливостях, розробити ефективнішу рекламну кампанію і поліпшити маркетингову стратегію в цілому.

Клієнтські дані є цінним активом, наявність якого забезпечує конкурентну перевагу компанії. Створення повноцінних платформ клієнтських даних можливе лише за рахунок сумісного використання даних, що відповідає тенденції платформізації економіки [5]. Це актуалізує проблему захисту даних і, водночас, полегшує проведення незалежного довіреного

контролю у рамках охорони персональних даних спеціалізованими інституціями. Нагромадження клієнтських даних має розглядатись як їх передача до трастового фонду на умовах, погоджених персонально з кожним клієнтом [6]. Тому система керування даними платформи має підтримувати визначення правил оброблення персональних даних згідно укладених з клієнтами угод з автоматичним накладанням відповідних обмежень, контроль уповноважених осіб за доступом і використанням даних, а також видалення або анонімізацію даних на вимогу власників.

## Висновки

Платформи клієнтських даних є перспективним засобом підвищення ефективності маркетингової діяльності за умови нагромадження різномірної інформації з численних джерел і підтримки формування цілісного профілю споживача. Розвиток таких платформ збільшує ризики неправомірного використання персональних даних, що зумовлює потребу у додаткових інструментах їх контролю.

## Список використаних джерел

- [1] Buttle F., Maklan S. Customer Relationship Management. London: Routledge, 2019.
- [2] Kotler M., Wang S., Qiao C. Marketing strategy in the digital age: applying Kotler's strategies to digital marketing. Singapore: World Scientific Publishing, 2020.
- [3] Ryan D. Understanding digital marketing (3d ed.). London: Kogan Page, 2014.
- [4] Augmented Customer Strategy: CRM in the Digital Age / Ed. G. N'Goala, V. Pez-Perard, I. Prim-Allaz. London: ISTE, 2019.
- [5] Digital Economy Report. Value Creation and Capture: Implications for Developing Countries (UNCTAD, 2019). Url: [https://unctad.org/system/files/official-document/der2019\\_en.pdf](https://unctad.org/system/files/official-document/der2019_en.pdf)
- [6] Waldman A.E. Privacy as Trust. Information privacy for an information age. Cambridge: Cambridge University Press, 2018.

## ПОШУК ЗАКОНОМІРНОСТЕЙ ДЛЯ КЛАСІВ ТЕКСТОВИХ ДОКУМЕНТІВ

Запропоновано алгоритм пошуку знань із неструктурованих даних, а саме текстових документів. Цей алгоритм розширює можливості інформаційного пошуку.

**Ключові слова:** *закономірність, синтез кон'юнкцій, генетичний алгоритм.*

### Вступ

Останнім часом на перший план виходить задача попередньої «когнітивної обробки» неструктурованих даних, зокрема текстових документів. Сучасні пошукові системи переходять від пошуку документів з заданої тематики до пошуку інформації в цих документах, а далі й до витягу знань з цієї інформації. Таким чином, на перший план висувається задача витягу інформації з подальшим її структуруванням.

### Постановка задачі пошуку закономірностей

Нехай маємо множину закономірностей  $Z$ , множину текстових документів  $D$ , в яких відображена множина закономірностей, і їх відношення  $\psi[z(d_j)]$  – функція пошуку закономірностей. Потрібно знайти таку функцію, при певному значенні якої стане можливим виявити всі документи, які відображають цю закономірність. Функцію  $\psi[z]$  будемо називати пошуковою [1]. Закономірність – об'єктивно існуючий, постійний і необхідний взаємозв'язок між предметами, явищами чи процесами, що впливає з їх внутрішньої природи, сутності [2]. У цьому дослідженні розглянуто закономірність як наявність кортежів термінів для класу текстових документів. Відповідні закономірності  $z$  документи згідно з пошуковою функцією  $\psi[z]$  будемо називати позитивними (positive), у протилежному випадку – негативними (negative).

Нехай  $P_c$  – число документів класу позитивних з вибірки  $D$ ;  $p_c(\psi)$  – число документів, для яких виконується умова  $\psi(d) = 1$ ;  $N_c$  – число

негативних для вказаної закономірності документів у вибірці  $D$ ;  $n_c(\psi)$  – число документів, для яких виконується умова  $\psi(d) = 0$ . Інформативність пошукової функції  $\psi$  підвищується зі збільшенням кількості позитивних документів, що вона знаходить, та зі зменшенням частки в них негативних документів. Отже, задача побудови пошукової функції  $\psi$  зводиться до оптимізації за двома критеріями:  $p_c(\psi) \rightarrow \max$  і  $n_c(\psi) \rightarrow \min$  [3]. Введемо позначення  $E_c$  для частини негативних документів серед усіх ресурсів, що виділяються, і  $D_c$  для частини позитивних документів, серед знайдених:  $E_c(\psi, D) = \frac{n_c(\psi)}{p_c(\psi) + n_c(\psi)}$ ,  $D_c(\psi) = \frac{p_c(\psi)}{N}$ , де  $N$  – кількість документів у колекції. Статистичною інформативністю пошукової функції  $\psi(d)$  будемо називати величину  $I_c(\psi, D) = \ln(h_P^n)$ . Пошукова функція  $\psi(x)$  буде статистично закономірною для множини відібраних документів, якщо  $I_c(\psi, D) > I_0$  при заданому досить великому  $I_0$ .

## Процедури пошуку логічних закономірностей

Нехай  $W$  – скінченна множина термів (словник). Розглянемо множину кон'юнкцій із обмеженим числом термів з  $W$ :  $K_k|W| = \{\psi = w_1 \cap w_2 \cap \dots \cap w_k | w_1, w_2, \dots, w_k \in W, k \leq K\}$ . Поставимо кожній кон'юнкції  $\psi$  у відповідність її окіл – множину кон'юнкцій  $V(\psi)$ , яку можна отримати з  $\psi$  через елементарні модифікації: додаванням, вилученням або модифікацією одного з термів кон'юнкції. Починаючи із заданої кон'юнкції  $\psi_0$ , будується послідовність кон'юнкцій  $\psi_0, \psi_1, \dots, \psi_n$ , в якій кожна наступна кон'юнкція  $\psi_i$  вибирається з околу попередньої  $V_i = V(\psi_{i-1})$  за критерієм максимуму інформативності. Ітераційний процес сходиться за скінченну кількість кроків до деякої локальної кон'юнкції, яку вже не можна покращити. Як відомо, це є метод градієнтного синтезу кон'юнкцій.

## Генетичний алгоритм синтезу кон'юнкцій

Подальшим удосконаленням алгоритму градієнтного синтезу кон'юнкцій є генетичний алгоритм, у принципах дії якого лежать ідеї дарвінівської еволюції. Головна його відмінність в тому, що на кожному кроці відбирається не найкраща кон'юнкція, а множина кращих кон'юнкцій, яку називають популяцією. З них породжується велика кількість кон'юнкцій – нащадків за допомогою двох генетичних операцій – кросинговеру та мутації. Так можна отримати величезну кількість нащадків, але на практиці формують обмежену кількість, не більше  $T_1$  нащадків через випадкові

схрещування і мутації. Потім проводиться природний відбір, в результаті якого до наступного покоління переходять не більше  $T_0$  найбільш інформативних нащадків.

Якщо побудувати матрицю взаємозв'язків термінів у колекції текстових документів [4] для кожного окремого класу текстових документів, то отримаємо, що зі збільшенням коефіцієнтів цієї матриці збільшується значуща кон'юнкція термів для цього класу. Отже, кон'юнкції будуть будуватися згідно з коефіцієнтами матриці взаємозв'язків. Для першої кон'юнкції відбираються два терми з найбільшим коефіцієнтом взаємозв'язку. Далі за допомогою функцій схрещування та мутації будуються нові кон'юнкції. Кон'юнкції створюють із урахуванням найбільш тісних і характерних взаємозв'язків для вибраних термів. Далі збудовані кон'юнкції перевіряються, чи будуть вони закономірністю для цього класу, чи ні. Тому для кожної сформованої кон'юнкції обраховують статистичну інформативність, і вона порівнюється з пороговою інформативністю, яка є налагоджуваним параметром. Відібрані кон'юнкції становлять множину закономірностей класу текстових документів.

## Висновки

Розв'язання задач здобуття інформації з текстових документів дозволяє автоматично переглядати великий обсяг текстів, що містять порівняно невелику кількість шуканої інформації. Виявлена в тексті інформація перетворюється у структуровану: виявляються цільові факти, об'єкти, відношення у вигляді, придатному для подальшого автоматичного оброблення, що значно розширює можливості інформаційного пошуку.

## Список використаних джерел

- [1] Воронцов К.В. Лекции по логическим алгоритмам классификации. 2007. Url: <http://www.ccas.ru/voron/download/LogicAlgs.pdf>
- [2] Díaz-Manríquez A., Ríos-Alvarado A.B., Barron-Zambrano J.H., et al. Automatic Document Classifier System Based on Genetic Algorithm and Taxonomy. *IEEE Access*. 2018. Vol. 6. P. 21552–21559.
- [3] Zhai C., Massung S. Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining. ACM Books, 2016.
- [4] Духновская К.К. Матрица взаимосвязи терминов во множестве информационных ресурсов для метода опорных векторов. *Управляющие системы и машины*. 2014. № 2. С. 4–9.

<sup>1</sup> к. ф-м. н., доцент кафедри прикладної математики Національного авіаційного університету, Київ

<sup>2</sup> старший викладач кафедри математичної фізики та диференціальних рівнянь НТУУ «Київський політехнічний інститут імені Ігоря Сікорського»

e-mail: <sup>1</sup> m\_ilicheva@ukr.net, <sup>2</sup> avdeeva.tetyana@gmail.com

## ДО МОДЕЛЮВАННЯ ВІДСОТКОВИХ СТАВОК ДИФУЗІЙНИМИ РІВНЯННЯМИ

У роботі представлені дифузійні рівняння моделей відсоткових ставок, які враховують стани економічної ситуації.

**Ключові слова:** дифузійне рівняння, вінерівський процес, марківський процес, умовне математичне сподівання.

### Вступ

Особливості поведінки фінансового ринку зазвичай пов'язують з геометричним броунівським рухом – випадковим процесом

$$S(t) = S(0) \exp \{ \sigma W(t) + \mu t \}, \sigma, \mu, S_0 \in \mathbb{R}, t \geq 0$$

$W = \{W(t), t \geq 0\}$  – броунівський рух, тобто випадковий процес  $W = \{W(t), t \geq 0\}$ , що задовольняє умовам: а)  $W(0) = 0$ ; б) процес  $W$  має незалежні прирости; в) випадкові величини  $W(t) - W(s)$  мають гауссів (нормальний) розподіл з середнім  $\theta$  і дисперсією  $t - s$ .

Геометричний броунівський рух:

$$S(t) = S_0 \exp \left\{ \left( \mu - \frac{\sigma^2}{2} \right) t + \sigma W(t) \right\}, t \geq 0$$

моделює зміну ціни акцій. Якщо  $S(t)$  і  $B(t)$  – ціни акцій і облігацій відповідно, то

$$dS(t) = S(t) (\mu dt + \sigma dW(t)), \quad dB(t) = r B(t) dt.$$

Тоді ціна акцій, дисконтована одиницею грошового ринку – ціною облігацій, має стохастичний диференціал [1, стор. 220]:

$$d \left[ \frac{S(t)}{B(t)} \right] = \left[ \frac{S(t)}{B(t)} \right] [(\mu - r) dt + \sigma dW(t)].$$

Стохастичні диференціальні рівняння – рівняння вигляду

$$dX(t) = b(t, X(t)) dt + \sigma(t, X(t)) dW(t), \quad X(0) = \xi, \quad 0 \leq t \leq T,$$

де  $\xi - F_0$  – вимірний випадковий величина широко використовуються при моделюванні фінансових станів ринку. Зміни відсоткових ставок  $r = r(t)$ ,  $t \geq 0$  відображають стан «економічної ситуації» на ринку.

Моделювання відсоткових ставок дозволяє аналізувати стан фінансового ринку. У свою чергу, важелі ринку відображаються через показники відсоткових ставок.

Нехай «економічна ситуація» характеризується випадковим процесом  $\Theta(t)$  (або випадковою величиною  $\Theta$ ). Із значеннями  $\Theta(t)$  пов'язаний випадковий процес  $X_s$ , який спостерігається до деякого моменту  $t$ , тобто  $s \leq t$ . На основі значень  $X_s$ ,  $s \leq t$  формують оцінку стану «економічної ситуації» (як умовне математичне сподівання). Одержана таким чином оптимальна оцінка стану «економічної ситуації» слугує значенням відсоткової ставки.

## Основні ідеї

Розглянемо модель банківського рахунку  $B = (B_n)_{n \geq 0}$ , для якого:

$$r_n = \frac{\Delta B_n}{B_{n-1}}, \quad (1)$$

тоді  $r = (r_n)_{n \geq 1}$  – відсоткова ставка. У випадку неперервного часу  $t \geq 0$  використовується неперервний аналог (1)  $dB_t = r(t) B_t dt$ , звідки  $B_t = B_0 \exp\left(\int_0^t r(s) ds\right)$ .

Нехай відсоткові ставки  $r = (r(t))_{t \geq 0}$  дифузійні випадкові процеси, траєкторії їх неперервні, причому  $(\Omega, F, (F_n)_{n \geq 0}, P)$  – стохастичний базис (ймовірнісний простір), який описує статистику фінансового ринку;  $(F_n)_{n \geq 0}$  – «інформація» фінансового ринку. Значення банківського рахунку  $B_n$  вважають  $F_{n-1}$  – вимірними.

Відома велика кількість моделей відсоткових ставок  $r = (r(t))_{t \geq 0}$ , які описуються дифузійними рівняннями виду [2 – 4]:

$$dr(t) = a(t, r(t)) dt + b(t, r(t)) dW_t. \quad (2)$$

$W = (W_t)_{t \geq 0}$  – стандартний вінерівський процес (броунівський рух).

Найбільш відомі наступні моделі дифузійних випадкових процесів [3, стор. 32]: Модель Дотхана:

$$dr(t) = r(t)(a dt + b dW_t),$$

Модель Кокса, Інгерсола і Росса:

$$dr(t) = \beta [r(t)]^{\frac{3}{2}} dW_t, \quad dr(t) = (\alpha - \beta r(t)) dt + b\sqrt{r(t)} dW_t,$$

Модель Хо і Лі:

$$dr(t) = \alpha(t) dt + b dW_t,$$

Моделі Халла і Уайта:

$$dr(t) = (\alpha(t) - \beta(t)r(t)) dt + b(t) dW_t,$$

$$dr(t) = (\alpha(t) - \beta(t)r(t)) dt + b(t) \cdot \sqrt{r(t)} \cdot dW_t,$$

Модель Блека і Карасинського:

$$dr(t) = r(t) \cdot (\alpha(t) - \beta(t) \ln r(t)) dt + b(t)r(t) dW_t,$$

Можна вважати, що відсоткові ставки залежать від загального стану «економічної ситуації». Нехай «економічна ситуація» знаходиться в безпосередній залежності від двох станів, тобто моделюється однорідним стрибкоподібним марківським процесом  $\Theta = (\Theta(t))_{t \geq 0}$  із двома станами:  $i = 0, 1$   $P(\Theta(0) = 0) = P(\Theta(0) = 1) = \frac{1}{2}$ , щільності ймовірностей переходів  $\lambda_{ij}$  такі:  $\lambda_{ii} = -\lambda$ ,  $\lambda_{ij} = \lambda$  при  $i \neq j$ . Тобто,  $\lambda_{00} = -\lambda$ ,  $\lambda_{01} = \lambda$ ,  $\lambda_{10} = \lambda$ ,  $\lambda_{11} = -\lambda$ . Стан «економічної ситуації» оцінюється за допомогою випадкового процесу  $X = (X_t)_{t \geq 0}$  з диференціалом:

$$dX_t = \Theta(t)dt + d\widetilde{W}_t, X(0) = 0, \quad (3)$$

$\widetilde{W} = (\widetilde{W}_t)_{t \geq 0}$  – вінерівський процес. Оптимальною (у середньоквадратичному сенсі) оцінкою стану  $\Theta(t)$  за спостереженнями  $X_s, s \leq t$  можна вважати значення відсоткової ставки:

$$r(t) = E(\Theta(t) | F_t^X),$$

де  $F_t^X = \sigma(X_s, s \leq t)$  Із загальної теорії статистичного оцінювання [3, стор. 400] впливає:

$$dr(t) = \lambda(1 - 2r(t)) dt + r(t)(1 - r(t))(dX_t - r(t)) dt \quad (4)$$

Оскільки процес:  $W_t = X_t - \int_0^t r(s)ds$  є вінерівським [3, стор. 297], то (4) можна записати таким чином:

$$dr(t) = \lambda(1 - 2r(t)) dt + r(t)(1 - r(t)) dW_t.$$

Якщо узагальнити вид рівняння (2):

$$dr(t) = a(\Theta(t), r(t)) dt + b(r(t)) d\widetilde{W}_t$$

одержимо, що

$$a(\Theta, r) = \lambda(1 - 2r) + r(1 - r)(\theta - r), \quad b(r) = r(1 - r).$$

Розглянемо інший підхід до введення величини, яка характеризує сан «економічної ситуації». Нехай  $\Theta$  – випадкова величина, яка приймає два значення 1 і 0 із ймовірностями  $p$  і  $(1 - p)$ , відповідно. Тоді процес  $(X_t)_{t \geq 0}$  виду (3) буде таким:

$$dX_t = \Theta dt + d\widetilde{W}_t, X(0) = 0.$$

Значення відсоткової ставки  $r(t)$  можна знайти з рівняння:

$$dr(t) = r(t)(1 - r(t))(dX_t - r(t)dt), \quad r(0) = p \quad (5)$$

і одержати наступне рівняння для  $r(t)$ :

$$\begin{aligned} dr(t) &= r(t)(1 - r(t)) dW_t & \frac{dr(t)}{r(t)(1 - r(t))} &= dW_t \\ \ln r(t)(1 - r(t)) + \ln p(1 - p) &= W_t & r(t)(1 - r(t)) &= \frac{1}{p(1 - p)} W_t \end{aligned}$$

Моделюючи вінерівський рух при даних умовах, можна одержувати значення відсоткових ставок.

## Висновки

Отже, в роботі одержано дифузійне рівняння для моделювання відсоткових ставок при розгляді «економічної ситуації» як випадкової величини або однорідного марківського процесу.

## Список використаних джерел

- [1] Борисенко О.Д., Мішура Ю.С., Радченко В.М., Шевченко Г.М. Збірник задач з фінансової математики: навчальне видання. Київ: «Техніка», 2007.
- [2] Ширяев А.Н. Основы стохастической финансовой математики. Том I. Москва: ФАЗИС, 1998.
- [3] Липцер Р.Ш., Ширяев А.Н. Статистика случайных процессов. Москва: Наука, 1974.
- [4] Поттосина С.А., Алёхина А.Э. Математика рынка ценных бумаг (с элементами технического анализа): учебно-метод. пособие. Минск: БГУИР, 2012.

## АЛГОРИТМИ РЕАЛІЗАЦІЇ ТЕОРЕТИКО-МНОЖИННИХ ОПЕРАЦІЙ У ТАБЛИЧНИХ АЛГЕБРАХ

Робота присвячена дослідженню алгоритмів, що реалізують операції перетину, об'єднання та різниці у таблицях. Для кожної операції розглянуто базовий алгоритм та досліджувалися його модифікації з метою визначення найшвидшого алгоритму. Для усіх розглянутих алгоритмів знайдено точну складність у середньому формулами, що не містять  $O$ -асимптотики. Для експериментального підтвердження теоретичних результатів розроблено програмну систему, яка підтвердила знайдені теоретичні оцінки.

**Ключові слова:** алгоритм, складність, бази даних.

### Вступ

В наш час світ інформаційних технологій широко розповсюдився майже у всі сфери людської діяльності. Переважна більшість сучасних інформаційних систем у своїй роботі використовує бази даних, роботу з якими забезпечують системи управління базами даних, що реалізуються різноманітними програмними продуктами. Наразі спостерігається постійний зріст обсягів інформації, що підвищує вимоги для її зберігання та обробки у реальних базах даних. Однією з цих вимог є збільшення швидкодії роботи системи з користувачем. Для більшості інформаційних систем ця вимога зазвичай реалізується через оптимізацію запитів, які є основним інструментом взаємодії між користувачем і базою даних та й взагалі з системою.

Дотепер найбільш розповсюдженими залишаються реляційні бази даних, математична модель яких вперше була запропонована Е. Коддом [1]. Табличні алгебри, що були введені В.Н. Редьком і Д.Б. Буєм [2, 3], побудовані на основі алгебр Е. Кодда, суттєво їх уточнюють та складають теоретичний фундамент мов запитів сучасних табличних баз даних. Аналіз літератури щодо оптимізації операцій у реляційних базах даних показав, що у переважній більшості випадків розглядаються підходи, які дозволяють одержати оптимізацію лише операції з'єднання; не було знайдено жодної спроби оптимізації теоретико-множинних операцій перетину, об'єднання, різниці. Проте ці операції є важливими для табличних

(реляційних) алгебр та досить вживаними у запитах. Тому дослідження теоретико-множинних табличних операцій є важливою задачею, результати якої можуть бути використаними для оптимізації запитів.

У роботі досліджуються алгоритми, що реалізують перетин, об'єднання та різницю таблиць. У кожному з цих шести випадків спочатку розглядаються базові, найбільш природні алгоритми, що їх реалізують. У процесі дослідження теоретико-множинних табличних операцій було розглянуто значну кількість (приблизно 15 для кожного випадку) модифікацій базових алгоритмів, які могли б зменшити кількість обчислень; для кожного алгоритму знайдено значення середньої кількості обчислень. Було розроблено програмну систему, яка для таблиць із заданими параметрами (кількість рядків, атрибутів, потужність атрибутів), знаходить фактичну кількість виконаних обчислень за кожним із розглянутих алгоритмів. Численні експерименти підтвердили теоретичні оцінки складності усіх наведених алгоритмів.

## Основні означення

Зафіксуємо деяку непорожню множину  $A = \{A_1, \dots, A_p\}$ , елементи якої називаються атрибутами. Довільну скінченну підмножину  $R = \{A'_1, \dots, A'_n\} \subseteq A$  назовемо схемою. Рядком  $s$  схеми  $R$  називається множина пар  $s = \{(A'_1, d_1), \dots, (A'_k, d_k)\}$ , проекція якої за першою компонентою дорівнює  $R$ . Таблицею схеми  $R$  називається скінченна множина рядків схеми  $R$ . Кількість рядків у таблиці  $T$  позначатимемо  $|T|$ .

На множині всіх таблиць схеми  $R$  введено такі параметричні операції:

- 1) перетин  $\bigcap_R$  таблиць схеми  $R$  – таблиця, що складається з тих і лише тих рядків, які належать одночасно всім вихідним таблицям;
- 2) об'єднання  $\bigcup_R$  таблиць схеми  $R$  – таблиця, що складається з тих і лише тих рядків, які належать хоча б одній з вихідних таблиць;
- 3) різниця  $T_1 - T_2$  двох таблиць схеми  $R$  – таблиця, що складається з тих і лише тих рядків, які належать таблиці  $T_1$  та не належать таблиці  $T_2$ .

Іншими словами операції перетину, об'єднання і різниці таблиць є обмеженням відповідно теоретико-множинних операцій перетину, об'єднання і різниці на множині таблиць однакової схеми.

В роботі ми будемо порівнювати алгоритми за їх обчислювальною складністю. Слідуючи [4], обчислювальною складністю алгоритму називається функція залежності «обсягу» роботи, яка виконується їм, від «роз-

міру» вхідних даних. Розглядають два види обчислювальної складності: часову, яка оцінює час виконання алгоритму (кількість виконуваних обчислень), і ємнісну, яка оцінює кількість елементів пам'яті, необхідних для роботи алгоритму. У даній роботі будемо досліджувати часову складність та вважаємо, що складність елементарних операцій однакова (тобто використовується так звана рівномірна вага).

Виділяють три типи часової складності алгоритму: складність у найгіршому випадку (максимальна кількість елементарних операцій, що виконує алгоритм), складність для майже всіх входів і складність в середньому (математичне сподівання кількості елементарних операцій, що виконує алгоритм). Дослідження складності алгоритму в найгіршому випадку істотно простіше, ніж дослідження інших двох типів складності, проте при аналізі часової складності обробки інформації в системах керування базами даних та обробки запитів найбільш прийнятним є використання складності в середньому. У нашій роботі швидкодія алгоритмів визначається саме за їхньою складністю в середньому.

## Теоретичні оцінки складності розглянутих алгоритмів

У якості базових розглянемо такі алгоритми перетину, об'єднання та різниці таблиць.

**Алгоритм БПТ** З кожним рядком  $s_1$  таблиці  $T_1$  порівнюємо всі рядки таблиці  $T_2$ : якщо існує такий рядок  $s_2 \in T_2$ , що  $s_1 = s_2$ , то рядок  $s_1$  додаємо до таблиці  $T$ , у якій міститься результат, та переходимо до наступного рядка таблиці  $T_1$ .

**Алгоритм БОТ** Додаємо всі рядки таблиці  $T_1$  до нової таблиці  $T$ , яка містить результат, а потім всі рядки  $s_2$  таблиці  $T_2$  порівнюємо з рядками таблиці  $T$ , і якщо не існує такого рядка  $s_1$ , що  $s_2 = s_1$ , то рядок  $s_2$  додаємо до таблиці  $T$ , а якщо такий рядок існує, то переходимо до наступного рядка таблиці  $T_2$ .

**Алгоритм БРТ** З кожним рядком  $s_1$  таблиці  $T_1$  порівнюємо всі рядки таблиці  $T_2$ : якщо не існує такого рядка  $s_2 \in T_2$ , що  $s_1 = s_2$ , то рядок  $s_1$  додаємо до нової результуючої таблиці  $T$ , а якщо такий рядок існує, то переходимо до наступного рядка таблиці  $T_1$ .

У процесі дослідження теоретико-множинних табличних операцій ми розглядали різні модифікації базових алгоритмів, які могли б зменшити кількість обчислень. У якості критерія оцінювання швидкодії алгоритмів розглядалася їх складність у середньому для найбільш загального, на наш погляд, випадку, за яким домен кожного атрибуту схеми таблиць

є фіксованим і заздалегідь відомим, та розподіл значень за кожним атрибутом в кожній таблиці є рівномірним. Іншими словами, для будь-яких атрибуту  $A \in R$  та рядка  $s$  будь-якої таблиці кожний елемент множини  $dom_A = \{d_1, \dots, d_n\}$  має однакову ймовірність бути значенням  $A$  в  $s$ :  $(A, d) \in s \rightarrow P\{d = d_1\} = P\{d = d_2\} = \dots = P\{d = d_n\}$ .

Найбільш швидкими виявилися такі модифікації базових алгоритмів.

**Алгоритм ШПТ** Якщо  $|T_1| \leq |T_2|$ , то з кожним рядком  $s_1$  таблиці  $T_1$  порівнюємо всі рядки таблиці  $T_2$ , і якщо не існує такого рядка  $s_2 \in T_2$ , що  $s_1 = s_2$ , то вилучаємо рядок  $s_1$  з таблиці  $T_1$ , а якщо такий рядок  $s_2$  існує, то вилучаємо його з таблиці  $T_2$  та переходимо до наступного рядка таблиці  $T_1$ ; результат міститься в таблиці  $T_1$ . Якщо ж  $|T_1| > |T_2|$ , то таблиці  $T_1$  і  $T_2$  міняються місцями.

**Алгоритм ШОТ** Якщо  $|T_1| \leq |T_2|$ , то з кожним рядком  $s_2$  таблиці  $T_2$  порівнюємо всі рядки таблиці  $T_1$ , і якщо існує такий рядок  $s_1 \in T_1$ , що  $s_2 = s_1$ , то вилучаємо рядок  $s_1$  з таблиці  $T_1$  і переходимо до наступного рядка таблиці  $T_2$ ; після цього додаємо всі рядки, що залишились, з таблиці  $T_1$  до таблиці  $T_2$ , в якій міститься результат. Якщо ж  $|T_1| > |T_2|$ , то таблиці  $T_1$  і  $T_2$  міняються місцями.

**Алгоритм ШРТ** З кожним рядком  $s_2$  таблиці  $T_2$  порівнюємо всі рядки таблиці  $T_1$ : якщо існує такий рядок  $s_1 \in T_1$ , що  $s_2 = s_1$ , то вилучаємо  $s_1$  з таблиці  $T_1$ , у якій міститься результат, і переходимо до наступного рядка таблиці  $T_2$ .

Знайдемо складність у середньому алгоритмів БПТ, БОТ, БРТ, ШПТ, ШОТ, ШРТ.

Нехай  $|T_1| = m_1$ ,  $|T_2| = m_2$ ,  $|A_1| = q_1, \dots, |A_n| = q_n$ . Позначимо  $m = \min\{m_1, m_2\}$  та  $Q = q_1 \cdot \dots \cdot q_n$ . Існує  $Q$  фіксованих рядків, кожний з яких незалежно від решти інших може належати чи не належати кожній з таблиць-аргументів. Нескладно бачити, що у випадку  $m_1 + m_2 \leq Q$  мінімальна кількість рядків таблиці  $T_1 \cap_R T_2$  дорівнює 0, у протилежному випадку мінімальна кількість рядків таблиці  $T_1 \cap_R T_2$  дорівнює  $m_1 + m_2 - Q$ , це число позначимо через  $z$ , максимально можлива кількість рядків таблиці  $T_1 \cap_R T_2$  становить  $m$ . Оскільки всі запропоновані алгоритми використовують порівняння рядків в таблицях  $T_1$  та  $T_2$  на предмет їх рівності, та складність усіх цих алгоритмів залежить від кількості рядків таблиці  $T_1 \cap_R T_2$ , то спочатку було знайдемо значення ймовірності  $P(j)$  того, що кількість таких рядків буде дорівнювати  $j$  ( $j \in \{z, \dots, m\}$ ). Далі для кожного алгоритму було знайдено значення  $W(j)$  середньої кількості об-

числень за умови  $|T_1 \cap_R T_2| = j$ . Будемо вважати складністю алгоритму число

$$W = \sum_{j=z}^m P(j) \cdot W(j)(1).$$

Усі наступні рівності роботи доведено у [5].

$$P(j) = \frac{m_1! \cdot m_2! \cdot (Q - m_1)! \cdot (Q - m_2)!}{j! \cdot (m_1 - j)! \cdot (m_2 - j)! \cdot (Q - m_1 - m_2 + j)! \cdot Q!}$$

Для знаходження  $W(j)$  було знайдемо середню кількість обчислень  $W'$ , необхідну для порівняння рядків  $s^1 = \{(A_1, d_1^1), \dots, (A_n, d_n^1)\} \in T_1$  та  $s^2 = \{(A_1, d_1^2), \dots, (A_n, d_n^2)\} \in T_2$  у випадку, коли  $s^1 \neq s^2$ .

$$W' = \frac{Q}{Q-1} \cdot \left( 1 \cdot \frac{q_1 - 1}{q_1} + 2 \cdot \frac{q_2 - 1}{q_1 \cdot q_2} + \dots + n \cdot \frac{q_n - 1}{Q} \right).$$

Для алгоритмів БПТ, ШПТ, БОТ, ШОТ, БРТ, ШРТ значення  $W(j)$  дорівнюють відповідно  $(m_1 m_2 - \frac{j}{2}(m_2 + 1)) W' + 2jn$ ,  $(m_1 m_2 - \frac{j}{2}(m_1 + m_2 + \frac{1}{2} - \frac{j}{2})) W' + jn + \min\{m_1, m_2\}n + 1$ ,  $(m_1 m_2 - \frac{j}{2}(m_2 + 1)) W' + m_1 n + m_2 n$ ,  $(m_1 m_2 - \frac{j}{2}(m_1 + m_2 + \frac{1}{2} - \frac{j}{2})) W' + \min\{m_1, m_2\}n + jn + 1$ ,  $(m_1 m_2 - \frac{j}{2}(m_2 + 1)) W' + m_1 n$ ,  $(m_1 m_2 - \frac{j}{2}(m_1 + m_2 + \frac{1}{2} - \frac{j}{2})) W' + 2jn$ .

Оскільки одержані формули складності алгоритмів є досить громіздкими, їх аналіз зробимо у наступному розділі цієї роботи.

## Експериментальне порівняння швидкодії алгоритмів

Для експериментального підтвердження теоретичних оцінок складності було розроблено програмну систему, яка для таблиць із заданими параметрами (кількість рядків, атрибутів, потужність атрибутів), знаходить фактичну кількість виконаних обчислень за кожним із запропонованих алгоритмів та порівнює їх із знайденими теоретичними оцінками, а також знаходить середнє значення фактичної кількості виконаних обчислень для будь-якої серії експериментів. Багаточисленні експерименти підтвердили теоретичні оцінки складності в середньому алгоритмів для таблиць, точність знайдених теоретичних оцінок виявилась достатньо високою, у жодній серії експериментів відхилення не перевищувало 0,25%. У Табл. 1 наведемо дані з двох серій експериментів: перша серія складалася з 500 експериментів з параметрами  $n = 10$ ,  $|T_1| = |T_2| = 1000$ ,  $|A_1| = 3$ ,  $|A_2| = \dots = |A_{10}| = 2$ , друга – з 2000 експериментів з параметрами  $n = 5$ ,  $|T_1| = |T_2| = 100$ ,  $|A_1| = |A_2| = |A_3| = 3$ ,  $|A_4| = |A_5| = 4$ .

| Алгоритм | Кількість обчислень у серії I |           |         | Кількість обчислень у серії II |           |         |
|----------|-------------------------------|-----------|---------|--------------------------------|-----------|---------|
|          | теорет.                       | експерим. | відх. % | теорет.                        | експерим. | відх. % |
| БПТ      | 1132071,9                     | 1131741,0 | 0,029   | 13324,34                       | 13336,58  | 0,092   |
| ШПТ      | 771415,3                      | 771135,0  | 0,036   | 12205,89                       | 12231,50  | 0,210   |
| БОТ      | 1139051,1                     | 1139279,6 | 0,020   | 14092,86                       | 14094,35  | 0,011   |
| ШОТ      | 771415,3                      | 771304,5  | 0,014   | 12205,89                       | 12212,60  | 0,055   |
| БРТ      | 1129051,1                     | 1128895,6 | 0,014   | 13592,86                       | 13595,13  | 0,017   |
| ШРТ      | 767924,7                      | 767386,5  | 0,070   | 11820,63                       | 11819,86  | 0,007   |

Табл. 1: Експериментальні значення складності.

## Висновки

У роботі досліджено алгоритми, що реалізують теоретико-множинні операції на таблицях. Знайдено точну часову складність у середньому розглянутих алгоритмів, завдяки чому вдалося визначити найшвидші модифікації базових алгоритмів на таблицях. Для експериментального цих результатів розроблено програмну систему, експерименти з якою підтвердили знайдені теоретичні оцінки. Результати можуть бути використані для оптимізації запитів у реляційних базах даних та для зменшення часу обробки інформації у системах управління базами даних. За темою дослідження в подальшому планується дослідити алгоритми, що реалізують інші табличні операції, а також алгоритми, що реалізують теоретико-множинні операції із використанням сортування, індексації, хешування та специфікації таблиць.

## Список використаних джерел

- [1] Codd E.F. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*. 1970. Vol. 13, No. 6. P. 377–387.
- [2] Red'ko V.N., Bui D.B. Foundations of the theory of relational database models. *Cybernetics and Systems Analysis*. 1996. Vol. 32, Issue 4. P. 471–478.
- [3] Редько В.Н., Брона Ю.Й., Буй Д.Б., Поляков С.А. Реляційні бази даних: табличні алгебри та SQL-подібні мови. Київ: Академперіодика, 2001.
- [4] Буй Д.Б., Скобелев В.Г. Сложность операций в базах данных (обзор). *Радиоэлектронні і комп'ютерні системи*. 2014. № 6. С. 53–59.
- [5] Канарська І.С. Алгоритми, що реалізують теоретико-множинні операції на таблицях та мультитаблицях. *Праці Інституту прикладної математики і механіки НАН України*. 2019. Т. 33. С. 100–111.

<sup>1</sup> д. пед. н., професор кафедри математики та інформатики Вінницького державного університету імені Михайла Коцюбинського

<sup>2</sup> магістрант факультету математики, фізики і комп'ютерних наук Вінницького державного університету імені Михайла Коцюбинського

e-mail: <sup>1</sup> klochkoob@gmail.com, <sup>2</sup> berezov.dima@ukr.net

## ІННОВАЦІЙНІ НАВИЧКИ ЯК КОМПЕТЕНТІСНИЙ ШЛЯХ РОЗКРИТТЯ ПОТЕНЦІАЛУ ОСВІТНЬОЇ РОБОТОТЕХНІКИ

Проаналізовано напрями перспективного розвитку професій відповідно до актуальних технологічних тенденцій трансформації виробництва. З'ясовано, що, відповідно до цих трансформацій та соціального замовлення, завданням освіти є формування та розвиток нових – інноваційних навичок. Введено означення інноваційних навичок в галузі освітньої робототехніки. Визначено інноваційні навички в галузі освітньої робототехніки, які необхідно формувати та розвивати у здобувачів середньої освіти.

**Ключові слова:** освітня робототехніка, середня освіта, STEM-освіта, інноваційні навички 21 століття, Нова українська школа.

### Вступ

Темпи впровадження інноваційних технологій в соціально-економічну сферу сьогодні залишаються незмінними та прискорюються в багатьох областях господарської системи. Цей процес відбувається інтеграційно з наукою та освітою – важливими факторами соціально-економічного розвитку країни. Пріоритетними для лідерів світової економіки в сучасних умовах залишаються хмарні обчислення, штучний інтелект, машинне навчання, негуманоїдна робототехніка, електронна комерція, кібербезпека [1]. Відповідно до прогнозів, складених за даними міжнародного опитування підприємств, частка негуманоїдних роботів та роботів-гуманоїдів, які будуть впроваджені у 2025 році, ймовірно, складе 10 % [1].

Зростаючим попитом серед роботодавців користуються інженери-робототехніки, Інтернету речей, спеціалісти зі штучного інтелекту, машинного навчання, автоматизації процесів [2] (рис. 1). Одним із напрямів перспективного розвитку цих професій є їх інтегроване поєднання в сфері робототехніки. На нашу думку та на думку багатьох науковців, дослідників розвитку ринку праці, ті технології, які були на початкових етапах розвитку натеper розвиваються прискореними темпами [3]. Однією з причин такого стану системи зайнятості населення є пандемія COVID-19, яка

спровокувала появу прискореного перерозподілу праці між людьми та робототехнічними системами та спричинила встановлення нової рівноваги на ринку робочих місць.



Рис. 1: Технологічні тенденції трансформації виробництва.

Перед майбутніми фахівцями постають нові виклики адаптації до ринку праці до нових професійних завдань, інтегрований зміст яких формується під впливом актуального суспільного замовлення, поєднує різні галузі, потребує формування інноваційних навичок.

## Результати дослідження

Досягнення в галузі технологій сприяють створенню нових професій та потребують адаптації до змін усіх стейкхолдерів [4, 5]. Одним зі шляхів вирішення даної проблеми підготовки фахівців нового покоління є впровадження метадисциплін, зміст яких є інтегрованим [3]. Викладання метадисциплін усуває традиційні бар'єри між предметами, є орієнтованим на інновації та прикладний зміст, спрямований на вирішення складних проблем із використанням сучасних технологій.

Одним із таких підходів є STEM-освіта, до якої відносимо й освітню робототехніку [6]. STEM-освіта інтегровано поєднує технології, інженерію, природничі та математичні науки, передбачає проведення наукових досліджень та інженерного проектування. Залучення учнів та студентів до високоякісної STEM-освіти потребує, детальної розробки навчальних програм, проведення кваліфікованого інструктажу.

Завданням освіти, відповідно до цих трансформацій та соціального замовлення, є формування та розвиток нових - інноваційних навичок, з метою підготовки учнів закладів загальної середньої освіти [7, 8] та студентів закладів вищої освіти до майбутнього трудового життя. Зазвичай, до інноваційних навичок відносять навички 21 століття [7 – 9]. На нашу думку, інноваційними навичками в області освітньої робототехніки є інтегровані когнітивні й практичні здатності особистості до інноваційної діяльності в галузі освітньої робототехніки. Така діяльність передбачає використання актуальних досягнень освіти, науки та техніки в різних сферах людської діяльності, а саме, техніки, інженерії, цифрових технологій, математики, організації праці та інших сферах, інтегрованих в процесі розробки робототехнічних систем.

Розглядаємо інноваційність, як ключову компетентність, відповідно до Державного стандарту базової середньої освіти [8], Концептуальних засад реформування середньої школи «Нова українська школа» [7]. Визначимо інноваційні уміння/ставлення в галузі освітньої робототехніки з точки зору інтеграції інноваційності математичної, інформатичної, технологічної, природничої освітніх галузей. Серед умінь/ставлень, які необхідно формувати та розвивати у здобувачів базової середньої освіти, з метою здійснення ефективної інноваційної діяльності виділяємо (рис. 2):

- уміння [8]: «генерувати нові ідеї щодо розв'язання проблемної ситуації, аналізувати та планувати їх втілення; знаходити, обробляти, зберігати інформацію природничого змісту, перетворювати її з одного виду на інший з використанням інформаційно-комунікаційних технологій; трансформувати здобуті знання про матеріали, технології та обладнання у вдосконалення технологічного процесу, новий чи покращений продукт (послугу) з новими якостями, що використовується у практичній діяльності, чи новий підхід до реалізації соціальних послуг, адаптувати їх до нагальних потреб суспільства; генерувати та реалізовувати ідеї з використанням цифрових технологій; використовувати та створювати цифровий контент природничого змісту; досліджувати довкілля за допомогою сучасних інформаційно-комунікаційних технологій; наводити приклади реалізації»

лізації інноваційних ідей у різних предметних галузях і життєвих ситуаціях з використанням інформаційно-комунікаційних технологій; розпізнавати та описувати поширення цифрових інновацій у науці і суспільстві»;

- ставлення [8]: «відкритість до інновацій, позитивне оцінювання та підтримка конструктивних ідей інших осіб; відкритість новому у сфері інформаційних технологій, готовність до змін, прийняття неочікуваних результатів під час здійснення інформаційних процесів; усвідомлення важливості інновацій у технологічних процесах для сталого розвитку суспільства; критичне оцінювання інформації природничого змісту, здобутої з різних джерел; готовність до інтеграції знань з різних предметних галузей під час створення інформаційних продуктів, навчання; дотримання авторського права, принципів академічної доброчесності та етичної взаємодії у віртуальному просторі».

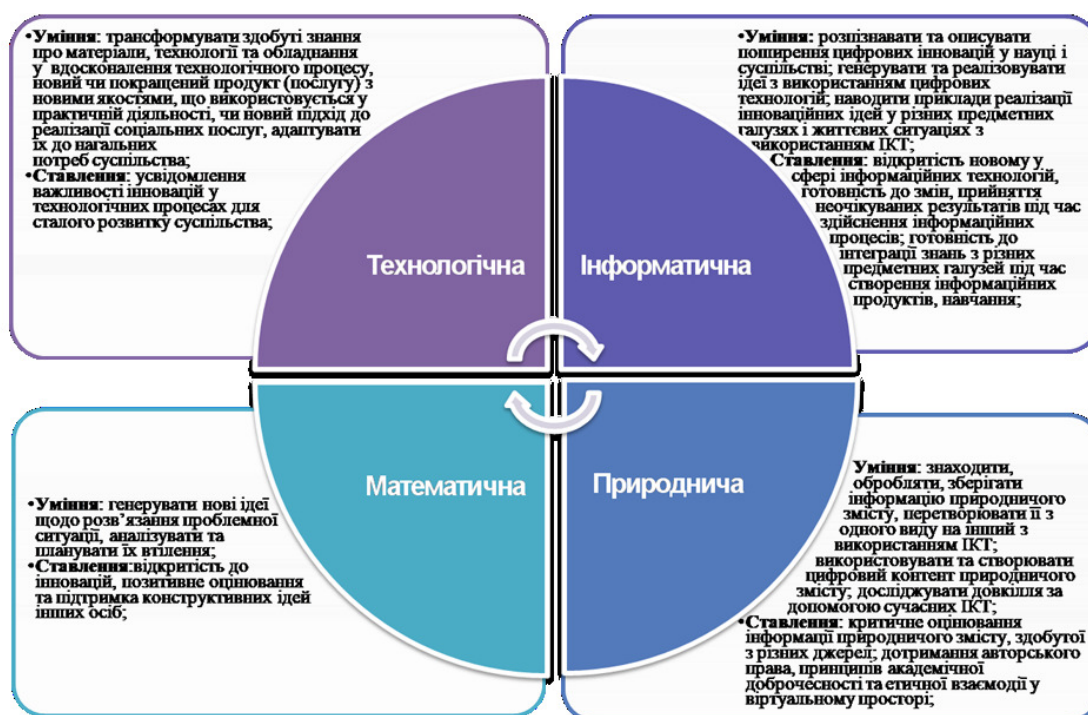


Рис. 2: Інноваційні уміння/ставлення в галузі освітньої робототехніки.

## Висновки

Набуття інноваційних навичок в галузі освітньої робототехніки, яка стрімко розвивається та може суттєво вплинути на природу технічної та наукової освіти на всіх рівнях, сприятиме творчому розвитку особистості, самовдосконаленню, самоосвіті, формуванню життєвого досвіду, кон-

курентоспроможності майбутніх фахівців, а також впровадженню Концептуальних засад реформування середньої школи «Нова Українська школа», навчання протягом усього життя, навичок 21-го століття, забезпечення сталого розвитку.

## Список використаних джерел

- [1] The Future of Jobs Report 2020. *World Economic Forum*, 2020.
- [2] White Paper: The Next Economic Growth Engine Scaling Fourth Industrial Revolution Technologies in Production. In collaboration with McKinsey & Company. *World Economic Forum*, 2018.
- [3] Клочко О.В. Теоретичні і методичні засади професійної підготовки майбутніх менеджерів аграрного виробництва засобами сучасних інформаційно-комунікаційних технологій: дис. ... д-ра педагогічних наук: 13.00.04 / Вінницький державний педагогічний університет імені Михайла Коцюбинського. Вінниця, 2018.
- [4] Klochko O.V., Nagayev V.M., Klochko V.I., Pradivliannyi M.G., Didukh L.I. Computer oriented systems as a means of empowerment approach implementation to training managers in the economic sphere. *Information Technologies and Learning Tools*. 2020. Vol 68, No. 6. P. 33–46.
- [5] Клочко О.В. Методична система навчання студентів геоінформаційним технологіям. *Вісник Луганського національного університету імені Т.Г. Шевченка. Педагогічні науки*. 2010. № 17 (204). С. 40–50.
- [6] Концепція розвитку природничо-математичної освіти (STEM-освіти). 2021. URL: <https://zakon.rada.gov.ua/laws/show/960-2020-r#Text>.
- [7] Нова українська школа : концептуальні засади реформування середньої школи. 2016. URL: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/nova-ukrainska-shkola-compressed.pdf>
- [8] Державний стандарт базової середньої освіти. 2020. URL: <https://www.kmu.gov.ua/npas/pro-deyaki-pitannya-derzhavnih-standartiv-povnoyi-zagalnoyi-serednoyi-osviti-i300920-898>.
- [9] Klochko O., Fedorets V., Maliar O., & Hnatuyk V. The use of digital models of hemodynamics for the development of the 21st century skills as a components of healthcare competence of the physical education teacher. *E3S Web of Conferences. EDP Sciences*. 2020. Vol. 166. Article 10033.

<sup>1</sup> магістрант факультету прикладних комп'ютерних технологій Національної металургійної академії України, Дніпро

<sup>2</sup> к. т. н., доцент кафедри інформаційних технологій та систем Національної металургійної академії України, Дніпро

*e-mail:* <sup>1</sup> korotchikov1@gmail.com, <sup>2</sup> tatyana-mikhaylovskaya@gmail.com

## **МЕТОДОЛОГІЯ РОЗРОБКИ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ДЛЯ РАНЖУВАННЯ РЕКЛАМНОГО КОНТЕНТУ**

Наведено опис вимог до підсистеми для ранжування рекламного контенту, а також основні складові такої підсистеми.

**Ключові слова:** *web-застосунок, реклама, контент, ранжування, системний аналіз.*

### **Вступ**

В даний час реклама є важливою частиною бізнесових компаній, які намагаються знайти вигідну та працюючі способи рекламувати свій продукт. Мета роботи – розглянути методологію розробки підсистеми для ранжування рекламного контенту, завдяки якій можна покращити релевантність інформації та розробити сервіс, що адаптує процес пошуку інформації до індивідуальних особливостей користувача.

### **Підсистеми для ранжування рекламного контенту**

Основними модулями такої підсистеми є:

- ядро;
- засоби аналізу та просування інтернет-ресурсу;
- модуль організації взаємодії з користувачем;
- модуль ідентифікації та автентифікації.

Ядро системи – основний модуль системи, в якому опрацьовуються всі операції та роботи з її базою даних. Ядро системи координує роботу решти модулів з метою формування цілісної системи. Особливою функцією зазначеного модуля є організація процесу введення вхідної інформації, її опрацювання та збереження [1]. Користувач займається розміщенням контенту та отримує кошти. Засоби аналізу інтернет-ресурсу містять коман-

ди, що реалізують: експрес-аналіз, аналіз та перевірку PR та тематичного індексу цитування (тІЦ), аналіз контенту, а менеджер визначає позицію рекламного контенту.

Система ICWTADS (рис. 1) надає два рівні доступу:

- менеджер (адміністратор) – модерує зміст рекламного контенту, перевіряє відповідність рекламного майданчика вимогам для розміщення рекламного контенту;
- користувач – реєструє рекламний майданчик (рис. 2 а) або розміщує рекламний контент у вигляді рекламного банеру (рис. 2 б).



Рис. 1: Інтерфейс ICWTADS (<https://icwt.io/>).

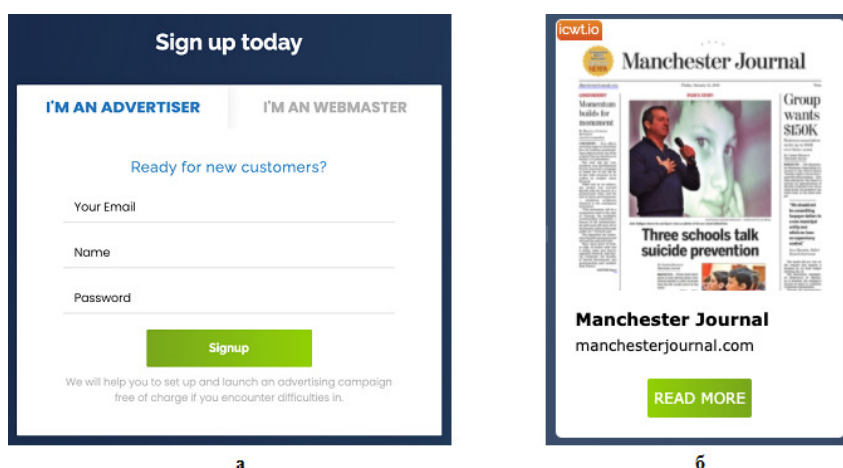


Рис. 2: Інтерфейс користувача системи ICWTADS: а) реєстрація рекламного майданчика; б) приклад рекламного контенту.

Для проектування системи підтримки прийняття рішень при просуванні інформаційного порталу по пошукових системах важливою вимогою є надання рекомендацій, які ґрунтуються на принципах системного аналізу.

Застосування підходів притаманних системному аналізу дає змогу відобразити особливості поведінки складних системи та врахувати важливі взаємні зв'язки між її елементами.

Опис сутності розроблений системи представлений за допомогою дерева цілей [2], яке зображено на рис. 3. Методологія системного аналізу застосовується для ранжування рекламного контенту, для обробки даних використовувалися такі методи, як кількісний і якісний аналіз. Також застосовувалися спеціальні способи кількісної обробки даних. Створення програми з модулів визначає виділення набору окремих компонентів, що взаємозалежні і мають добре визначені інтерфейсні зв'язки з іншими компонентами програми. Вони визначають форму всіх взаємодій між ними, проте не містять з'єднань на їх внутрішню структуру. Керування базами даних здійснюватиметься у двох аспектах: керування внутрішніми та зовнішніми базами [3].

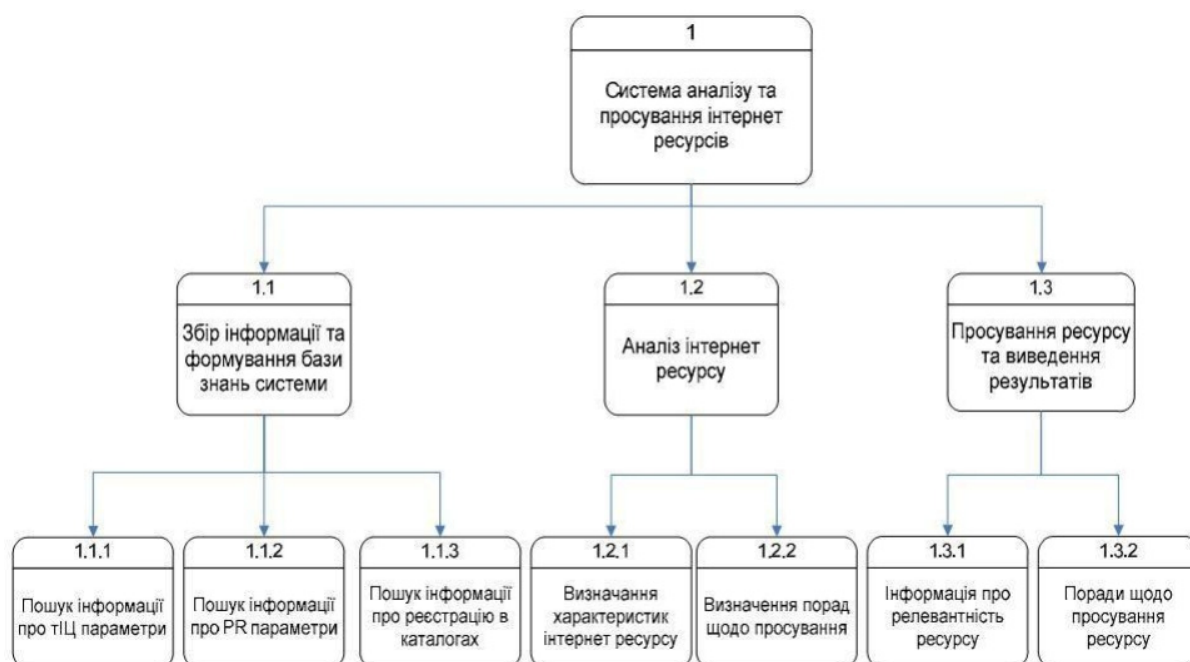


Рис. 3: Дерево цілей підсистеми для ранжування рекламного контенту.

У кожному конкретному випадку наше програмне забезпечення використовує шлях і вирішує ті завдання, які оптимальною мірою відповідають досягненню отримати прибуток та підібрати правильно контент, щоб отримати свою цільову аудиторію. Істотними чинниками ранжування рекламного контенту є і ключові слова та категорії. Також якісний контент є дуже важливим для отримання високих позицій. До факторів, які негативно впливають на ранжування рекламного контенту, віднесено низьку ціну в порівнянні з іншими, що знижує ймовірність його показу. Комплексне використання усіх цих факторів підвищить позиції. Для рекламного

контенту слід визначити прогнозовану реакцію цільової аудиторії, яку бажає досягти рекламодавець на різних етапах рекламної кампанії. Також було розглянуто платформи для здійснення взаємодії з метою ранжування контенту, що розглядається як завдання з підвищення ефективності реклами.

Система має свій фінансовий обліковий запис для кожного користувача, списування коштів з рахунку рекламодавця відбувається автоматично після перегляду кимось його реклами і автоматично потрапляють на обліковий запис користувача який надав майданчик. Реклами та майданчики проходять стадію модерації адміністратором, щоб відсікти неприйнятний контент.

## **Висновки**

У роботі досліджено методологію розробки підсистеми для ранжування рекламного контенту, яка забезпечує взаємодію між рекламними майданчиками та рекламодавцями з метою підвищення ефективності реклами.

## **Список використаних джерел**

- [1] Тарасов О.В., Федько В.В., Лосєв М.Ю. Організація баз даних та знань. Проектування баз даних: навч.-практ. посібник. Харків: ХНЕУ, 2011.
- [2] Берко А.Ю., Верес О.М., Пасічник В.В. Системи баз даних і знань. Книга 1. Організація баз даних і знань: навч. посібник. Львів: Магнолія, 2011.
- [3] Білас О.Є. Якість програмного забезпечення та тестування: навч. посібник. Львів: Львівська політехніка, 2013.

**Крак Ю.В., Пашко А.О., Стеля О.Б.**

<sup>1</sup> ч-к. НАНУ, д. ф.-м. н., п.н.с. Інституту кібернетики імені В.М. Глушкова НАН України, завідувач кафедри теоретичної кібернетики Київського національного університету імені Тараса Шевченка

<sup>2</sup> д. ф.-м. н., професор кафедри теоретичної кібернетики Київського національного університету імені Тараса Шевченка

<sup>3</sup> к. ф.-м. н., завідувач НДЛ «Обчислювальних методів в механіці суцільних середовищ» Київського національного університету імені Тараса Шевченка

e-mail: <sup>1</sup> krak@univ.kiev.ua, <sup>2</sup> aap2011@ukr.net, <sup>3</sup> oleg.stelya@gmail.com

## **ПОСЛІДОВНИЙ АНАЛІЗ ЕКГ-СИГНАЛІВ**

В роботі досліджується використання методів статистичного послідовного аналізу для обробки електрокардіограм. На сьогоднішній день актуальною є обробка електрокардіограм в реальному часі, коли немає можливості записувати результати на носій, впродовж великого проміжку часу: від декількох годин – до декількох діб. За основу обробки даних в реальному часі взято послідовний аналіз Вальда. Результати дослідження опробовані на реальних даних.

**Ключові слова:** *послідовний аналіз, електрокардіограма, QRS – комплекс.*

## **Основи обробки електрокардіограми**

Електрокардіограма (ЕКГ) використовується для діагностики серцевих захворювань. ЕКГ хорошої якості використовуються лікарями для інтерпретації та виявлення фізіологічних та патологічних явищ.

Електрокардіограма - це запис коливань різниць потенціалів, яка виникає у серці під час його збудження. ЕКГ є незамінним у діагностиці порушень ритму і провідності, гіпертрофії, ішемічної хвороби серця. Однак, у реальних ситуаціях записи ЕКГ часто спотворюються різними шумами та завадами, серед яких найчастіше відбуваються такі:

- високочастотний шум, що викликаний електроміограмою, перешкодами від ліній електропередач, або механічні сили, що діють на електроди;
- базове відхилення, яке може бути пов'язане з диханням чи рухом пацієнтів та інструментів.

Ці шуми та завади серйозно обмежують якість записаних ЕКГ, тому їх необхідно видалити для кращої клінічної оцінки.

У роботі продовжуються дослідження, представлені в роботах [1-4], де розглядалися завдання очищення ЕКГ, видалення базової лінії тренду,

фільтрація сигналу, виділення основних параметрів ЕКГ для подальшої обробки.

## Алгоритм послідовного аналізу Вальда

При дослідженні процесів у техніці, економіці, біології та інших галузях часто зустрічаються процеси, що міняють свої характеристики у випадкові моменти часу – моменти «розладів». Виявлення таких розладів швидше і точніше дозволяє проводити управління спостереженнями в стохастичних системах та управляти об'єктами в умовах неповної інформації.

Проблема найкращого вибору вперше були представлена у роботах відомого американського статистика А.Вальда у задачах послідовного розпізнавання [5]. А. Вальдом розглядалися послідовності незалежних однаково розподілених випадкових величин з невідомим законом розподілу. Методами математичної статистики визначався тип розподілу цих спостережень. Кількість спостережень заздалегідь не фіксується, а моменти зупинок мають випадковий характер і визначаються дослідником залежно від значень величин, що спостерігаються.

Нехай  $\xi$  - випадкова величина, що приймає значення  $0, 1, \dots$ , а спостереження  $y_1, y_2, \dots$ , такі, що за умови  $\xi = n$ , величини  $y_1, y_2, \dots, y_{n-1}$  незалежні, однаково розподілені та мають функцію розподілу  $F_0(x)$ . Випадкові величини  $y_n, y_{n+1}, \dots$  – також незалежні та однаково розподілені, але мають функцію розподілу  $F_1(x) \neq F_0(x)$ . Іншими словами, в моменти  $\xi$  відбувається зміна імовірнісних показників у процесі, що спостерігається. Виникає завдання, як, послідовно спостерігаючи  $y_1, y_2, \dots$ , вирішити питання про те, в який момент слід зробити висновок що «розлад» відбувся. Іншими словами, треба визначити зміну ймовірнісних характеристик, але так, щоб з одної сторони уникнути помилки визначення «розладу» в той момент, коли він ще насправді не відбувся, а з іншого боку – мінімізувати час між виникненням «розладу» та його фактичним виявленням. Задача розглядається як задача перевірки гіпотези. У роботах [5-7] досліджені та отримані оцінки моментів розладу.

## Результати експериментальних досліджень

Відведення біопотенціалів серця з певних ділянок поверхні тіла людини відбувається за допомогою електродів, співвідношення яких один з одним або з електрокардіографом формує систему електрокардіографічних відведень. Зазвичай використовують 12 відведень (існує близько 42 варіан-

тів). Взагалі майже всі виміри робляться в другому відведенні, а всі інші використовуються для діагностики конкретних ділянок серця, так, наприклад, перше відведення дає чітку інформацію про передні стінки серця, а третє відведення робить акцент на задніх стінках серця.

В реальному часі немає можливості використовувати багато каналів. Це, як правило, один або два канали. На тіло хворої людини наклеюють спеціальний пристрій, який знімає біопотенціал серця. Дискретність сигналу поряка 5-10 вимірів в секунду (Гц): на один удар серця приходить в середньому 400 вимірів. Як правило, записувати цю інформацію для подальшої обробки немає можливостей, необхідно її обробляти в реальному часі.

При нормальній роботі серця статистичні характеристики сигналу з плином часу міняються дуже повільно.

Аритмія – група порушень діяльності серця, пов'язаних з розладом ритмічності, послідовності та сили скорочень серцевого м'яза. Тому, статистичні характеристики будуть мінятись набагато швидше.

Також актуальною є задача визначення моменту зміни статистичних характеристик.

В медицині виділяють 11 видів аритмії. Для здорової людини пульс складає 60-80 ударів за хвилину. Пульс для деяких видів аритмії:

- При миготливій аритмії (AFIB) – 400 ударів за хвилину;
- При брадикардії (SB) –  $< 60$  ударів за хвилину;
- При тахікардії (ST) – 70-130 ударів за хвилину;
- При тріпотінні (AF) – 600 ударів за хвилину.

В стаціонарному режимі для аналізу ЕКГ можна використовувати методи інтелектуального аналізу. Використання цих методів дозволяє виділити основні компоненти ЕКГ, провести аналіз QRS комплексів та PP-комплексів, що є визначальними при діагностуванні аритмії. В режимі реального часу використання цих методів не завжди можливе.

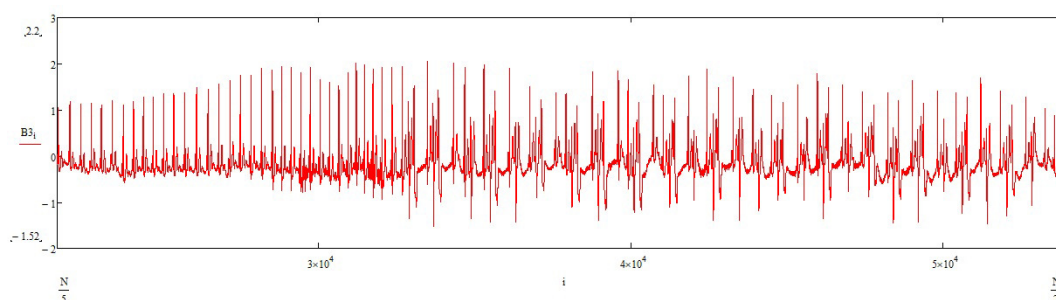


Рис. 1: Фрагмент кардіограми.

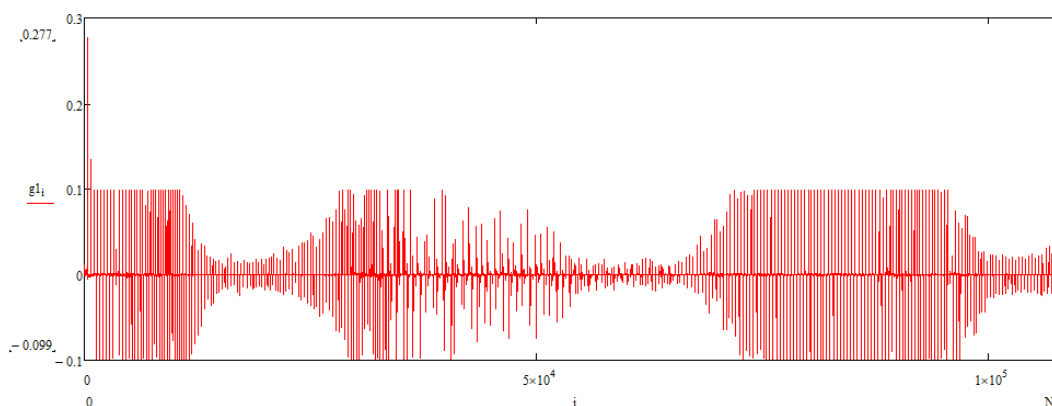


Рис. 2: Дисперсія кардіограми.

Для дослідження використовувалась кардіограма довжиною 270 хвилин. На рис. 1. зображено фрагмент кардіограми, на якому видно зміни в роботі серця.

Обробка кардіограми велась при допущенні, що збір даних ведеться в реальному часі. Оброблялись дані довжиною три хвилини.

На рис. 2. зображена поведінка дисперсії. Перед обчисленням дисперсії сигнал попередньо нормувався. Нормування проводилось за рекурентними формулами. Для оцінки часу настання «розладу» досліджувався закон розподілу поведінки фрагменту дисперсії і перевірялась гіпотеза про рівність розподілів.

## Висновки

Поведінка дисперсії дозволяє виділити періоди збоїв у роботі серця, а також визначити час розладу. Це дозволяє вчасно інформувати пацієнта про зміни в роботі серця, що дозволить пацієнтові вчасно прийняти ліки, або виконати інші рекомендації лікаря. Запропонований підхід використовується для обробки кардіограми в реальному часі. Розроблені алгоритми можна використовувати в різних гаджетах, якими користуються зацікавлені споживачі.

## Список використаних джерел

- [1] Krak I.V., Stelia O.B., Pashko A.O., Efremov M.S., Khorozov O.A. Electrocardiogram Classification Using Wavelet Transformations. In *2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)* (Lviv-Slavske, Feb. 25-29, 2020). 2020. P. 930–933.

- [2] Krak I.V., Pashko A.O., Khorozov O.A., Stelia O.B. Physiological Signals Analysis, Recognition and Classification Using Machine Learning Algorithms. *CEUR Workshop Proceedings*. 2020. Vol. 2608. P. 955–965.
- [3] Krak I.V., Pashko A.O., Stelia O.B., Barmak O.V., Pavlov S.V. Selection Parameters in the ECG Signals for Analysis of QRS Complexes. *CEUR Workshop Proceedings*. 2020. Vol. 2623. P. 1–13.
- [4] Pashko A.O., Krak I.V., Stelia O.B., Khorozov O.A. Isolation of Informative Features for the Analysis of QRS Complex in ECG Signals. In book *Lecture Notes in Computational Intelligence and Decision Making*. 2020. P. 409–422.
- [5] Вальд А. Последовательный анализ. Москва: ФИЗ-МАТЛИТ, 1960.
- [6] Ширяев А.Н. Вероятностно-статистические методы в теории принятия решений. Москва: МЦНМО, 2014.
- [7] Ширяев А.Н. Стохастические задачи о разладке. Москва: МЦНМО, 2016.

## ПРО ПРОЦЕДУРУ РОЗВ'ЯЗУВАНOSTI ДЛЯ НЕЧІТКОЇ ПРОПОЗИЦІОНАЛЬНОЇ ЛОГІКИ

Наведено опис процедури перевірки виконуваності формул нечіткої пропозиційної логіки з можливістю апаратної реалізації.

**Ключові слова:** багатозначна пропозиційна логіка, нечітка пропозиційна логіка, розв'язуваність

### Вступ

У даній роботі розглядається процедура перевірки виконуваності формул нечіткої пропозиційної логіки на основі  $n$ -значних логік. Особливістю такої процедури є те, що вона може реалізуватися апаратно. Це досягається за допомогою зведення проблеми виконуваності формул нечіткої логіки до проблеми виконуваності  $n$ -значної логіки, а проблема виконуваності  $n$ -значної логіки розв'язується за допомогою кодування значеннями двозначної логіки.

### $n$ -значна пропозиційна логіка та її розв'язуваність

Алфавіт і синтаксис цієї логіки визначаються так само, як і для  $L_2$  – класичної пропозиційної логіки висловлювань. Різниця полягає в тому, що множина логічних значень  $L_n$ -логіки складається не із двох значень 0 і 1, як у логіці висловлювань, а із  $n$  значень –  $0, \frac{1}{n-1}, \frac{2}{n-1}, \dots, \frac{n-2}{n-1}, 1$ . Значення 0 і 1, як і в численні висловлень, представляють хибність і істину відповідно, а решта значень – ступінь істинності між 0 і 1. Зокрема, при  $n = 3$  маємо 3-значну логіку ( $L_3 = \{0, \frac{1}{2}, 1\}$ ), при  $n = 4$  маємо 4-значну логіку ( $L_4 = \{0, \frac{1}{3}, \frac{2}{3}, 1\}$ ) і т. д.

**Означення 1.** Відображення  $h : AP \rightarrow L_n$  називається інтерпретацією, де  $AP$  – алфавіт атомарних формул пропозиційної  $n$ -значної логіки, а  $L_n = \{0, \frac{1}{n-1}, \frac{2}{n-1}, \dots, \frac{n-2}{n-1}, 1\}$ .

Якщо інтерпретація задана, то можна обчислювати значення формул  $L_n$ -логіки. Обчислення значень формул  $n$ -значної логіки, як і формул  $L_2$ -логіки, виконуються шляхом застосування операцій, правила обчислення яких задаються за допомогою таблиць. Ці таблиці визначають загальний спосіб обчислення значень формул для довільного  $n$ . Отже, якщо  $A(A_1, A_2, \dots, A_k)$  означає, що формула  $A$  побудована із атомів  $A_1, A_2, \dots, A_k$  за допомогою логічних зв'язок, то довільна формула  $n$ -значної логіки  $A$ , яка залежить від атомарних формул  $A_1, A_2, \dots, A_k$  може мати  $n^k$  інтерпретацій. Звідси випливає, що *число всіх формул  $n$ -значної логіки, які залежать від атомарних формул  $A_1, A_2, \dots, A_k$ , дорівнює  $n^k$* .

Розглянемо модельні (семантичні) аспекти  $n$ -значної логіки на прикладі 3-значної логіки, оскільки ці аспекти характерні і для  $L_n$ -логік, де  $n > 3$ .

Нагадаємо, що формула  $P$  називається **наслідком** множини формул  $\Gamma$ , якщо за довільної інтерпретації всі формули множини  $\Gamma$  приймають логічні значення істини, то формула  $P$  теж істинна за цієї самої інтерпретації (тобто, не існує інтерпретації, за якої всі формули із множини  $\Gamma$  істинні, а формула  $P$  за цієї інтерпретації хибна).

**Аргументом** називається одна або більше формул, що складається з певних припущень і додаткової формули, яка є наслідком цих припущень. Говорять, що аргумент **обґрунтований (valid)**, якщо він є наслідком множини припущень або що наслідок впливає з обґрунтованих припущень.

Розглянемо таблиці істинності  $L_3$ -логіки Лукасевича [1]:

| $A$ | $B$ | $A \wedge B$ | $A \vee B$   | $A \rightarrow B$    | $\neg A$ |
|-----|-----|--------------|--------------|----------------------|----------|
| $x$ | $y$ | $\min(x, y)$ | $\max(x, y)$ | $\min(1, 1 - x + y)$ | $1 - x$  |
| 1   | 1   | 1            | 1            | 1                    | 0        |
| 1   | 1/2 | 1/2          | 1            | 1/2                  | 0        |
| 1   | 0   | 0            | 1            | 0                    | 0        |
| 1/2 | 1   | 1/2          | 1            | 1                    | 1/2      |
| 1/2 | 1/2 | 1/2          | 1/2          | 1                    | 1/2      |
| 1/2 | 0   | 0            | 1/2          | 1/2                  | 1/2      |
| 0   | 1   | 0            | 1            | 1                    | 1        |
| 0   | 1/2 | 0            | 1/2          | 1                    | 1        |
| 0   | 0   | 0            | 0            | 1                    | 1        |

Табл. 1: Таблиці істинності  $L_3$ -логіки Лукасевича.

В даній таблиці пояснення вимагає лише обчислення значення формули  $A \rightarrow B$ , оскільки решта формул обчислюється традиційно. Формула  $A \rightarrow B$  в  $L_3$ -логіці, як і в  $L_2$ -логіці, вважається істинною, якщо формула  $B$  не менш істинна, ніж формула  $A$ , тобто логічне значення формули

$B$  не менше логічного значення формули  $A$ . Це узгоджується з тим, що формули  $0 \rightarrow 0$ ,  $0 \rightarrow 1$ ,  $1 \rightarrow 1$  істинні, як і у випадку  $L_2$ -логіки. Якщо  $x$  – значення формули  $A$ , а  $y$  – значення формули  $B$ , то різниця  $x - y$  показує наскільки значення формули  $A \rightarrow B$  відхиляється від істини. Отже, формулі  $A \rightarrow B$  можна співставити значення  $1 - (x - y) = 1 - x + y$ . Комбінуючи обидва можливі випадки  $x \leq y$  і  $x > y$ , формулі  $A \rightarrow B$  співставляється значення  $\min(1, 1 - x + y)$ .

Використовуючи логічні зв'язки імплікації та заперечення будується аксіоматична система  $L_3$ -логіки Лукасевича, яку позначають  $L_3A$ ; ця система є несуперечною та повною. Наведемо схеми аксіом логіки  $L_3A$ :

**L<sub>3</sub>1.**  $A \rightarrow (B \rightarrow A)$ ;

**L<sub>3</sub>2.**  $(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C))$ ;

**L<sub>3</sub>3.**  $(\neg A \rightarrow \neg B) \rightarrow (B \rightarrow A)$ ;

**L<sub>3</sub>4.**  $((A \rightarrow \neg A) \rightarrow A) \rightarrow A$ .

Єдиним **правилом виведення** є *modus ponens* (MP): із  $A$  і  $A \rightarrow B$  впливає  $B$ .

Решта логічних зв'язок, введених Лукасевичем в  $L_3$ -логіку, визначаються таким чином:

$$A \vee B = (A \rightarrow B) \rightarrow B; \quad A \wedge B = \neg(\neg A \vee \neg B);$$

$$A \leftrightarrow B = (A \rightarrow B) \wedge (B \rightarrow A).$$

**Означення 2.** Формула  $A$  пропозиційної  $n$ -значної логіки називається  $L_n$ -тавтологією ( $L_n$ -суперечністю), якщо вона приймає значення 1 (0) за довільної інтерпретації  $h$ .

Формула  $A$  пропозиційної  $n$ -значної логіки називається квазі-тавтологією (квазі-суперечністю), якщо вона не приймає значення 0 (1) за довільної інтерпретації  $h$ .

Зокрема, формула  $A$  пропозиційної  $L_3$ -логіки називається квазі-наслідком множини формул  $\Gamma$ , якщо не існує інтерпретації, за якої формули із  $\Gamma$  приймають значення 1 або 1/2, а формула  $A$  приймає значення 0 за тієї самої інтерпретації.

Вищенаведене означення логічного слідування в  $L_n$ -логіці можна визначити й іншим способом [2].

**Означення 3.** Формула  $B$  називається логічним наслідком формул  $A_1, A_2, \dots, A_k$  в  $L_n$ -логіці (позначення  $A_1, A_2, \dots, A_k \vdash_{L_n} B$ ) тоді та тільки тоді, коли для довільної інтерпретації  $h$  має місце нерівність

$$h(A_1) + \dots + h(A_k) - h(B) \leq k - 1.$$

Відомо, що закон «виключення третього» не виконується в  $L_n$ -логіці, де  $n \geq 3$ . А це означає, що  $L_n$ -логіка суттєво відрізняється від  $L_2$ -логіки. Крім того, цей факт свідчить про те, що формули  $A \rightarrow B$  і  $\neg A \vee B$

не еквівалентні між собою в цій логіці.

Перевірку виконуваності формули  $L_n$ -логіки і того, що вона є  $L_n$ -тавтологією ( $n \geq 3$ ), можна реалізувати апаратно шляхом кодування значень  $L_n$ -логіки значеннями  $L_2$ -логіки із застосуванням функціональних елементів (автоматів Мілі). Розглянемо таку перевірку для  $L_3$ -логіки Лукасевича.

Для занурення  $L_3$ -логіки в  $L_2$ -логіку застосовується таке кодування значень  $L_3$ -логіки:  $1 \rightarrow 10, 1/2 \rightarrow 01$  і  $0 \rightarrow 00$ .

| $A = x$ | $B = y$ | $A \wedge B =$<br>$= \min(x, y)$ | $A \vee B =$<br>$= \max(x, y)$ | $A \rightarrow B =$<br>$= \min(1, 1 - x + y)$ | $\neg A =$<br>$= 1 - x$ |
|---------|---------|----------------------------------|--------------------------------|-----------------------------------------------|-------------------------|
| 10      | 10      | 10                               | 10                             | 10                                            | 00                      |
| 10      | 01      | 01                               | 10                             | 01                                            | 00                      |
| 10      | 00      | 00                               | 10                             | 00                                            | 00                      |
| 01      | 10      | 01                               | 10                             | 10                                            | 01                      |
| 01      | 01      | 01                               | 01                             | 10                                            | 01                      |
| 01      | 00      | 00                               | 01                             | 01                                            | 01                      |
| 00      | 10      | 00                               | 10                             | 10                                            | 10                      |
| 00      | 01      | 00                               | 01                             | 10                                            | 10                      |
| 00      | 00      | 00                               | 00                             | 10                                            | 10                      |

Табл. 2: Таблиця істинності логічних операцій при кодуванні.

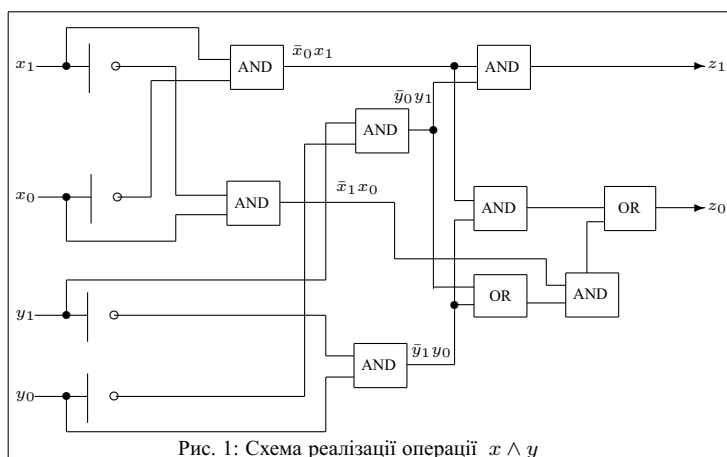
Оскільки при кодуванні значень  $L_3$ -логіки використовується два біти, то вихідні значення відповідних автоматів теж будуть двобітовими. Нехай  $x = x_1x_0, y = y_1y_0, z = z_1z_0$  – бітове зображення вхідних і вихідних значень змінних. За таблицею кодових значень знаходимо логічні вирази для вихідних значень. Це відбувається таким чином: знаходимо в таблиці рядки, в яких значення  $z_1 = 1$ ; за значеннями кодів аргументів в цих рядках будуємо логічний вираз для  $z_1$ .

Покажемо це кодування лише для операції кон'юнкції, тому що для решти операцій воно виконується аналогічно.

Для цієї операції в таблиці кодових значень є лише один рядок, який відповідає значенню  $z_1 = 1$ . За значеннями аргументів у цьому рядку будуємо вираз  $z_1 = x_1\bar{x}_0y_1\bar{y}_0$ , а вихідній змінній  $z_0 = 1$  відповідають рядки 2, 4 і 5. За значеннями кодів аргументів в цих рядках знаходимо після тотожних перетворень логічний вираз

$$z_0 = x_0\bar{x}_1(y_1\bar{y}_0 \vee \bar{y}_1y_0) \vee \bar{x}_0x_1\bar{y}_1y_0.$$

За цими логічними виразами будується логікова схема реалізації, яка обчислює значення операції кон'юнкції.

Рис. 1: Схема реалізації операції  $x \wedge y$ 

**Теорема 1.** *Приведена реалізація логічних операцій для тризначної логіки коректна, тобто правильно обчислює значення цих операцій.*

Доведення зводиться до розгляду правильності реалізації кожної з операцій. Наведемо доведення лише для операції кон'юнкції, а повне доведення можна знайти в роботі [3].

Операції  $x \wedge y$  згідно з таблицею кодування відповідає ДНФ такого вигляду:

$$z_0 = \bar{x}_0x_1y_0\bar{y}_1 \vee \bar{x}_1x_0y_1\bar{y}_0 \vee \bar{x}_1x_0y_0\bar{y}_1,$$

яка редукується до остаточного вигляду

$$z_0 = \bar{x}_1x_0(y_1\bar{y}_0 \vee \bar{y}_1y_0) \vee \bar{x}_0x_1y_0\bar{y}_1,$$

а для  $z_1 = \bar{x}_0x_1y_1\bar{y}_0$  ніяких редукцій виконувати не потрібно. Саме ці вирази і реалізуються схемою для  $x \wedge y$ . ■

## Нечітка пропозиційна логіка та її розв'язуваність

Аксиоматична система нечіткої пропозиційної логіки, яку будемо позначати  $F_LA$ , включає перші три схеми аксіом  $L_3$ -логіки Лукасевича та одну додаткову схему аксіоми  $F_LA4$ :

$$F_LA1. \quad P \rightarrow (Q \rightarrow P);$$

$$F_LA2. \quad (P \rightarrow Q) \rightarrow ((Q \rightarrow R) \rightarrow (P \rightarrow R));$$

$$F_LA3. \quad (\neg P \rightarrow \neg Q) \rightarrow (Q \rightarrow P);$$

$$F_LA4. \quad ((P \rightarrow Q) \rightarrow Q) \rightarrow ((Q \rightarrow P) \rightarrow P);$$

Єдиним правилом виведення є *modus ponens*:  $P, P \rightarrow Q \vdash Q$ .

Обчислення значень формул  $F_LA$ -логіки виконується за правилами 1-7  $L_3$ -логіки Лукасевича, які наведені нижче. Нехай  $h$  – інтерпретація, тобто  $h(P)$  означає логічне значення формули  $P$ , яке належить інтервалу дійсних чисел  $[0,1]$ , тобто  $h(P) \in [0,1]$ , тоді

1.  $h(\neg P) = 1 - h(P)$ ,
2.  $(P \wedge Q) = \min(h(P), h(Q))$ ,
3.  $(P \vee Q) = \max(h(P), h(Q))$ ,
4.  $h(P \rightarrow Q) = \min(1, 1 - h(P) + h(Q))$ ,
5.  $h(P \leftrightarrow Q) = \min(1, 1 - h(P) + h(Q), 1 - h(Q) + h(P))$ ,
6.  $h(P \& Q) = \max(0, h(P) + h(Q) - 1)$ ,
7.  $h(P \nabla Q) = \min(1, h(P) + h(Q))$ .

Останні дві логічні зв'язки називаються *чіткою кон'юнкцією* і *чіткою диз'юнкцією* відповідно.

Процедур, які розв'язують проблему перевірки виконуваності формул  $F_L A$ -логіки є декілька. Найзрозумілішою є процедура, запропонована у роботі [4], і яка полягає в наступному. Спочатку визначається  $n$ -значна пропозиційна логіка Лукасевича для кожного натурального числа  $n \geq 3$  зі значеннями з інтервалу  $\{0, \frac{1}{n-1}, \dots, \frac{n-2}{n-1}, 1\}$ , де значення формул цієї логіки обчислюються за вищенаведеними правилами 1 – 7.

Наступним кроком визначаються числа  $\#O(P)$ , значення яких дорівнюють кількості входжень атомарних формул у формулі  $P$ . Наприклад,  $\#O(B)$  і  $\#O(\neg B)$  дорівнює 1, а  $\#O(B \nabla C) = \#O(B \nabla B)$  дорівнює 2. Автори роботи [4] показали, що має місце така теорема.

**Теорема 2.** *Якщо формула  $P$  включає лише логічні зв'язки заперечення і чіткої диз'юнкції, то вона буде  $F_L$ -тавтологією тоді та тільки тоді, коли  $P \in L_n$ -тавтологією, де  $n = 2^{\#O(P)} + 1$ .*

Наприклад, формула  $B \nabla \neg B$  буде тавтологією  $F_L$ -логіки тоді та тільки тоді, коли вона є  $L_5$ -тавтологією, а формула  $\neg B \vee (B \nabla \neg C)$  буде  $F_L$ -тавтологією тоді та тільки тоді, коли вона є  $L_9$ -тавтологією. Оскільки формули всіх логік, які розглядалися, будуються за допомогою логічних зв'язок заперечення та імплікації, то для обґрунтування такої процедури потрібно показати, що імплікація виражається через заперечення та чітку диз'юнкцію. Але це майже очевидно, дійсно

$$P \rightarrow Q \leftrightarrow \neg P \nabla Q = \neg \neg P \rightarrow Q \leftrightarrow P \rightarrow Q,$$

на підставі закону подвійного заперечення. Це підтверджується також логічними значеннями формул:

$$\begin{aligned} h(P \rightarrow Q) &= \min(1, 1 - h(P) + h(Q)) = h(\neg P \nabla Q) = \\ &= \min(1, h(\neg P) + h(Q)) = \min(1, 1 - h(P) + h(Q)). \end{aligned}$$

Після цього множина тавтологій довільної пропозиційної  $L_n$ -логіки Лукасевича розв'язує проблему виконуваності  $F_L$ -логіки, оскільки існує механічна процедура перетворення формул  $F_L$ -логіки у формули відповідної  $L_n$ -логіки. А з попереднього підрозділу випливає можливість і апа-

ратної реалізації алгоритму перевірки тавтологічності формул  $F_L$ -логіки, шляхом зведення до  $L_n$ -логіки та зануренням її в  $L_2$ -логіку.

### Список використаних джерел

- [1] Лукасевич Я. Аристотелевская силлогистика с точки зрения современной формальной логики. Москва: Иностранная Литература, 1959.
- [2] Gabbay D.M. Elementary Logics: a Procedural Perspective. Prentice Hall (Europe), 1998.
- [3] Крытый С.Л., Опанасенко В.Н., Завьялов С. Б. Разбиение множества векторов с целыми координатами логическими аппаратными средствами. *Кибернетика и системный анализ*. 2019. Т. 55, № 3. С. 471–478.
- [4] Aguzzoli S, Ciabattoni A. Finiteness in Infinite-Valued Lukasiewicz Logic. *Journal of Logic, Language and Information*. 2000. Vol. 9, Issue 1. P. 5–29.

**Кузнєцов В.О., Кондратюк С.С., Крак Ю.В., Петрович В.М.,  
Самойлов О.М.**

<sup>1</sup> к. т. н., молодший науковий співробітник відділу інтелектуальних інформаційних технологій № 115 Інституту кібернетики імені В.М. Глушкова НАН України, Київ

<sup>2</sup> асистент кафедри теоретичної кібернетики Київського національного університету імені Тараса Шевченка

<sup>3</sup> ч-к. НАНУ, д. ф.-м. н., п.н.с. Інституту кібернетики імені В.М. Глушкова НАН України, завідувач кафедри теоретичної кібернетики Київського національного університету імені Тараса Шевченка

<sup>4</sup> к. т. н., старший науковий співробітник лабораторії комунікативних інформаційних технологій Інституту кібернетики імені В.М. Глушкова НАН України, Київ

<sup>5</sup> аспірант Інституту кібернетики імені В.М. Глушкова НАН України, асистент кафедри теоретичної кібернетики Київського національного університету імені Тараса Шевченка

*e-mail:* <sup>1</sup> kuznetsow.wlad@gmail.com, <sup>3</sup> krak@univ.kiev.ua

## **ДО РОЗРОБКИ ЛЮДИНО-МАШИННИХ ІНТЕРФЕЙСІВ НА ОСНОВІ ЖЕСТОВОЇ ІНФОРМАЦІЇ**

Розглядаються проблеми побудови систем людино-машинної взаємодії на основі жестової інформації, де найбільш важливе застосування пов'язане із жестовою мовою людей із вадами слуху. Окрім моделювання та вивчення жестової мови, розглянуті застосування жестової інформації в різних сферах, таких як автоматизовані робототехнічні системи із залученням оператора (для керування робототехнічною маніпуляційною системою), системи доповненої або віртуальної реальності з розпізнаванням рухів для інтерактивного управління і введення інформації тощо.

**Ключові слова:** *людино-комп'ютерний інтерфейс, відеозображення, розпізнавання жестів, маніпуляційна система.*

### **Вступ**

Для виявлення і постановки проблем розробки людино-комп'ютерних інтерфейсів на основі жестової інформації розглянемо такі системи на прикладі розпізнавання мови жестів методами комп'ютерного зору [1].

### **Жестова інформації у людино-машинної взаємодії**

Задача навчання жестовим мовам є складною для людини, нарівні навчання природним мовам; навчання для вільного володіння може тривати багато років, оскільки вміння покращуються повільно і поступово. Для пришвидшення навчання була запропонована технологія [2], яка поєднує ін-

формацію про значення жестів в Українській мові, так і про правильність показу жестових рухів при жестикулюванні. Отримані рухи відображаються в реальному часі та порівнюються з рухами носія жестової мови. Для отримання положення руки застосовуються алгоритми сегментації зображення. Для аналізу траєкторій під час показу жестів застосовано алгоритм динамічної трансформації часової шкали, що визначає подібність жестів між собою. Також була створена реалізація яка відстежує рухи рухів без застосування алгоритмів локалізації їх на зображенні. Відсутність такого компоненту, однак, не впливає суттєво на алгоритми розпізнавання рухів. Для захоплення рухів використовується датчик просторового зору Microsoft Kinect і відповідне програмне забезпечення для реконструкції взаємного розташування рук і кісток скелету. Залежно від реалізацій і типів даних досягалась точність розпізнавання до 80-95% при відтворенні рухів носіями жестової мови [3].

## Висновки

Самі по собі жести передають різний сенс: лінгвістичний, кінематичний і фізичний, але їм присутні спільні властивості – у всіх випадках рух передбачає перехід стану множини точок поверхні тілі з одного положення в інше, і є чітко усвідомлюваним рухом, що асоціюється із певною дією. Тому у людино-машинних інтерфейсах рухам присутні наступні властивості: рухи є природними і відтворюваними; існує скінченний набір таких рухів або елементів таких рухів; існує критерій правильності (закінченості) виконання руху; існує можливість чітко формалізувати як множину рухів; у відповідність досліджуваному руху можна поставити певний еталон; характеристики рухів мають однакові порядки величин.

## Список використаних джерел

- [1] Кривонос Ю.Г., Крак Ю.В., Бармак О.В. Системи жестової комунікації: моделювання інформаційних процесів. Київ: Наукова думка, 2014.
- [2] Сергієнко І.В., Крак Ю.В., Бармак О.В., Куляс А.І. Системи жестової комунікації: моделювання та розпізнавання дактильної інформації. Київ: Наукова думка, 2019.
- [3] Ю.В. Крак, Ю.Г. Кривонос, А.В. Бармак, А.С. Тернов Подход к определению эффективных признаков и синтезу оптимального полосно-разделяющего классификатора для элементов дактильно-жестовой речи. *Кибернетика и системный анализ*. 2016. Т. 52, № 2. С. 3–10.

**Кузнєцов В.О., Крак Ю.В., Петрович В.М., Самойлов О.М.,  
Куляс А.І.**

<sup>1</sup> к. т. н., молодший науковий співробітник відділу інтелектуальних інформаційних технологій № 115 Інституту кібернетики імені В.М. Глушкова НАН України, Київ

<sup>2</sup> ч-к. НАНУ, д. ф.-м. н., п.н.с. Інституту кібернетики імені В.М. Глушкова НАН України, завідувач кафедри теоретичної кібернетики Київського національного університету імені Тараса Шевченка

<sup>3</sup> к. т. н., старший науковий співробітник лабораторії комунікативних інформаційних технологій Інституту кібернетики імені В.М. Глушкова НАН України, Київ

<sup>4</sup> аспірант Інституту кібернетики імені В.М. Глушкова НАН України, асистент кафедри теоретичної кібернетики Київського національного університету імені Тараса Шевченка

<sup>5</sup> к. т. н., учений секретар Інституту кібернетики імені В.М. Глушкова НАН України

*e-mail:* <sup>1</sup> kuznetsow.wlad@gmail.com, <sup>2</sup> krak@univ.kiev.ua

## **ОЦІНКА НАДІЙНОСТІ ПОДАННЯ ЗНАНЬ ІЗ ВИКОРИСТАННЯМ ОПТИМАЛЬНОГО ШЛЯХУ МІЖ ВЛАСТИВОСТЯМИ**

У доповіді обговорюється питання оцінки надійності штучного інтелекту, а саме моделі знань, що описується за допомогою структурного подання «entity-relationship» і онтологій. Розглянуто підхід до побудови надійних зв'язків між властивостями із різних доменів знань, що ґрунтується на пошуку оптимального шляху між різними доменами знань на підставі ймовірнісних оцінок, таких як якість та надійність подання. Для подання зв'язків («link») між властивостями із різних доменів використовуються ймовірнісні зв'язки, що дозволяє видобути нові зв'язки та оцінити ступінь надійності існуючих.

**Ключові слова:** *модель знань, сутність-відношення, концепти, ймовірнісна оцінка, онтологія.*

### **Вступ**

При побудові надійних систем штучного інтелекту для описання структурних моделей знань [1], важливою є оцінка якості побудованих зв'язків «entity-relationship», що містяться в середині домена, а також між окремими доменами знань. Набільший інтерес проявляють зв'язки між різними доменами знань, оскільки дозволяє видобувати додаткову інформацію із іншого домена знань при відсутності інформації в окремих доменах знань. Такі зв'язки можуть бути як явно виражені, так і з певною ймовірністю, де велике значення ймовірності позначає сильний зв'язок між різними властивостями, а відповідно мале – його відсутність або незначний зв'язок. Для видобутку таких зв'язків можливе використа-

ння різних систем опису моделей знань та певних алгоритмів, що автоматизують видобуток зв'язків на підставі моделей знань.

## **Алгоритм подання знань із використанням оптимального шляху між властивостями**

Для видобутку таких зв'язків можливе використання різних систем опису моделей знань та певних алгоритмів, що автоматизують видобуток зв'язків на підставі моделей знань. За основу взято алгоритм оптимального транспорту [2], що в контексті моделі знань оперує доменами 1 та 2 і будує ймовірнісні зв'язки між кожним елементом множини 1 та 2; отримані величини ймовірностей показують величину надійності зв'язку між цими множинами. При побудові зв'язків важливо мати модель, що описує концепти різних рівнів, такими як «робастність», «функціональна надійність», «керованість» тощо. На кожному з рівнів структурні зв'язки матимуть різну вірогідність, оскільки в кожному з концептів зв'язки між властивостями мають більшу важливість відносно окремо взятого концепту, при цьому одна властивість може мати зв'язки на різних рівнях з високою ймовірністю, так і мати тільки один зв'язок. Для побудови таких зв'язків використовується відстань Сінкорна [2], що описує зв'язки між різними доменами знань для окремо взятих властивостей.

## **Висновки**

Отримання оптимального шляху між доменами знань дозволяє оцінити надійність подання інших моделей знань, за умови використання спільних концептів, в яких оцінюється зв'язок між різними властивостями в моделях знань, що дозволяє реалізувати однаковий порядок властивостей в різних моделях знань. Експерименти на різних моделях показали ефективність такого підходу для оцінки надійності подання знань.

## **Список використаних джерел**

- [1] Manziuk E.A., Krak Iu. V., Barmak O.V., Mazurets O.V., Kuznetsov V.O. Alignment Method of Conceptual Categories of Ontology and Formalized Domain. *CEUR Workshop Proceedings*. 2021. Vol. 3003. P. 11–22.
- [2] Cuturi M. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *26th International Conference on Neural Information Processing Systems NIPS'13*. 2013. Vol. 2. P. 2292–2300.

<sup>1</sup> к. т. н., доцент кафедри комп'ютерних наук Хмельницького національного університету

<sup>2</sup> ч-к. НАНУ, д. ф.-м. н., п.н.с. Інституту кібернетики імені В.М. Глушкова НАН України, завідувач кафедри теоретичної кібернетики Київського національного університету імені Тараса Шевченка

<sup>3</sup> д. т. н., завідувач кафедри комп'ютерних наук Хмельницького національного університету

e-mail: <sup>1</sup> eduard.em.km@gmail.com, <sup>2</sup> krak@univ.kiev.ua, <sup>3</sup> alexander.barmak@gmail.com

## ПРО РОЗРОБКУ МЕТОДІВ МАШИННОГО НАВЧАННЯ НА ОСНОВІ ДОВІРИ ШТУЧНОМУ ІНТЕЛЕКТУ

В даній доповіді представлені дослідження з розробки методів машинного навчання, орієнтованих на практичне використання в системах штучного інтелекту та в системах прийняття рішень, що характеризуються високою відповідальністю та довірою до прийнятих рішень. Значне використання засобів штучного інтелекту в широкому спектрі практичних застосувань та підвищення його важливості та впливовості в житті, як окремої людини так людства в цілому, призвело до актуалізації цілого ряду питань щодо створення нової дійсності, в якій штучний інтелект відіграє значну а в окремих випадках вирішальну роль. Приведені методи машинного навчання для забезпечення довіри штучному інтелекту на прикладному рівні.

**Ключові слова:** машинне навчання, довіра штучному інтелекту, ансамблі моделей, кластеризація, прийняття рішень.

### Вступ

В зв'язку з наявною актуальністю щодо формалізації вимог стосовно засобів обчислювального інтелекту в 2019 році Група експертів високого рівня зі штучного інтелекту (AI HLEG), створена Європейською комісією, розробила та опублікувала Етичні рекомендації стосовно Довіри штучному інтелекту [1]. Обчислювальна система, яка розробляється, впроваджується, або використовується, повинна відповідати ряду високорівневих концептів, як зазначено в етичних рекомендаціях для забезпечення довіри штучному інтелекту (ШІ). Довіра є визначальною умовою використання обчислювального інтелекту при отриманні відповідальних рішень на основі обробки масивів даних. Довіра є складовою частиною етичного штучного інтелекту [2].

## **Довіра штучному інтелекту як основа прийняття ефективних рішень**

Відзначимо, що поряд з концептами етики важливою частиною є технічна складова обчислювальних систем, заснованих на методах ШІ, для прийняття ефективних рішень. Досліджено наступні рівні: фізичний – практичне втілення з системами отримання інформації з давачів і довіра до цієї інформації; інфраструктурний – побудова систем взаємодії, передачі та обробки даних; прикладний – розробка програмних систем та реалізації методів обробки даних на основі довіри до отриманих результатів. Це є визначальною складовою ефективності застосування систем прийняття рішень в напрямі довіри користувачів і осіб, які приймають рішення до отриманих результатів, що забезпечується методами машинного навчання, які дозволяють реалізувати прикладний рівень.

## **Результати і експериментальні випробування**

Розроблено методи машинного навчання, які дозволяють забезпечити довіру до прийнятих рішень на прикладному рівні. Досліджувались базові принципи довіри, які стали основою розроблених методів. Поряд з тим, забезпечувалось дотримання концептів довіри на рівні методів і прийнятих рішень та даних, вдалось забезпечити якість прийняття рішень на високому рівні. В основі методів машинного навчання покладений розроблений підхід з використанням ансамблів моделей та класифікації на основі ієрархічної кластеризації [3]. Проведені експерименти показали ефективність пропонованого підходу з точки зору показників якості прийнятих рішень та забезпечення вимог довіри ШІ.

## **Список використаних джерел**

- [1] High-level expert group on artificial intelligence set up by the European Commission. Ethics guidelines for trustworthy AI. 2019. Url: <https://op.europa.eu/en/publication-detail/-/publication/d3988569-0434-11ea-8c1f-01aa75ed71a1>.
- [2] Manziuk E.A., Barmak O.V., Krak I.V., Mazurets O.V., Skrypnyk T.K. Formal Model of Trustworthy Artificial Intelligence Based on Standardization. *CEUR Workshop Proceedings*. 2021. Vol. 2853. P. 190–197.
- [3] Manziuk E.A., Wojcik W., Barmak O.V., Krak I.V., et al. Approach to creating an ensemble on a hierarchy of clusters using model decisions correlation. *Przegląd Elektrotechniczny*. 2020. Vol. 96, No 9. P. 108–113.

<sup>1</sup> д. ф.-м. н., професор кафедри математичної інформатики Київського національного університету імені Тараса Шевченка

<sup>3</sup> к. ф.-м. н., науковий співробітник відділу інтелектуалізації інформаційних технологій Міжнародного науково-навчального центру інформаційних технологій та систем НАНУ та МОН України

e-mail: <sup>1</sup> rozenkrans17@gmail.com, <sup>2</sup> enasirov@gmail.com

## БЛОЧНО-ДІАГОНАЛЬНИЙ ПІДХІД ДО НЕВІД'ЄМНОЇ ФАКТОРИЗАЦІЇ РОЗРІДЖЕНИХ ЛІНГВІСТИЧНИХ МАТРИЦЬ НАДВЕЛИКОЇ РОЗМІРНОСТІ

В роботі описано алгоритми невід'ємної факторизації розріджених матриць та тензорів, популярну технологію в штучному інтелекті взагалі, та в комп'ютерній лінгвістиці зокрема. Наведено існуючі підходи до факторизації матриць та тензорів. Запропоновано використання латентного розподілу Діріхле для приведення матриць та тензорів до блочно-діагональної форми для паралелізації та пришвидшення невід'ємної факторизації лінгвістичних матриць та тензорів надвеликої розмірності. Використання пам'яті та мережових операцій було проаналізовано та оптимізовано.

**Ключові слова:** *штучний інтелект, комп'ютерна лінгвістика, паралельні обчислення.*

### Вступ

Сьогодні невід'ємна факторизація матриць та тензорів є дуже популярною технологією в штучному інтелекті взагалі, та в комп'ютерній лінгвістиці зокрема. Використовуючи невід'ємну факторизацію в рамках парадигми латентно-семантичного аналізу, комп'ютерні лінгвісти застосовують даний підхід для розв'язання таких прикладних задач, як класифікація, кластеризація текстів та термінів [1, 2], побудова мір семантичної близькості[3], автоматичне виділення лінгвістичних структур та відношень (Selectional Preferences [4] та Verb Sub-Categorization Frames [5]) і багато інших. Дана робота описує побудову моделі паралелізації обчислення невід'ємної факторизації розріджених матриць надвеликої розмірності, що представляє особливий інтерес та актуальність з точки зору її застосування у великих NLP системах загального тематичного призначення, не обмежених використанням лише для вузьких предметних областей. Задача невід'ємної факторизації розріджених матриць та тензорів надвеликої розмірності постала в процесі розробки системи визначення

міри семантичної близькості-зв'язності за технологією Латентного Семантичного Аналізу[6]. Для побудови системи був оброблений великий текстовий корпус статей англomовної Вікіпедії.

## Алгоритм невід'ємної матричної факторизації

Алгоритм невід'ємної матричної факторизації виконує декомпозицію невід'ємної матриці  $V$  розміру  $[n; m]$  у добуток невід'ємних матриць  $W$  та  $H$  розмірами  $[n; k]$  та  $[k; m]$  відповідно, таким чином, щоб  $V \approx WH$ . Мета алгоритму – мінімізувати функцію оцінки, яка кількісно визначає якість наближення. В якості такої функції може бути використана функція виміру відстані між двома невід'ємними матрицями. Однією з таких мір є квадрат Евклідової метрики між матрицями  $V$  та  $WH$ :

$$\mu(V, W, H) = \sum_{i=1}^n \sum_{j=1}^m (V_{i,j} - (WH)_{i,j})^2.$$

В роботі [7], автори запропонували простий ітеративний алгоритм, що дозволяє отримати наближені матриці  $W$  and  $H$ . Він складається з двох послідовних правил оновлення матриць  $W$  та  $H$  (1) та (2) відповідно.

$$(W_t)_{ij} = (W_{t-1})_{ij} \frac{(V H_t^T)_{ij}}{(W_{t-1} H_t H_t^T)_{ij}} \quad (1)$$

$$(H_t)_{ij} = (H_{t-1})_{ij} \frac{(W_{t-1}^T V)_{ij}}{(W_{t-1}^T W_{t-1} H_{t-1})_{ij}} \quad (2)$$

В (1) та (2),  $H_t$  та  $W_t$  – це матриці отримані на ітерації  $t$ . Матриці  $H_0$  та  $W_0$  ініціалізуються випадковими значеннями з  $[0; 1)$ . Алгоритм продовжується до тих пір, поки не буде досягнуто стаціонарної точки або не буде виконано задану максимальну кількість ітерацій.

## Блочно-діагональний підхід до факторизації матриць

Суть методу полягає у переході від необхідності факторизувати надвеликий розріджений лінгвістичний тензор або матрицю до невід'ємної факторизації набору лінгвістичних тензорів та матриць значно зменшеного розміру.

Розрідженість матриць та тензорів дозволяє привести їх до блочно-діагональної форми за допомогою перестановок шарів в тензорах або рядків та стовпчиків в матриці, змінюючи базис факторизації. У випадку, коли слово може належати декільком блокам, слово може бути розділене на дві копії, що відповідатимуть одному слову.

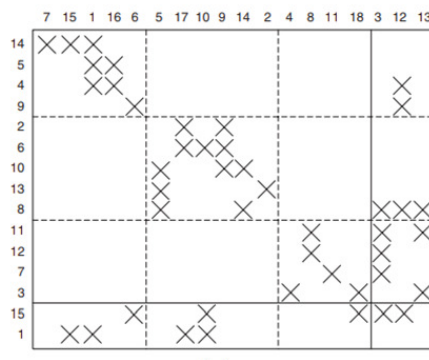


Рис. 1: Графічне представлення матриці блочно-діагонального виду.

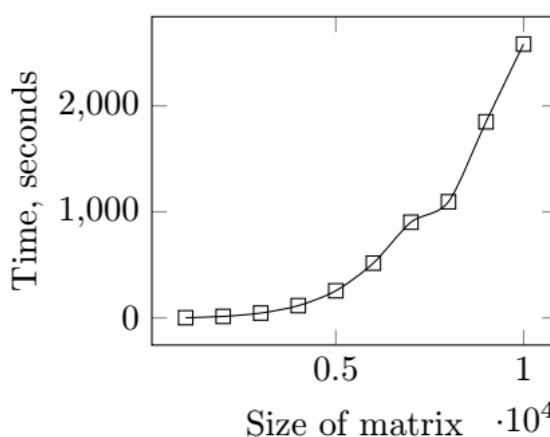


Рис. 2: Необхідний час для факторизації матриць різних розмірів.

Переваги такого підходу:

- Не потрібно мережевих операцій на кожній ітерації. Необхідно лише початкове розподілення блоків матриці або тензора між вузлами.
- Швидша збіжність. На Рис. 2 показано час необхідний для факторизації квадратних тензорів різних розмірів відповідно, в один потік без використання GPU. На Рис. 3 показано необхідну кількість ітерацій для невід'ємної факторизації квадратних матриць різних розмірів.
- Зменшення  $k$ , а отже, і зменшення розмірів результатів та об'ємів необхідних даних для зберігання в пам'яті під час ітерацій.
- Швидший підрахунок функції оцінки  $\mu$ . Швидший підрахунок добутку  $W \times H$ , оскільки зменшується  $k$ , а отже, і розміри матриць  $W$  та  $H$ . Кількість елементів для підрахунку функції оцінки менша в  $N$  разів, кількість необхідних перевірок менша, так як зменшується загальна кількість ітерацій.
- При цьому також гарантовано ефективно підтримується масштабованість моделі. Тобто для її розширення та збагачення новими дани-

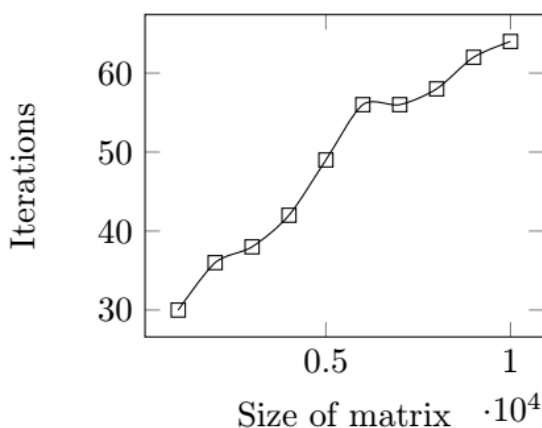


Рис. 3: Необхідна кількість ітерацій для факторизації матриць різних розмірів.

ми з додаткових текстових корпусів немає необхідності виконувати невід'ємну факторизацію всього надвеликої матриці з самого початку. Обчислення виконуються лише на окремих блоках лінгвістичної матриці з подальшою модифікацією тільки відповідних фрагментів векторів моделі.

- Швидке відновлення значень після факторизації. Всі ненульові значення розподілені тільки по діагональним блокам, тому всі значення що знаходяться поза діагональними блоками не можуть бути відмінні від 0. Для отримання значення достатньо відновлення значення лише в блоках входження слів. У випадку зведеної до блочно-діагональної матриці, в результаті факторизації окремих блоків для кожного слова буде отримано один або декілька рядків та ідентифікатор блоку в якому це слово знаходиться. При відновленні, якщо слово та стаття знаходяться в різних блоках – результатом буде 0, інакше результатом буде добуток відповідних рядка та стовпчика.

Недоліком запропонованого методу є те, що створення копій слів при групуванні необхідних ненульових комірок матриці в блоки відбувається випадковим чином та може призвести до створення багатьох копій одного слова.

## **Приведення розріджених матриць до блочно-діагонального виду**

Розрідженість матриць та тензорів дозволяє спробувати привести їх до блочно-діагональної форми концентруючи ненульові значення в діагональних блоках за допомогою перестановок шарів в тензорах або рядків та стовпчиків в матрицях.

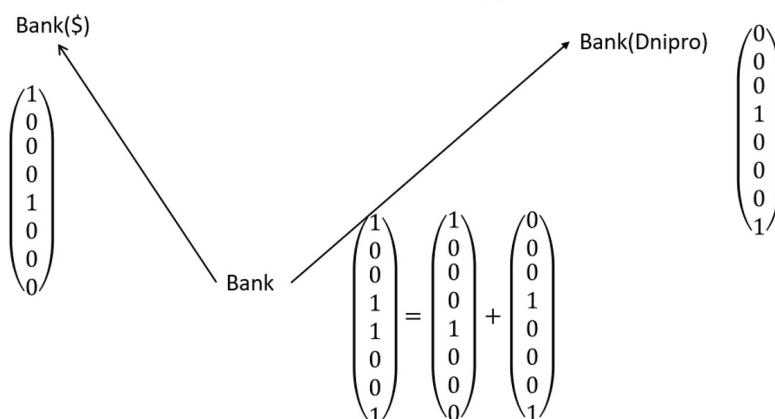


Рис. 4: Розщеплення вектору валентності слова Bank.

Проте не кожна розріджена матриця може бути зведена до блочно-діагональної форми перестановками рядків та стовпчиків, і може бути отримана блочно-діагональна форма з обрамленням (Рис. 1). Простим прикладом може бути слово «Слово», яке може зустрічатись з будь-яким іншим словом в реченнях, наприклад, в яких описуються кількості голосних та приголосних літер в слові: «Слово корпорація містить 5 голосних та 5 приголосних літер.» А отже вектор цього слова «Слово» може бути цілком щільний.

Для вирішення такої проблеми для лінгвістичних матриць та тензорів може бути використана непритаманна для звичайних матриць особливість пов'язана з особливостями природньої мови.

Слово може мати декілька значень та використовуватись у декількох темах. Тому можна припустити розщеплення векторів семантико-синтаксичної валентності слів на складові вектори його окремих значень.

Так, наприклад, слово англійської мови Bank може бути використано в наступних значеннях:

- Bank – банк, фінансова установа.
- Bank – берег річки.

Тоді може бути створена копія слова для іншого значення, а вектор семантико-синтаксичної валентності розщеплений на два вектори, що відповідають різним значенням (Рис. 4). Таким чином, для кожного значення буде отримано власний вектор:

$$V_{bank} = V_{bank\_\$} + V_{bank\_river} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (3)$$

Отже, у випадку, коли слово може належати декільком блокам, вектор/слово може бути розділене на дві копії, що відповідатимуть одному слову.

У випадку зведення матриці до блочно-діагонального виду з тематичними діагональними блоків, можна зробити припущення, що слово може використовуватись не тільки в різних значеннях, але й з одним значенням хоча і в текстах різних тематик. Отже будь-який вектор слова може бути розкладений на суму векторів, що вказують на семантико-синтаксичну валентність слів у відповідній тематиці.

Таким чином, щоб відновити початкове значення з вихідної матриці, необхідно знайти суму значень для всіх можливих значень слів. А отже, буде відновлено частоту вживання слова у всіх його значеннях та у всіх тематиках.

При побудові діагональних блоків розщепленням слів, аналізуючи їх входження до різних блоків додатково можна робити припущення про наявність у слова декількох значень, широту вживаності в різних тематиках текстів. Також з'являється можливість досліджувати частоту вживаності комбінацій слів в різних тематиках.

Для перевірки швидкодії була обрана матриця Підмет-Присудок розмірами [8537;7234] яка була зведена до блочно-діагонального вигляду. Таким чином було отримано 3 діагональних блоки розмірами [2304;1847], [2523;2235], [3729;3212]. Час витрачений на факторизацію початкової матриці - 1095 секунд. Для факторизації діагональних блоків знадобилось 21, 35 та 56 секунд відповідно.

Таким чином, з наведених даних видно, що навіть на не великих розмірах матриць запропонований підхід дозволяє зменшити час факторизації на 89% зі збереженням точності. При цьому факторизація окремих діагональних блоків може бути виконана паралельно на різних обчислювальних вузлах.

## Висновки

Запропоновано блочно-діагональний підхід до факторизації невід'ємних розріджених лінгвістичних матриць, які можуть бути зведені до блочно-діагональної форми. Цей підхід може прискорити факторизацію, потребує менше мережевих операцій та пам'яті для ітерацій і зберігання результатів. Суть методу полягає у переході від необхідності факторизувати надвелику розріджену лінгвістичну матрицю до невід'ємної факторизації набору лінгвістичних матриць значно зменшеного розміру. Запропоновано розщеплення векторів семантико-синтаксичної валентності слів на складові вектори їх окремих значень для зведення лінгвістичних матриць до блочно-діагональної форми. Запропонований підхід продемонстрував значне зменшення часу факторизації та необхідних ресурсів.

## Список використаних джерел

- [1] Xu W., Liu X., Gong Y. Document Clustering Based on Non-negative Matrix Factorization. In *SIGIR'2003 : proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Toronto, 2003). 2003. P. 267–273.
- [2] Shahnaz F., Berry M.W., Pauca V.P., Plemmons R.J. Document clustering using nonnegative matrix factorization. *Information Processing and Management*. 2006. Vol. 42, Issue 2. P. 373–386.
- [3] Anisimov A., Marchenko O., Nikonenko A., Porkhun E., Taranukha V. Ukrainian WordNet: Creation and Filling. In: *Flexible Query Answering Systems FQAS-2013 : 10th International Conference* (Granada, 2013). 2013. P. 267–273.
- [4] Van de Cruys T. A Non-negative Tensor Factorization Model for Selectional Preference Induction *Journal of Natural Language Engineering*. 2010. Vol. 16, Issue 4. P. 417–437.
- [5] Van de Cruys T. Multi-way Tensor Factorization for Unsupervised Lexical Acquisition. In *COLING-2012 : 24th International Conference on Computational Linguistics* (Mumbai, 2012). 2012. P. 2703–2720.
- [6] Deerwester S., Dumais S.T., Furnas G.W., Landauer T.K., Harshman R. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*. 1990. Vol. 41, Issue 6. P. 391–407.
- [7] Lee D.D., Seung H.S. Algorithms for non-negative matrix factorization. In *NIPS'00 : Proceedings of the 14th International Conference on Neural Information Processing Systems* (Denver, 2000). 2001. P. 556–562.

**Нікітенко О.М., Дейнеко Ж.В., Грищенко Т.Б.**

<sup>1</sup> к. т. н., доцент кафедри інформаційно-вимірних технологій Харківського національного університету радіоелектроніки

<sup>2</sup> к. т. н., завідувач кафедри медіасистеми та технологій Харківського національного університету радіоелектроніки

<sup>3</sup> директор наукової бібліотеки Харківського національного університету радіоелектроніки

*e-mail:* <sup>1</sup> nikonxipe@gmail.com, <sup>2</sup> zhanna.deineko@nure.ua, <sup>3</sup> tamara.gryshchenko@nure.ua

## **ДОСЛІДЖЕННЯ ПЕРЕВАГ ЗАСТОСУВАННЯ ГРАФІКІВ У СИСТЕМІ $\text{\LaTeX}$ ПРИ ОФОРМЛЕННІ НАУКОВИХ ПРАЦЬ**

В статті наведені якісні переваги використання системи  $\text{\LaTeX}$  як засобу створення математичних і технічних текстів для публікації в наукових виданнях. В роботі досліджено актуальність застосування мови розмітки даних та пакетів макросів  $\text{\TeX}$  для високоякісного оформлення графіків математичних функцій.

**Ключові слова:** *TEX, LATEX, верстка, TikZ.*

### **Вступ**

Сьогодні одним з найпоширеніших засобів діяльності людини є інформаційно-комунікаційні технології, які стають могутнім каталізатором і визначальним джерелом об'єктивного розвитку інформаційного суспільства [1]. Інформатизація суспільства – це інтелектуальний і матеріальний процес, спрямований на створення глобальної інфраструктури сучасних засобів опрацювання, зберігання, розповсюдження і застосування інформації, яка стає стратегічним ресурсом суспільства [2]. Формування в тих, хто навчається, інформаційної компетентності стало одним з основних пріоритетів сучасної освіти, який має загально-навчальний і загально-інтелектуальний характер.

В процесі навчання студенти неминуче стикаються з проблемою оформлення результатів своїх досліджень: лабораторні та практичні роботи, реферати, кваліфікаційні роботи, науково-дослідницька діяльність, творчі проекти тощо [3]. За останні роки у багатьох вишах для оформлення звітів до лабораторних робіт, курсових робіт та проектів, а також випускних кваліфікаційних робіт студентів заохочують до використання системи оформлення документів  $\text{\LaTeX}$ . Зручність полягає у тому, що спираючись на шаблон, студенти якісно оформлюють звіти до своїх ро-

біт та пояснювальні записки, оскільки вписати свої персональні дані у розроблену форму зовсім не складно, а викладачі отримують оформлені за стандартами відповідні роботи. Ці практичні навички дозволять молодим науковцям з легкістю готувати весь спектр наукових публікацій (препринти, статті в міжнародних реферованих журналах, усні доповіді на семінарах та конференціях, постери та презентації для конференцій) з найвищою якістю, високою сумісністю створених робіт і в ногу з часом, а також привчатиме їх до структурування й методичності власної наукової роботи.

Одним з найскладніших елементів оформлення робіт є побудова в них графіків функцій різних видів. Ця робота присвячена опису побудови різноманітних графіків функцій за допомогою пакетів видавничої системи L<sup>A</sup>T<sub>E</sub>X.

## Характеристики пакетів для побудови графіків функцій

На сьогоднішній день, поширеною альтернативою, що допомагає дотримуватися стандартів журналів, індексованих Scopus і Web of Science є стандарт видавничої системи L<sup>A</sup>T<sub>E</sub>X. Цю систему використовують для створення наукових документів, написання книжок, а також багатьох інших форм публікацій. Вона дозволяє не тільки створювати красиво оформлені документи, але й надає користувачам можливість дуже швидко реалізовувати такі складні елементи друкованого набору, як математичні вирази, таблиці, посилання та бібліографії, отримуючи узгоджену розмітку по всіх розділах вихідного документу [4-6].

Переваги L<sup>A</sup>T<sub>E</sub>X:

- високоякісне верстання – текст виглядає «як в книжці»;
- зручна робота зі складними математичними формулами;
- відмінна крос-платформна сумісність;
- автоматизація багатьох рутинних процесів:
  - нумерація формул, рисунків, таблиць, розділів документу;
  - організація перехресних посилань;
  - створення колонтитулів;
  - оформлення стилів заголовків тощо;
- широке коло користувачів та розробників;
- пакети розширення «на всі випадки життя».

Недоліки L<sup>A</sup>T<sub>E</sub>X:

1. Для друку L<sup>A</sup>T<sub>E</sub>X-документів на папері потрібен принтер високої якості.

2.  $\text{\LaTeX}$  не є системою типу WYSIWYG (що друкую – те й бачу): створення документу та перегляд того, як виглядає документ при друці є різними операціями.

Видавнича система  $\text{\LaTeX}$  є стандартом у науковому світі й орієнтована на читача, котрому в своїй діяльності необхідно готувати видання високої якості, тексти яких містять формули, графіки, кольорові діаграми, ілюстрації, а також на фахівців з комп'ютерної графіки. Найкращі математичні, фізичні та економічні журнали видаються в  $\text{\LaTeX}$  й рекомендують авторам використовувати цю систему для підготовки рукописів. Існує велика кількість різноманітних систем для побудови графіків функцій та візуалізації даних, починаючи від електронних таблиць Excel і закінчуючи системами комп'ютерної математики (СКМ) Mathematica і Maple. Доречі, СКМ Maple має вбудовані команди експорту математичних формул та графіків функцій в  $\text{\LaTeX}$ .

Видавнича система  $\text{\LaTeX}$  має зручні пакети для графічних побудов.  $\text{\TeX}$  дозволяє будувати зображення безпосередньо в тілі документу. Для цього в місці розташування ілюстрації описують алгоритм побудови зображення, яке будуватиметься на етапі компіляції. Таким чином  $\text{\LaTeX}$  перебирає на себе функції графічного редактора.

Можливості, які має вбудоване оточення `picture` надто обмежені; пакет `PStricks` орієнтовано на мову PostScript (не працює з `pdflatex`); система `MetaPost` є найпотужнішою зі всіх у цьому напрямку, але вона функціонує з використанням окремого інтерпретатора. Одним з пакетів, що активно розвивається і є зручним, є пакет `PGF/TikZ`, що написано професором Тилем Тантау (Till Tantau), співробітником інституту теоретичної інформатики Любекського університету, у 2007 році.

Ця система має три інтерфейсних рівні:

- Системний рівень (`PGFSys`) складають команди, які безпосередньо здійснюють вимальовування зображення. Команди цього рівня можуть використовуватися в модулях, але не в вихідному документі, тобто користувач не взаємодіє з цим рівнем безпосередньо.
- Базовий рівень (`PGF` – назва `PGF` є аббревіатурою перших літер фрази `portable graphics format` (графічний формат, який переносять)) – це надбудова над системним рівнем, яка реалізує всі можливості модуля `PGF/TikZ`. Команди цього рівня мають  $\text{\TeX}$  синтаксис, та є доступними користувачеві. Область застосування інтерфейсу `PGF` – створення нових макросів і процедур, а також взаємодія з іншими модулями. Модуль `PGF` підключають командою `\usepackage{pgf}`.
- Рівень користувача (`TikZ` – це рекурсивний акронім, що розшифро-

вують як TikZ ist kein Zeichenprogramm (нім. TikZ не є програмою для малювання). TikZ це лише оболонка (малює PGF) – це оболонка, що забезпечує зручний інтерфейс для кінцевого користувача. TikZ підключають за допомогою команди `\usepackage{tikz}`. Під час підключення також стають доступними всі команди PGF [6].

Пакет TikZ надає кілька різних інтерфейсів для побудови графіків.

## Приклади застосування

Для побудови графіків функцій в  $\text{\LaTeX}$  найчастіше використовують або примітив `plot`, або команду `addplot` з оточення `axis`. Найзручнішою є команда `addplot`, яка містить всі опції примітиву `plot` і ще низку додаткових опцій. Використовуючи код [8] можна побудувати графіки різноманітних функцій як двовимірних

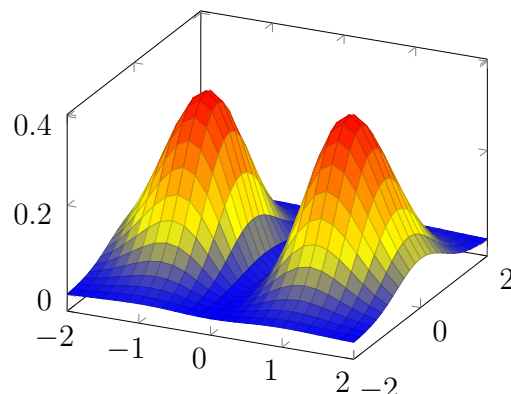
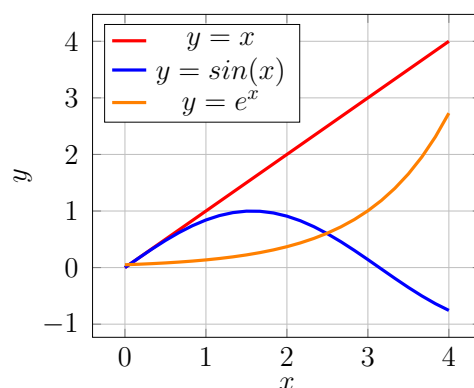
```
\begin{tikzpicture}[domain=0:4,
scale=0.75]
\begin{axis}
[xlabel={\$x\$}, ylabel={\$y\$},
grid=major, legend pos={north
west}]
\addplot+[draw=red,mark=none,
ultra thick]{x};
\addlegendentry{\$y=x\$}
\addplot+[draw=blue,mark=none,
ultra thick]{sin(deg(x))};
\addlegendentry{\$y=sin(x)\$}
\addplot+[draw=orange,mark=none,
ultra thick]{0.05*exp(x)};
\addlegendentry{\$y=e^x\$}
\end{axis}
\end{tikzpicture}
```

так і тривимірних поверхонь [9]

```
\begin{tikzpicture}
\begin{axis}
\addplot3[surf][domain=-2:2]
{x^2*exp(-x^2-y^2)};
\end{axis}
\end{tikzpicture}
```

та графіків

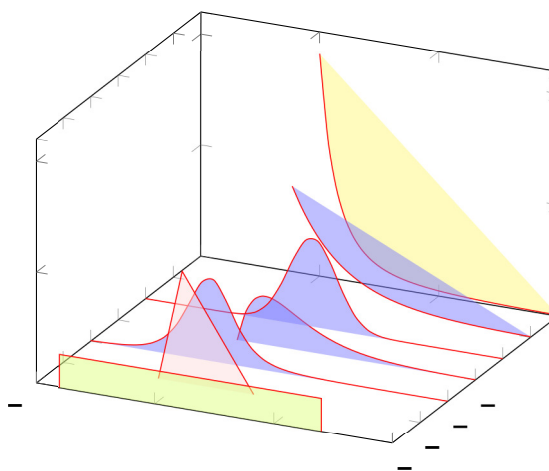
```
\begin{tikzpicture}
\begin{axis}[samples=30,domain=0:10, xlabel={\$x\$},
zmin=0,zlabel={\$z\$}, samples y=0, ytick={-4,-3,-2,
```



```

-1,0,1,2},ylabel={\$y\$}, area plot/.style={
fill opacity=0.5, draw=red, fill=blue!50,mark=none,smooth}}
\addplot3 [area plot,fill=yellow!50] (x,2,{1/(1+x/2)^3});
\addplot3 [area plot] (x,-1,{x*exp(-x/2)/2^(x/2+1)});
\addplot3 [area plot] (x,1,{exp(-x/2)/2});
\addplot3 [area plot,domain=-5:10] (x,0,{normal(2,1)});
\addplot3 [area plot,domain=-5:10] (x,-2,{1.329/sqrt(4*pi)/
(1+x^2/4)^2.5)); % Стьюдента
\addplot3 [fill opacity=0.5,draw=red,fill=pink!50,mark=none]
coordinates{(-1,-3,0) (0,-3,0.5) (3,-3,0)}; % трикутний
\addplot3 [fill opacity=0.5,draw=red,fill=lime!50,mark=none]
coordinates{(-4,-4,0) (-4,-4,0.15) (7,-4,0.15) (7,-4,0)};
\end{axis}
\end{tikzpicture}

```

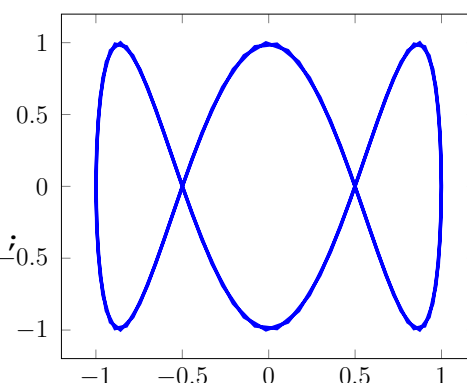


Під час теоретичних досліджень часто доводиться використовувати функції, що задані параметрично. Використання пакету TikZ значно спрощує їх побудову. Нижче наведено одну з фігур Лісажу

```

\begin{tikzpicture}
\begin{axis}
\addplot +[mark=none,ultra
thick,samples=100,domain=0:2*pi]
({cos(deg(2*x))},{sin(deg(6*x))});
\end{axis}
\end{tikzpicture}

```

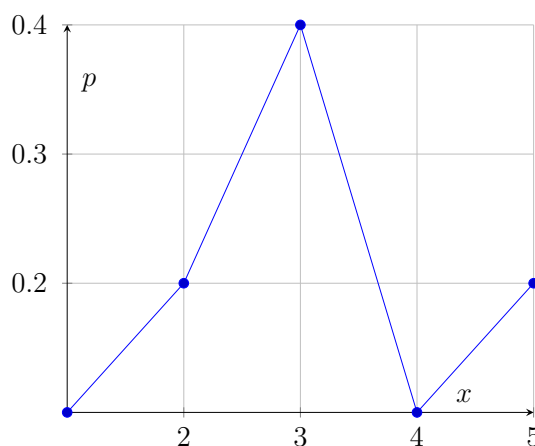


Під час наукових досліджень аналітичне зображення залежностей, що вивчають, не є відомими. Однак можна виявити певні її значення під час експериментальних досліджень. У цьому випадку ми матимемо множину експериментальних точок у вигляді таблиці або файлу прийнятного формату, найчастіше txt. Подібні графіки будуються через з'єднання точок, для надання такому графіку плавності, використовують опцію smooth.

```

\begin{tikzpicture}
\begin{axis}[xlabel={$x$},
ylabel={$p$}, axis lines=middle,
xstep=1,ystep=0.1,grid=major,
mark=none]
%задаються координати точок для
побудови графіка
%координати точок задаються у
форматі:
(x,y)
\addplot coordinates
{(1,0.1) (2,0.2) (3,0.4) (4,0.1)
(5,0.2)};
\end{axis}
\end{tikzpicture}

```



Наведені приклади є наочними і зручними для використання під час оформлення результатів досліджень.

## Висновки

Таким чином, використання видавничої системи  $\text{\LaTeX}$  та пакету розширення TikZ надає широкі можливості щодо оформлення науково-технічних документів, зокрема з візуалізації двох- та тривимірних даних.

## Список використаних джерел

- [1] Манако А. Ф., Синица К. М. ИКТ в обучении: взгляд сквозь призму трансформаций. *Образовательные технологии и общество*. 2012. Т. 15, № 3. С. 392–414.
- [2] Про Концепцію Національної програми інформатизації : Закон України від 04.02.1998 р. № 75/98-ВР. *Відомості Верховної Ради*. 1998. № 27–28. Ст. 182.
- [3] Крохмаль Т. М., Нікітенко О. М. Впорядкування звітів про результати дослідницьких робіт засобами  $\text{\LaTeX}$ . *Проблеми та інновації в природничо-математичній, технологічній і професійній освіті* : матеріали XII-ї Міжнародної науково-практичної онлайн-інтернет конференції. Кропивницький, 2021. С. 33–34.
- [4] Соколова А. Н., Чупраков Д. В. Оформление результатов исследовательской работы студентов в  $\text{\LaTeX}$  : учебное пособие. Киров: Изд-во ВятГГУ, 2013.
- [5] Лисенко С. М., Кришук А. Ф., Дзюбак Ю. П. Дослідження переваг застосування  $\text{\LaTeX}$  при оформленні наукових праць. *Вісник Хмельницького національного університету*. 2012. № 5. С. 225–234.
- [6] Грищенко Т. Б., Нікітенко О. М., Дейнеко Ж. В. Підготовка електронних підручників у системі  $\text{\LaTeX}$ . *«Поліграфічні, мультимедійні та web-технології» (PMW-2021)* : тези доповідей VI міжнародної науково-технічної конференції. Харків, 2021. С. 62–64.
- [7] Грищенко Т. Б., Нікітенко О. М., Дейнеко Ж. В. Створення електронних підручників засобами видавничої системи  $\text{\LaTeX}$  *Поліграфічні, мультимедійні та web-технології* : колективна монографія. Харків: «Друкарня Мадрид», 2021. С. 80–96.
- [8] Tantau T. The TikZ and PGF Packages. Manual for version 3.0.1a. August 29, 2015. Institut für Theoretische Informatik Universität zu Lubeck, 2015.
- [9]  $\text{\LaTeX}$  в дії. Методичні рекомендації з використання видавничої системи  $\text{\LaTeX}$  для студентів, науковців, викладачів / Упоряд. О. М. Нікітенко. Харків, 2018.

**Парамонова К.О., Шевченко Н.Ю.**

<sup>1</sup> магістрант факультету автоматизації машинобудування та інформаційних технологій Донбаської державної машинобудівної академії, Краматорськ

<sup>2</sup> к. е. н., доцент кафедри інтелектуальних систем прийняття рішень Донбаської державної машинобудівної академії, Краматорськ

*e-mail:* <sup>1</sup> ekaterinaparamonova5230@gmail.com, <sup>2</sup> natalygustav@gmail.com

## **МОДУЛЬ ГЕНЕРАЦІЇ ЕЛЕМЕНТІВ СТЕГОГРАФІЧНОЇ СИСТЕМИ ДЛЯ ЗАХИСТУ ІНФОРМАЦІЇ ПРИ ПЕРЕДАЧІ ДАНИХ ВІДКРИТИМИ КАНАЛАМИ ЗВ'ЯЗКУ**

Описано алгоритм підвищення інформаційної безпеки даних, який передбачає шифрування інформації та приховування даних у стегоконтейнері. В якості стегоконтейнеру запропоновано використовувати стохастичні фрактали «Плазма». Шифрування даних відбувається алгоритмом RSA, а приховування зашифрованого повідомлення здійснюється в молодших бітах фрактала методом LSB.

**Ключові слова:** *криптографія, стеганографія, фрактал.*

### **Вступ**

Передача інформації, а особливо конфіденційної, відкритими каналами зв'язку завжди супроводжується ризиком атаки на неї. Тому для збереження даних, а також забезпечення обмеженого доступу до них, використовують два способи програмного захисту: криптографічні та стеганографічні методи.

Сутність криптографічного захисту полягає в тому, що інформація зашифровується певним алгоритмом в нечитабельний формат. Стеганографічний захист – це приховування самого факту існування інформації шляхом вбудовування її в цифрові об'єкти (контейнери). Поєднання цих методів забезпечує надійність передачі даних та дозволяє зберегти її конфіденційність.

### **Математична модель**

В загальному випадку стегосистему можна представити як сукупність контейнерів, повідомлень та перетворень, що їх зв'язують.

Припустимо, що в якості повідомлення передаються згенеровані випадковим чином дані: логін і пароль. Завжди контейнери обираються та-

ким чином, щоб заповнений контейнер майже не відрізнявся від порожнього контейнера. В якості візуальних контейнерів можуть використовуватися фрактали. Серед поширених фракталів виділяють три основні групи: алгебраїчні, геометричні та стохастичні.

Алгоритмічна модель для безпечної передачі даних через відкритий канал зв'язку передбачає реалізацію п'яти етапів: генерація логіна і пароля; побудова стохастичного фрактала «Плазма»; шифрування логіна і пароля алгоритмом RSA; приховування зашифрованого повідомлення в молодших бітах зображення фрактала методом LSB (Least Significant Bits); відправлення зображення на електронну адресу користувача; розшифрування та витягування логіну та паролю з фрактала «Плазма».

Для побудови стохастичного фракталу «Плазма» використовується алгоритм Diamond Square [1]. Спочатку чотирьом кутам квадрата присвоюються випадкові величини. Після того, як задані межі квадрата, він розбивається на чотири рівних квадрати, в кожному з яких відомо значення одного з кутів. Значення висоти центральної точки – сума усереднення висот всіх чотирьох кутових точок і випадкового значення (шуму), що зображено на рис. 1.

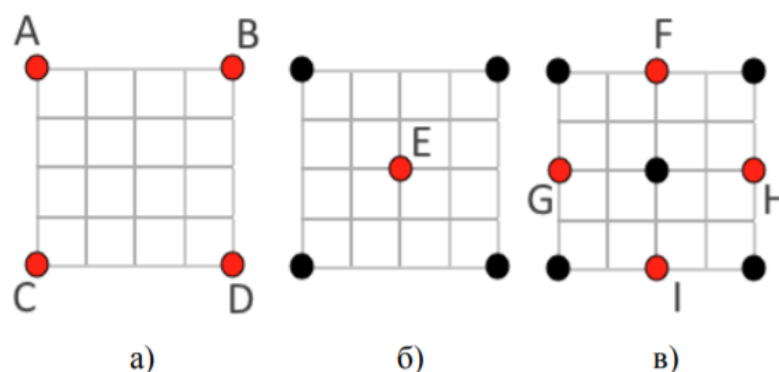


Рис. 1: Алгоритм Diamond Square: а) висота в кутових точках; б) значення в центральній точці; в) значення на середині сторін.

Алгоритм RSA складається з 4 етапів [2]: генерації ключів, шифрування, розшифрування та розповсюдження ключів. Безпека алгоритму RSA побудована на принципі складності факторизації цілих чисел. Алгоритм використовує два ключі – відкритий (public) і закритий (private), разом відкритий і відповідний йому закритий ключі утворюють пари ключів (keypair). Якщо повідомлення було зашифровано відкритим ключем, то розшифрувати його можна тільки відповідним закритим ключем.

Сутність методу заміни найменш значущого біта LSB полягає в приховуванні інформації шляхом зміни останніх бітів зображення, які кодують колір на біти приховуваного повідомлення. У форматі BMP зображення

зберігається як матриця значень відтінків кольору для кожної точки зображення.

Для реалізації наведеного алгоритму приховування та шифрування даних був спроектований та розроблений відповідний програмний модуль.

Діаграма варіантів використання модуля для генерації окремих елементів стеганографічної системи наведена на рис. 2.

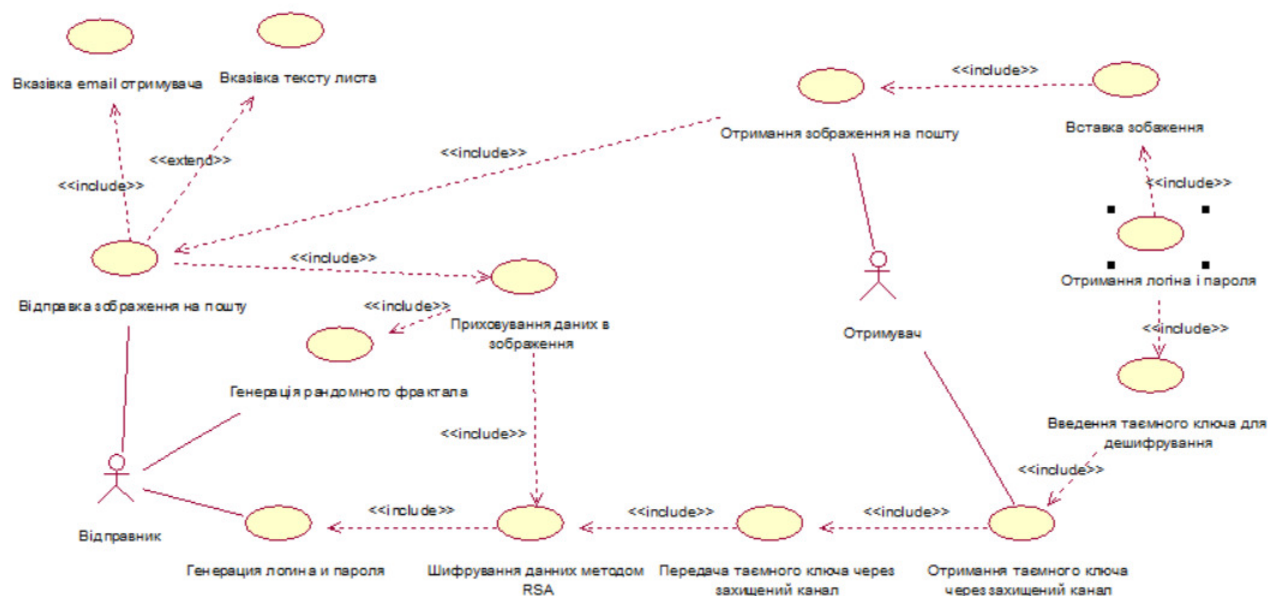


Рис. 2: Діаграма варіантів використання.

На даній діаграмі дійовими особами або акторами є відправник та отримувач таємного повідомлення. Зв'язком простої асоціації актор «Відправник» пов'язаний з прецедентами «Генерація випадкового фрактала» і «Генерація логіна і пароля». При виборі «Генерація випадкового фрактала» відправник може приховати дані в зображення, відправити зображення на пошту. При цьому введення електронної пошти отримувача є обов'язковою дією, а введення тексту листа – ні. При виборі варіанту «Генерація логіна і пароля» відправник може зашифрувати дані та відправити таємний ключ отримувачу. Приховування інформації в зображенні неможливе без шифрування даних, які були згенеровані на етапі зв'язку простої асоціації «Генерація логіна і пароля». Зв'язком простої асоціації актор «Отримувач» пов'язаний з прецедентами «Отримання зображення через пошту» і «Отримання таємного ключа через захищений канал». Отримання зашифрованих даних включає в себе введення таємного ключа та завантаження зображення.

Приклад роботи модуля наведений на рис. 3.

На вкладці «Приховування даних» проходить повний цикл дій для отримання готового зображення із зашифрованими даними для ролі «Від-

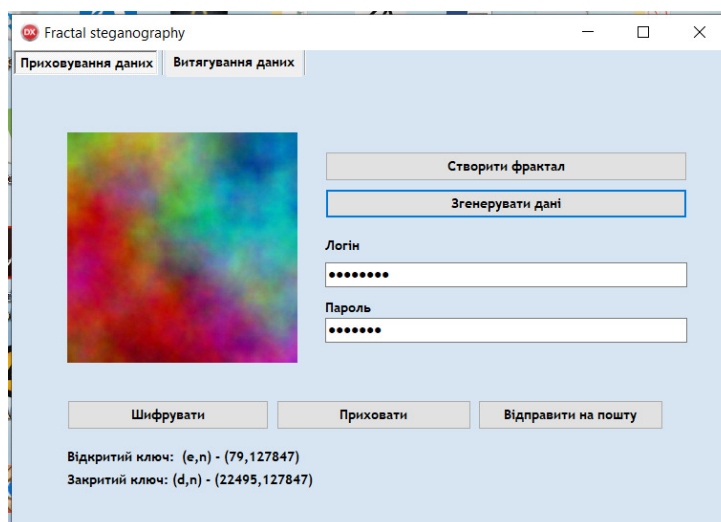


Рис. 3: Генерація логіну та паролю.

правник». На початку треба натиснути на кнопку «Створити фрактал» для генерації стохастичного фрактала «Плазма». Після відображення фракталу на формі для генерації логіну та паролю потрібно натиснути на кнопку «Генерувати дані». Дані невидимі для відправника, захищені зірками. Наступним кроком циклу дій для отримання готового зображення із зашифрованими даними (приховування даних) буде шифрування логіну та паролю методом RSA. Для цього треба натиснути на кнопку «Шифрувати». Після цієї дії нижче кнопки з'являється відкритий та закритий ключі. Закритий ключ передається користувачу через захищений канал зв'язку. Після шифрування треба натиснути на кнопку «Приховати». Відбудеться приховування даних у останні біти згенерованого раніше фракталу методом LSB, растрівання та зберігання зображення. Для відправки збереженого зображення на електронну пошту треба натиснути на кнопку «Відправити на пошту».

На вкладинці «Витягування даних» проходить повний цикл дій для отримання логіна та пароля для ролі «Отримувач». На початку треба перейти до своєї електронної пошти та знайти там відповідне повідомлення з прикріпленим зображенням (рис. 4). Повідомлення повинно мати: тему «Фрактал», текст, який надрукував «Відправник», а також прикріплений малюнок стохастичного фрактала «Плазма».

Далі потрібно натиснути на кнопку «Дешифрування» для того, щоб система почала витягувати дані з зображення, а потім дешифрувати за допомогою закритого ключа. Після описаних вище дій в полях «Логін» та «Пароль» з'являться передані користувачеві вхідні дані.

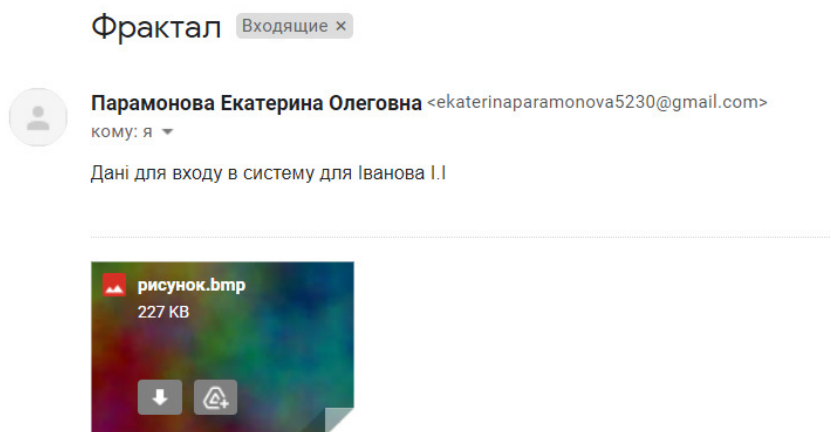


Рис. 4: Повідомлення, яке прийшло на e-mail отримувача.

## Висновки

Отже, в умовах необхідності підвищення інформаційної безпеки даних, що передаються відкритими каналами зв'язку доцільно поєднувати криптографічні та стеганографічні методи, генеруючи при цьому для підвищення якості приховування даних контейнери – стохастичні фрактали.

## Список використаних джерел

- [1] Fournier A., Fussell D., Carpenter L. Computer rendering of stochastic models. *Communications of the ACM*. 1982. Vol. 25, No. 6. P. 371–384.
- [2] Menezes A.J., van Oorschot P.C., Vanstone S.A. Handbook of Applied Cryptography. CRC Press, 1996.

<sup>1</sup> к. ф-м. н., учений секретар Інституту прикладної математики і механіки НАНУ, Слов'янськ

<sup>2</sup> аспірант відділу теорії керуючих систем Інституту прикладної математики і механіки НАНУ, Слов'янськ

e-mail: <sup>1</sup> sapunov\_sv@yahoo.com, <sup>2</sup> eveltves@gmail.com

## РОЗПІЗНАВАННЯ РОЗМІТКИ ВЕРШИН ПОЗНАЧЕНОГО ГРАФА КВАДРАТНОЇ РЕШІТКИ

В роботі розглядається задача розпізнавання розмітки нескінченного (в усі сторони) графа квадратної решітки за допомогою графохідного автомата. Автомат отримує на свій вхід позначки усіх вершин із замкненого околу поточної вершини та пересувається по між суміжними вершинами, вибираючи цільову вершину за її позначкою. Автомат не має компасу, тобто розрізняє вершини тільки за їх позначками і не розрізняє взаємне розташування вершин. Априорі є відомою множина позначок вершин, а також те, що функція розмітки є детермінованою, тобто усі вершини із замкненого околу будь-якої вершини мають різні позначки. Доведено, що автомат з одним камінцем вирішує задачу розпізнавання. Побудовано метод та алгоритм розв'язання цієї задачі.

**Ключові слова:** *графохідний автомат, позначений граф, граф квадратної решітки, розпізнавання розмітки.*

## Вступ

Основною проблемою теоретичної кібернетики є проблема взаємодії керуючої та керованої систем (керуючого автомата та його операційного середовища) [1, 2]. Взаємодія таких систем часто представляється як процес пересування автомата на графі або лабіринті середовища [3, 4]. Однією з центральних та актуальних як в теоретичному, так і в прикладному аспектах проблем, які виникають при дослідженні взаємодії автоматів та графів, є проблема аналізу або розпізнавання властивостей графа за різної апріорної інформації та за різних способів взаємодії автомата та графа. Один із підходів до розв'язання цієї проблеми ґрунтується на тому, що операційне середовище розглядається як граф з позначеними вершинами.

В роботі розглядається наступна задача. Априорі відомо що усі вершини нескінченного (в усі сторони) графа двовимірної квадратної решітки позначено позначками з деякої відомої множини, але конкретна функція розмітки невідома. Графохідний автомат встановлюється на довільній вершині цього графа. На свій вхід автомат отримує інформацію щодо

позначок поточної вершини та вершин з її околу. Виходом автомата є пересування на суміжну вершину, яку він вибирає за її позначкою. Завданням автомата є визначення двох двійок протилежних напрямків на укладанні графа на цілочисельну решітку, або розпізнавання розмітки. Після такого розпізнавання автомат отримує можливість спрямовано пересуватися на графі.

## Означення

Нехай  $\mathbb{Z}$  позначає множину усіх цілих чисел, а  $\mathbb{N}$  – множину усіх натуральних чисел.

Ми використовуватимемо стандартну термінологію для графів (за довідками відсилаємо читача до [5]). Нескінченим в обидві сторони графом шляху називається нескінченне дерево, кожна вершина якого має ступінь 2. Нескінченим (в усі сторони) графом двовимірної квадратної решітки  $G_{\boxplus}$  називається декартів квадрат нескінченного в обидві сторони графа шляху. Ми розглядатимемо укладку  $G_{\boxplus}(\mathbb{Z}^2)$  графа двовимірної квадратної решітки на цілочисельній решітці  $\mathbb{Z}^2$ . При цьому вершини відповідають різним точкам  $\mathbb{Z}^2$  та дві вершини з'єднані ребром тоді і тільки тоді, коли відстань між відповідними точками дорівнює 1. Укладена нескінченна решітка є природною дискретною версією евклідової площини. Покладемо, що іменами вершин графа  $G_{\boxplus}(\mathbb{Z}^2)$  є координати відповідних точок на решітці  $\mathbb{Z}^2$ . Напрямок на графі  $G_{\boxplus}(\mathbb{Z}^2)$  відносно поточної вершини назвемо довільний вектор з двох взаємно простих цілих чисел. Будемо казати, що вершина  $v' = (b_1, b_2)$  знаходиться в напрямку  $(d_1, d_2)$  від вершини  $v = (a_1, a_2)$ , якщо  $b_1 = a_1 + kd_1$  і  $b_2 = a_2 + kd_2$ , де  $k \in \mathbb{N}$ .

Позначеним графом називатимемо простий зв'язний граф з позначеними вершинами  $G = (V, E, M, \mu)$ , де  $V$  – множина вершин,  $E$  – множина ребер,  $M$  – множина позначок,  $\mu : M \rightarrow V$  – сюр'єктивна функція розмітки. Граф із застосованою до нього функцією розмітки також називатимемо розміткою вихідного графа, або, скорочено, розміткою. Околом вершини  $v$  в графі  $G$  називається множина  $N_v$  усіх вершин в  $G$ , суміжних з  $v$ . Замкненим околом вершини  $v$  в графі  $G$  називається множина  $N_{[v]} = N_v \cup \{v\}$ .

Ізоморфізмом позначених графів  $G_1 = (V_1, E_1, \mu_1)$  та  $G_2 = (V_2, E_2, \mu_2)$  називається бієкція  $\varphi : V_1 \rightarrow V_2$ , за якої виконуються такі дві умови:

- (1)  $\forall v', v'' \in V_1 ((v', v'') \in E_1 \Leftrightarrow (\varphi(v'), \varphi(v'')) \in E_2)$ ;
- (2)  $\forall v \in V_1 (\mu_1(v) = \mu_2(\varphi(v)))$ .

За допомогою  $M^+$  позначимо множину усіх непорожніх слів над ал-

фавітом  $M$ . Інверсією слова  $w = a_1 \dots a_k$  назвемо слово  $w^{rev} = a_k \dots a_1$ . Введемо часткову операцію  $\star : V \times M^+$  за допомогою співвідношення: для будь-якої вершини  $v \in V$  та будь-якого слова  $w \in M^+$  за допомогою  $v \star w$  позначимо множину усіх вершин  $v' \in V$  таких, що існує шлях із  $v$  до  $v'$ , позначений словом  $w$ . Нехай  $a, b \in M$  і  $u, w \in M^+$ , тоді за допомогою  $wa \circ bu$  позначимо слово  $waui$ , якщо  $a = b$ , інакше, якщо  $a \neq b$ , то  $wa \circ bu$  невизначено.

Послідовність позначок вершин  $w = \mu(v_1) \dots \mu(v_k)$ , яка відповідає деякому шляху  $p = v_1 \dots v_k$  в графі  $G$ , назвемо словом довжини  $k$ , що визначається вершиною  $v_1$ . Мовою  $L_v$  вершини  $v$  назвемо множину усіх слів, які визначаються цією вершиною. Мовою  $L_G$  позначеного графа  $G$  назвемо множину слів  $\bigcup_{v \in V} L_v$ .

Графохідним автоматом на графі  $G$  назвемо шістку  $A = (S, X, Y, s_0, \delta, \lambda)$ , де  $S$  – скінчена множина внутрішніх станів,  $X = \{(a_0, \{a_1, \dots, s_k\})\}$  – скінчений вхідний алфавіт (де  $a_0$  – позначка поточної вершини,  $\{a_1, \dots, a_k\}$  – множина (або мультимножина) позначок усіх вершин з околу поточної вершини,  $k$  – ступінь поточної вершини),  $Y = M$  – скінчений вихідний алфавіт ( $y = a, y \in Y$  означає, що автомат пересувається з поточної вершини на суміжну до неї вершину з позначкою  $a$ ),  $s_0 \in S$  – початковий стан,  $\delta : S \times X \rightarrow S$  – функція переходів,  $\lambda : S \times X \rightarrow Y$  – функція виходів. Автомат функціонує наступним чином: оглядає позначки усіх вершин з колу поточної вершини, вибирає деяку позначку та пересувається на вершину з цією позначкою. Автомат не має компасу, що означає, що він не розрізняє напрямки та взаємне розташування вершин. Таким чином, автомат не розрізняє вершини з однаковими позначками. В [6] показано, що поодинокий автомат не може підтримувати напрямок руху на графі, у якого усі вершини непозначені або, що еквівалентно, позначені однією і тією самою позначкою.

Нехай автомат  $A$  в момент часу  $t$  знаходиться на вершині  $v(t)$  графу  $G(\mathbb{Z})$ . Пересування автомата називатимемо рівномірним та спрямованим, якщо існує натуральний період  $T$  такий, що  $v(t+T) - v(t) = v(t+2T) - v(t+T)$  виконується для будь-якого моменту часу  $t$ .

Експериментом з графом  $G$  відносно апріорної інформації  $I$  і мети  $C$  з використанням засобів  $S$  назвемо процес, який складається з трьох етапів: (етап 1) побудова тесту  $P \subseteq M^+$  на основі  $I$  із урахуванням  $C$ ; (етап 2) отримання породжуваних тестом  $P$  із використанням  $S$  експериментальних даних  $W$ ; (етап 3) виведення висновків щодо властивостей графа на основі  $W$  та  $I$ . Апріорна інформація – це клас графів, до якого нале-

жить  $G$ . Основна відмінність експериментів з графами від експериментів з автоматами [7] полягає в методах і засобах отримання експериментальних даних, а саме, можливостях графохідного автомата пересуватися на графі, сприймати локальну інформацію щодо його вершин та ставити на них додаткові стирані/нестирані позначки.

Експеримент назвемо розпізнавальним, якщо відома множина позначок, відомий тип розмітки, графохідний автомат встановлено на довільну початкову вершину, а метою є розпізнавання розмітки з точністю до ізоморфізму позначених графів.

Розглядатимемо наступні міри складності експериментів. Операційною кратністю експерименту назвемо величину, яка дорівнює, в залежності від способу проведення експерименту, або кількості екземплярів автомата, використаних при отриманні  $W$ , або кількості встановлень автомата експериментатором на початкову вершину досліджуваного графа. Якщо використовується тільки один екземпляр автомата (одно встановлення), то експеримент назвемо простим. Операційною висотою експерименту назвемо найбільшу із довжин шляхів, які проходить кожен з екземплярів автомата протягом експерименту.

## Прохідна розмітка вершин

Оберемо на графі  $G_{\boxplus}(\mathbb{Z}^2)$  дві двійки протилежних напрямків відповідно до осей абсцис та ординат площини  $\mathbb{Z}^2$ . Ці напрямки у подальшому називатимемо базовими. Легко бачити, що пересування автомата у будь-якому фіксованому напрямку, відмінному від базових, можна представити як комбінацію пересувань у базових напрямках. Отже, ми можемо обмежитися розглядом пересувань автомата лише у базових напрямках. Введемо необхідні додаткові поняття. Розмітку називатимемо періодичною у напрямку  $(q, r) \in \mathbb{Z}$ , якщо  $\mu((i + q, j + r)) = \mu((i, j))$  для усіх  $i, j \in \mathbb{Z}$ . Розмітку називатимемо прохідною, якщо за її допомогою деякий графохідний автомат може пересуватися по графу у будь-якому напрямку. Розмітка, яка використовує мінімальну кількість позначок необхідну для даного графа, називається мінімальною розміткою цього графа.

Розмітку називатимемо детермінованою (або Д-розміткою), якщо усі вершини у замкненому околі будь-якої вершини мають різні позначки. Граф з детермінованою розміткою називатимемо детермінованим (або Д-графом). В [8] було доведено наступні дві властивості, які безпосередньо впливають з означення Д-графа: (1) для будь-якої вершини такого графа будь-яке слово в алфавіті позначок визначає не більше одного шляху з

цієї вершини; (2) відстань між двома вершинами з однаковими позначками більше або дорівнює трьом (чотирьом, якщо рахувати у вершинах). Ці властивості відкривають можливість спрямованого пересування графові-дного автомата на Д-графі. Наприклад, якщо нам відома розмітка шляху, який з'єднує дві вершини такого графа, то ми можемо побудувати автомат, який може пройти цей шлях.

В [9] доведено, що для побудови мінімальної прохідної детермінованої розмітки (або, скорочено, МПД-розмітки) графа  $G_{\boxplus}$  необхідно і достатньо мати п'ять різних типів позначок. Також для такої розмітки було показано, що рівності  $\mu((i, j)) = \mu((i + 5k, j))$  та  $\mu((i, j)) = \mu((i, j + 5l))$  виконуються для усіх  $i, j, k, l \in \mathbb{Z}$ . Таким чином, мінімальну прохідну детерміновану розмітку нескінченного графа квадратної решітки можна задати із використанням прямокутного шаблону із  $5 \times 5$  позначених вершин.

Зафіксуємо множину позначок  $M = \{a, b, c, d, e\}$ . Безпосередньою перевіркою можна встановити, що за допомогою цих позначок можна побудувати 240 різних МПД-розміток графа  $G_{\boxplus}$ . Оскільки в розмітці околу кожної вершини МПД-розмітки графа  $G_{\boxplus}$  присутні усі позначки з множини  $M$ , то можна припустити, що автомат завжди встановлюється на вершину з однією і тією самою позначкою, наприклад, позначкою « $a$ ». Дійсно, якщо позначка початкової вершини відрізняється від « $a$ », то автомат завжди може перейти на вершину з цією позначкою, знайшовши її в околі поточної вершини. Тоді можна розглядати лише 48 різних МПД-розміток графа  $G_{\boxplus}$ . Оскільки автомат спостерігає лише позначки вершин і не може визначити їх імена, то він не може відрізнити один від одного ізоморфні позначені графи. Розіб'ємо усі розмітки графа  $G_{\boxplus}$  на класи, кожен з яких складається з усіх розміток, які можна перетворити одна в одну за допомогою симетричних перетворень. У такий спосіб усі можливі МПД-розмітки можна розбити на шість класів. Представники (позначені шаблони  $5 \times 5$ ) кожного із класів зображені на рис. 1. За кожним представником можна відновити весь його клас за допомогою симетричних перетворень.

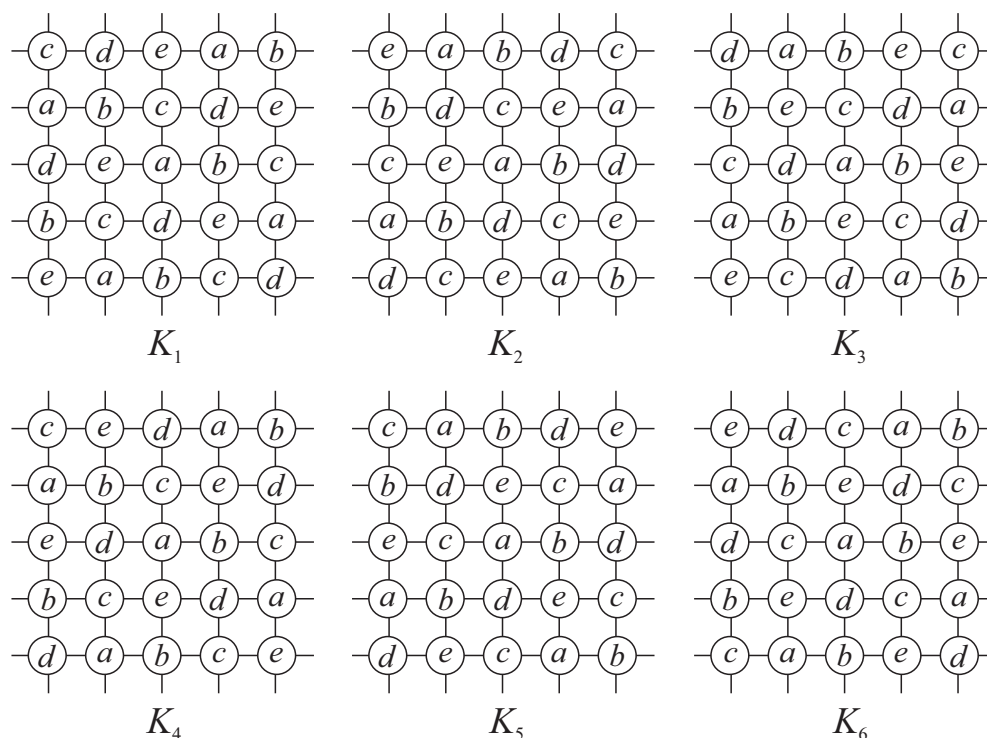


Рис. 1: Представники класів ізоморфних графів.

**Теорема 1.** При фіксованій множині позначок будь-які представники різних класів симетрії МПД-розміток графа  $G_{\boxplus}$  неізоморфні.

**Наслідок 1.** Множина усіх МПД-розміток графа  $G_{\boxplus}$  з фіксованою множиною позначок, розбивається на шість класів еквівалентності, яка породжується ізоморфізмом позначених графів.

## Розпізнавання розмітки

**Теорема 2.** Будь-яка МПД-розмітка графа  $G_{\boxplus}$  визначає одну і ту саму мову, з умови, що множина позначок є фіксованою.

**Наслідок 2.** Один автомат не може відрізнити одну від одної розмітки графа  $G_{\boxplus}$ , які належать до різних класів розмітки з фіксованою множиною позначок.

Справедливість цього твердження впливає із означення автомату без компаса, який спостерігає лише позначки вершин і не розрізняє їх імена.

Узагальнимо поняття розпізнавального експерименту наступним чином. Нехай автомат до початку експерименту розташовує на початковій вершині досліджуваного графа унікальний маркер (т. з. камінець). Таким чином автомат при відвідуванні цієї вершини може відрізнити її від усіх інших вершин за наявністю на ній камінця. Отже, автомат з камінцем може визначити, чи відповідає задане слово циклічному шляху по заданому

графу. На рис. 2(а) плюс у комірці таблиці означає, що слово, яким позначено рядок, визначає цикл на усіх розмітках з класу, яким позначено стовпчик.

**Теорема 3.** Множина слів  $\{abdca, abeda, abcd a, abeca\}$  є розпізнавальним тестом для будь-якої МПД-розмітки графа  $G_{\boxplus}$  з множиною позначок  $\{a, b, c, d, e\}$ .

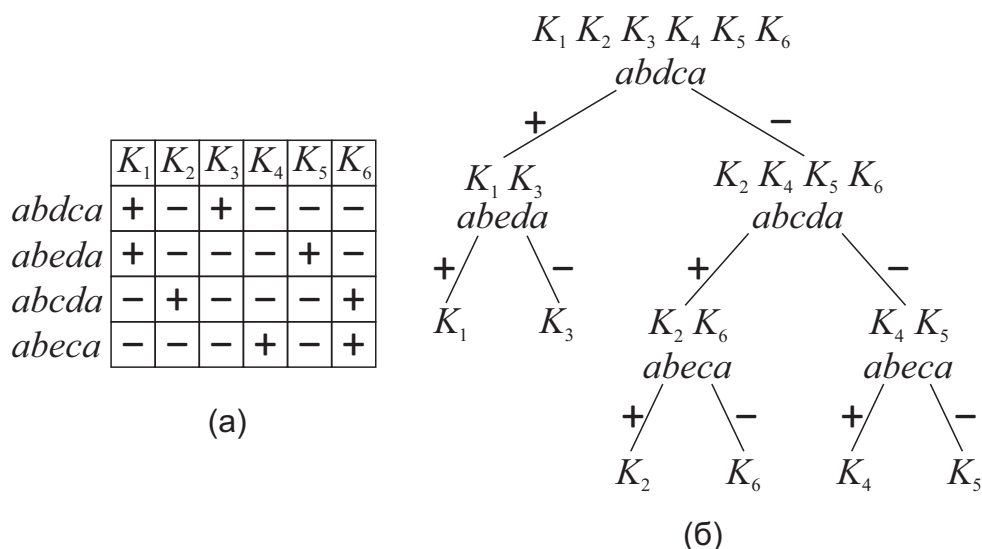


Рис. 2: Розпізнавання розмітки графа.

Етапи розпізнавального експерименту, породженого тестом  $P$ , опишемо за допомогою такого алгоритму. Нехай  $M = \{a, b, c, d, e\}$ . Покладемо  $w_1 = abdca$ ,  $w_2 = abeda$ ,  $w_3 = abcd a$ ,  $w_4 = abeca$ . Позначимо за допомогою  $\mathcal{L}$  розбиття  $\{K_1, K_2, K_3, K_4, K_5, K_6\}$  МПД-розміток графа  $G_{\boxplus}$  на класи еквівалентності, які породжуються ізоморфізмом позначених графів.

#### Алгоритм 1. РОЗПІЗНАВАННЯ РОЗМІТКИ

**Дано:**  $P = \{w_1, w_2, w_3, w_4\}$ ,  $\mathcal{L} = \{K_1, K_2, K_3, K_4, K_5, K_6\}$ , автомат встановлено на вершину  $v$  з позначкою  $a$

**Треба:** клас розмітки досліджуваного графа

- 1: Автомат проходить шлях, який відповідає слову  $w_1$
- 2: **if** на вершині  $v \star w_1$  є камінець **then**
- 3:      $\mathcal{L} = \{K_1, K_3\}$
- 4:     Автомат проходить шлях, який відповідає слову  $w_2$
- 5:     **if** на вершині  $v \star w_2$  є камінець **then**
- 6:          $\mathcal{L} = \{K_1\}$
- 7:     **else**
- 8:          $\mathcal{L} = \{K_3\}$
- 9:     **end if**

```

10: else
11:     Автомат проходить шлях, який відповідає слову  $w_1^{rev}$ 
12:     Автомат проходить шлях, який відповідає слову  $w_3$ 
13:     if на вершині  $v \star w_3$  є камінець then
14:          $\mathcal{L} = \{K_2, K_6\}$ 
15:         Автомат проходить шлях, який відповідає слову  $w_4$ 
16:         if на вершині  $v \star w_4$  є камінець then
17:              $\mathcal{L} = \{K_2\}$ 
18:         else
19:              $\mathcal{L} = \{K_6\}$ 
20:         end if
21:     else
22:          $\mathcal{L} = \{K_4, K_5\}$ 
23:         Автомат проходить шлях, який відповідає слову  $w_3^{rev}$ 
24:         Автомат проходить шлях, який відповідає слову  $w_4$ 
25:         if на вершині  $v \star w_4$  є камінець then
26:              $\mathcal{L} = \{K_4\}$ 
27:         else
28:              $\mathcal{L} = \{K_5\}$ 
29:         end if
30:     end if
31: end if

```

Описаний експеримент є простим, а його операційна висота дорівнює довжині слова *abdcacdbabcbdadcbabeca*, тобто дорівнює 21. Процес виконання алгоритму схематично зображено у вигляді дерева на рис. 2(б).

## Висновки

Таким чином, у роботі запропоновано метод побудови та проведення експерименту з розпізнавання мінімальної прохідної детермінованої розмітки нескінченного позначеного графа квадратної решітки. У подальших дослідженнях планується визначити необхідні і достатні умови, за яких розмітка графа квадратної решітки є прохідною для графохідного автомата.

## Список використаних джерел

- [1] Novikov D.A. Cybernetics: From Past to Future. Heidelberg: Springer, 2016.

- [2] Kline R.R. The cybernetics moment: Or why we call our age the information age. Baltimore: Johns Hopkins University Press, 2015.
- [3] Килибарда Г., Кудрявцев В.Б., Ушчумлич Ш.М. Независимые системы автоматов в лабиринтах. *Дискретная математика*. 2003. Т. 15, вып. 2. С. 3–39.
- [4] Килибарда Г., Кудрявцев В.Б., Ушчумлич Ш.М. Коллективы автоматов в лабиринтах. *Дискретная математика*. 2003. Т. 15, вып. 3. С. 3–39.
- [5] Diestel R. Graph Theory, 5th ed. Heidelberg: Springer, 2017.
- [6] Курганский А.Н., Сапунов С.В. О направленном перемещении коллектива автоматов без компаса на одномерной целочисленной решетке. *Изв. Саратов. ун-та. Нов. сер. Сер. Математика. Механика. Информатика*. 2016. Том 16, вып. 3. С. 356–365.
- [7] Kohavi Z., Jha N.K. Switching and Finite Automata Theory, 3rd ed. Cambridge: Cambridge University Press, 2009.
- [8] Грунский И.С., Сапунов С.В. Восстановление графа операционной среды мобильного робота путем разметки вершин, пригодной для дальнейшей навигации. *Искусственный интеллект*. 2012. Том 4. С. 420–428.
- [9] Sapunov S.V. Collectives of automata on infinite grid graph with deterministic vertex labeling. *Праці ІПММ НАН України*. 2019. Том 33. С. 176–193.

## ПРО ГРАЦІОЗНУ ТА РІЗНИЦЕВУ КВАДРАТНУ РОЗМІТКИ ГРАФІВ

Узагальнено спосіб переносів для побудови граціозних дерев на графи певного виду. Представлена операція розширення графа і доведено умову існування різницевої квадратної розмітки для графа, отриманого за допомогою цієї операції.

**Ключові слова:** *граф, граціозна розмітка, різницева квадратна розмітка.*

### Вступ

Розглядатимемо скінчені неорієнтовані графи без петель та кратних ребер. Під оптимальною нумерацією (або розміткою) графа розуміють присвоєння номерів його вершинам таким чином, щоб сума модулів різниці між номерами суміжних вершин була мінімальною. Задача побудови такої нумерації відноситься до задач комбінаторної оптимізації. Інтерес викликають різні варіанти оптимальної нумерації, у тому числі граціозна та різницева квадратна розмітки. Більше інформації з цього питання можна знайти в [1]. В даній роботі представлено один з методів побудови граціозної та різницевої квадратної розмітки графів, також доведена умова існування квадратних різницевих графів певного виду.

### Про граціозну та різницеву квадратну розмітку

Ін'єктивну функцію  $f : V \rightarrow \{0, 1, 2, \dots, q\}$  називають *граціозною розміткою* графа  $G = (V, E)$  розміру  $q$ , якщо вона індукує таку реберну розмітку  $f^* : E \rightarrow \{1, 2, 3, \dots, q\}$ , що  $f^*$  – бієкція і  $f^*(uv) = |f(u) - f(v)|$  для будь-яких суміжних вершин  $u, v \in V$ . Граф  $G$  – *граціозний*, якщо він допускає граціозну розмітку.

Для побудови граціозних дерев з дерев менших порядків існують декілька версій методу переносів. Зупинимося на методі запропонованому в [2]. Узагальнимо його на графи певного виду. Розглянемо такий граф  $G = (V, E)$  в якому можна вибрати вершини  $u, v, u_1, u_2 \in V$  так, щоб

$uu_1, uu_2 \in E$  і  $u_1u_2 \notin E$ ,  $\deg u = 2$ . Припустимо, що існують підграфи  $G_1 = (V_1, E_1)$ ,  $G_2 = (V_2, E_2)$  і  $G_3 = (V_3, E_3)$  графа  $G$  з  $u, u_1 \in V_1$ ,  $u, u_2 \in V_2$ ,  $V_1, V_2 \subset V$ ,  $V_1 \cap V_2 = u$ ,  $E_1 \cap E_2 = \emptyset$ ,  $V_3 = (V - (V_1 \cup V_2)) \cup \{u\}$ . Граф  $H$  одержуємо переносом графа  $G_1 \cup G_2$  з вершини  $u$  в вершину  $v \in V_3, v \neq u$  (рис.1). Спосіб побудови графа назвемо *узагальненим методом переносів*.

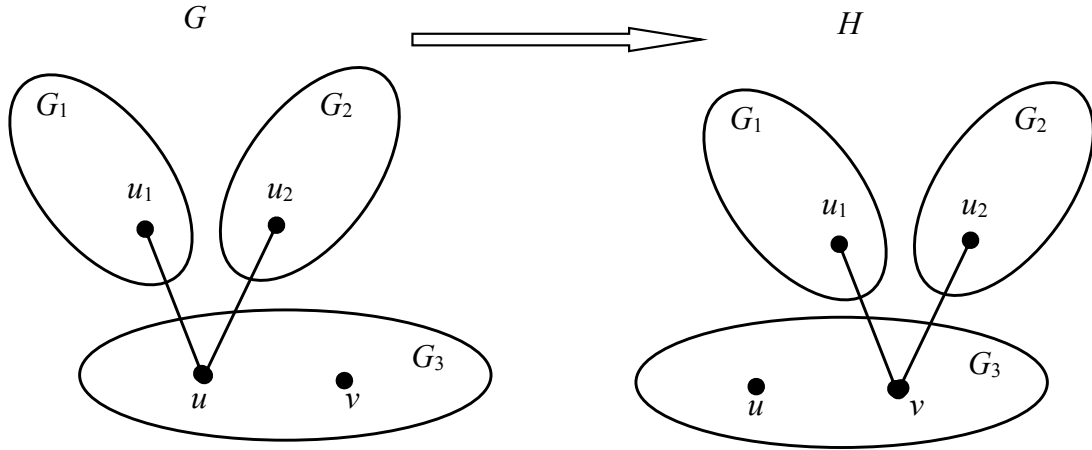


Рис. 1: Інтерпретація узагальненого методу переносів.

Справедлива лема 1.

**Лема 1.**

Нехай  $G = (V, E)$  – граф з граціозною розміткою  $f$  і граф  $H$  одержано узагальненим методом переносів. Якщо  $u_1 \neq u_2$  і

$$f(u_1) + f(u_2) = f(u) + f(v), \quad (1)$$

то  $f$  – граціозна розмітка графа  $H$ .

Якщо  $u_1 = u_2$  і

$$2f(u_1) = f(u) + f(v), \quad (2)$$

то  $f$  – граціозна розмітка графа  $H$ .

**Доведення.** Припустимо, що  $f$  – граціозна розмітка графа  $G = (V, E)$  з  $u, u_1, u_2 \in V, uu_1, uu_2 \in E, u_1u_2 \notin E$ . Для  $G = (V, E)$  і  $H = (V', E')$  маємо  $V' = V, E' = (E - \{uu_1, uu_2\}) \cup \{vu_1, vu_2\}$ . Нехай  $f^*$  – реберна розмітка, породжена  $f$ . Залишимо мітки вершин графа  $H$  без змін, тобто застосуємо розмітку  $f$  для  $V'$ , яка індукує на  $E'$  розмітку  $f^{**}$ . Мітки ребер з множини  $E - \{uu_1, uu_2\}$  для графів  $G$  і  $H$  співпадають. Виходячи з (1), для ребер  $vu_1, vu_2 \in E'$  при  $u_1 \neq u_2$  отримаємо:

$$f^*(u_1u) = |f(u_1) - f(u)| = |f(u) + f(v) - f(u_2) - f(u)| = f^{**}(vu_2),$$

$$f^*(u_2u) = |f(u_2) - f(u)| = |f(u) + f(v) - f(u_1) - f(u)| = f^{**}(vu_1).$$

Ліві частини двох останніх рівностей різні, тому різні і їх праві частини.

Якщо  $u_1 = u_2$ , то аналогічно отримаємо  $f^*(u_1u) = f^{**}(u_1v)$ .

В обох випадках реберна розмітка графа  $H$  є бієкцією з множини ребер  $E'$  в множину чисел  $\{1, 2, \dots, |E|\}$ . Отже,  $f$  – граціозна розмітка графа  $H$ .  $\square$

**Означення 1.** Для графа  $G = (V, E)$  порядку  $p$  бієктивне відображення  $f$  множини  $V$  в множину  $\{0, 1, 2, \dots, p-1\}$  назвемо  $n$ -ою різницевою розміткою ( $n \geq 2$ ), якщо  $f$  породжує таку реберну розмітку  $f^*$ , що  $f^*(uv) = f^n(u) - f^n(v)$  для будь-яких суміжних вершини  $u, v \in V$  і  $f^*$  – ін'єкція з множини  $E$  в множину натуральних чисел. Граф, що допускає таку розмітку назвемо  $n$ -им різницевим. Для різних значень  $n$  отримаємо частинні випадки введеної розмітки. Так при  $n = 2$ ,  $f$  є різницевою квадратною розміткою (або  $SD$  розміткою) [3].

Нехай задано граф  $G$ . Побудуємо граф  $G^{(1)}$ , додаванням до кожної вершини  $G$  одного висячого ребра. Цей процес приєднання висячих ребер можна продовжити далі і на  $s$ -м кроці отримати граф  $G^{(s)}$ , який назвемо  $s$ -розширенням  $G$ .

**Теорема 1.**  $G^{(s)}$  є квадратним різницевим графом для будь-якого  $s$  і різницевого квадратного графа  $G$ .

Теорему 1 доведено методом математичної індукції.

## Висновки

У роботі запропоновано способи побудови граціозних та різницевих квадратних графів з графів менших порядків.

## Список використаних джерел

- [1] Gallian J.A. A dynamic survey of graph labeling, 24th edition. *The Electronic Journal of Combinatorics* 2021. Vol. 24, #DS6.
- [2] Hrnčiar P., Haviar A. Haviar All trees of diameter five are graceful. *Discrete Mathematics*. 2001. Vol. 233, Issue 1–3. P. 133–150.
- [3] Ajitha V., Princy K.L., Lokesh V., Ranjini P.S. On Square Difference Graphs. *International Journal of Mathematical Combinatorics*. 2012. Vol.1. P. 31–40.

## ВЛАСТИВОСТІ ТЕОРЕТИКО-МНОЖИННИХ ОПЕРАЦІЙ У ТАБЛИЧНИХ АЛГЕБРАХ

Наведено взаємозв'язки перетину, об'єднання та різниці таблиць між собою та з іншими сигнатурними операціями табличних алгебр: проекцією, селекцією, перейменуванням, насиченням, активним доповненням, з'єднанням. Ці взаємозв'язки виражені у вигляді критеріїв дистрибутивності або критеріїв розповсюдження аналогів відомих законів теорії множин. Наведені критерії в основному виражені в термінах активних доменів вихідних таблиць та є природними.

**Ключові слова:** *перетин, об'єднання, різниця, табличні алгебри.*

### Вступ

В наш час бази даних широко використовуються майже в усіх сферах людської діяльності. При цьому, найбільш розповсюдженими залишаються табличні (реляційні) бази даних, що засновані на реляційних принципах та моделях Е. Кодда [1]. З математичної точки зору реляційна база даних є скінченим набором скінчених відношень різної розмірності між заздалегідь визначеними множинами елементарних даних – доменами. Табличні алгебри, які було введено В.Н. Редьком та Д.Б. Буєм [2], побудовані на основі реляційних алгебр Е. Кодда та суттєво їх уточнюють. Вони складають теоретичний фундамент мов запитів сучасних табличних баз даних. Елементи носія табличної алгебри уточнюють реляційні структури даних, а сигнатурні операції побудовані на базі основних табличних маніпуляцій у реляційних алгебрах та SQL-подібних мовах.

Однією з актуальних задач у реляційних та табличних алгебрах є задача еквівалентного перетворення виразів з метою їх спрощення (мінімізації) або приведення до стандартного вигляду; ця задача є одним з етапів оптимізації запитів [3], розв'язання цієї задачі може суттєво зменшити час обробки інформації в реляційних системах управління базами даних. Для еквівалентного перетворення виразів, що містять табличні операції, потрібно використовувати взаємозв'язки між ними.

Значну кількість таких взаємозв'язків встановлено у монографії [4], у більшості випадків ці взаємозв'язки для загального випадку виконуються у вигляді включень. Автором були знайдені критерії переходу деяких з цих включень у рівності. Ці критерії виражені в термінах активних доменів таблиць та є природними. Важливість знайдених критеріїв для теорії табличних алгебр полягає в тому, що еквівалентні перетворення виразів, що містять табличні операції, можна здійснювати лише на основі рівностей. У цій роботі наведено всі знайдені на цей час взаємозв'язки перетину, об'єднання, різниці між собою та іншими операціями табличних алгебр: проекцією, селекцією, перейменуванням атрибутів, насиченням, активним доповненням та з'єднанням.

## Основні означення

Зафіксуємо деяку непорожню множину  $A = \{A'_1, \dots, A'_n\}$ , елементи якої будемо називати атрибутами. Довільну скінченну підмножину  $R = \{A_1, \dots, A_k\} \subseteq A$  назовемо схемою. Рядком  $s$  схеми  $R$  є множина пар  $s = \{(A_1, d_1), \dots, (A_k, d_k)\}$ , проекція якої за першою компонентою дорівнює  $R$ . Таблицею схеми  $R$  є скінченна множина рядків схеми  $R$ . Виокремлюють дві особливі таблиці: таблицю  $T_\varepsilon = \{\varepsilon\}$ , де  $\varepsilon$  – порожній рядок, при цьому схема таблиці  $T_\varepsilon$  є порожнім рядком, та таблицю  $T_\emptyset = \emptyset$  – порожня множина рядків довільної схеми.

На множині всіх таблиць схеми  $R$  введено такі операції:

- 1) перетин  $\bigcap_R$  таблиць схеми  $R$  – таблиця, що складається з тих і лише тих рядків, які належать одночасно всім вихідним таблицям;
- 2) об'єднання  $\bigcup_R$  таблиць схеми  $R$  – таблиця, що складається з тих і лише тих рядків, які належать хоча б одній з вихідних таблиць;
- 3) різниця  $T_1 - T_2$  двох таблиць схеми  $R$  – таблиця, що складається з тих і лише тих рядків, які належать таблиці  $T_1$ , та не належать таблиці  $T_2$ .

Іншими словами, операції перетину, об'єднання і різниці таблиць є обмеженням відповідно теоретико-множинних перетину, об'єднання і різниці на множині таблиць однакової схеми.

Проекцією за множиною атрибутів  $X \subseteq R$  називається унарна параметрична операція  $\pi_X$ , значенням якої є таблиця, що складається з обмежень за  $X$  усіх рядків вихідної таблиці:  $\pi_X(T) = \{s \parallel X \mid s \in T\}$  (тут обмеження розуміємо стандартно:  $s \parallel x = s \cap (X \times pr_2 s)$ , де  $pr_2 s$  – проекція рядка  $s$  за другою компонентою). Селекцією за предикатом  $P : S \rightarrow \{true, false\}$ , де  $S$  – множина всіх рядків, є унарна параме-

трична операція  $\sigma_P$ , яка таблиці співставляє її підтаблицю, що містить рядки, на яких предикат  $P$  приймає значення ІСТИНА. Операція перейменування  $Rn(T)_\xi$  зводиться до перейменування атрибутів таблиці  $T$  у відповідності до відображення-параметру  $\xi$ , тобто, перейменування таблиці є перейменуванням перших компонентів пар – елементів рядків.

Для визначення операції насичення введемо одне важливе поняття. Активним доменом атрибуту  $A$  відносно таблиці  $T$  називається множина  $D_{A,T} = \{d | \exists s \in T \wedge (A, d) \in s\}$ . Змістовно кажучи,  $D_{A,T}$  містить усі значення атрибуту  $A$  в таблиці  $T$ . Насиченням  $C(T)$  називається таблиця  $\prod_{A \in R} D_{A,T}$ , де  $R$  – схема таблиці  $T$ , а  $\prod$  – оператор прямого (декартового) добутку, що відповідає індексуванню  $A \mapsto D_{A,T}$ ,  $A \in R$ . Активним доповненням таблиці  $T$  схеми  $R$  називається таблиця  $\tilde{T} = C(T) -_R T$ .

Для визначення операції з'єднання введемо одне допоміжне поняття. Бінарні відношення  $\rho$  і  $\tau$  називаються сумісними (позначається  $\rho \approx \tau$ ), якщо  $\rho \parallel X = \tau \parallel X$ , де  $X = pr_1(\rho) \cap pr_1(\tau)$ . З'єднанням називається бінарна операція  $\otimes$ , значенням якої є таблиця, що складається з усіх можливих об'єднань сумісних рядків вихідних таблиць, тобто  $T_1 \otimes T_2 = \{s_1 \cup s_2 \mid s_1 \in T_1 \wedge s_2 \in T_2 \wedge s_1 \approx s_2\}$ .

Табличною алгеброю називають часткову алгебру з носієм – множиною всіх таблиць довільної схеми, наведеними вище операціями (насичення вважається допоміжною операцією), а також операцією ділення таблиць, яка може бути вираженою через проекцію, різницю та з'єднання; в цій роботі операція ділення не використовується. Далі, для компактності викладання результатів, покладемо, що всі зазначені таблиці мають непорожню схему  $R$  з кількістю атрибутів  $k > 0$ .

## Властивості перетину таблиць

Відомо, що перетин може бути вираженим через різницю таблиць:  $T_1 \cap_R T_2 = T_1 -_R (T_1 -_R T_2)$ . Також відомо, що перетин таблиць є комутативною, асоціативною та ідемпотентною операцією, тобто виконуються рівності  $T_1 \cap_R T_2 = T_2 \cap_R T_1$ ,  $T_1 \cap_R (T_2 \cap_R T_3) = (T_1 \cap_R T_2) \cap_R T_3$  та  $T \cap_R T = T$ . Крім того, відомо, що операції перетину та об'єднання таблиць є дистрибутивними між собою двома варіантами, тобто виконуються рівності  $T_1 \cup_R (\cap_i T_i) = \cap_i (T_1 \cup_R T_i)$  та  $T_1 \cap_R (\cup_i T_i) = \cup_i (T_1 \cap_R T_i)$ .

Розглянемо взаємозв'язки між перетином та різницею таблиць. Операція перетину таблиць є дистрибутивною відносно різниці.

**Лема 1.** Виконується рівність  $T_1 \bigcap_R (T_2 - T_3) = (T_1 \bigcap_R T_2) - (T_1 \bigcap_R T_3)$ .

Оскільки різниця таблиць (на відміну від їх перетину) не є комутативною операцією, можна розглянути два варіанти дистрибутивності між різницею та перетином таблиць.

У першому варіанті різниця не є дистрибутивною відносно перетину. Для загального випадку справедлива

**Лема 2.** Виконується включення  $T_1 - (T_2 \bigcap_R T_3) \supseteq (T_1 - T_2) \bigcap_R (T_1 - T_3)$ .

Знайдено критерій перетворення цього включення у рівність.

**Лема 3.** Рівність  $T_1 - (T_2 \bigcap_R T_3) = (T_1 - T_2) \bigcap_R (T_1 - T_3)$  виконується тоді та лише тоді, коли  $T_1 \bigcap_R T_2 = T_1 \bigcap_R T_3$ .

У другому варіанті різниця є дистрибутивною відносно перетину. Справедливе таке твердження.

**Лема 4.** Виконується рівність  $(T_1 \bigcap_R T_2) - T_3 = (T_1 - T_3) \bigcap_R (T_2 - T_3)$ .

Доведення леми 4 наведено у [6], доведення всіх інших тверджень цього підрозділу наведено у [5].

Розглянемо взаємозв'язки між перетином та проекцією. Операція проєкції не є дистрибутивною відносно перетину таблиць, проте виконується включення  $\pi_X(\bigcap_i T_i) \subseteq \bigcap_i \pi_X(T_i)$ , що доведено у [4]. Знайдено критерій, за яким це включення перетворюється у рівність.

**Теорема 1.** Рівність  $\pi_X(\bigcap_i T_i) = \bigcap_i \pi_X(T_i)$  виконується тоді та лише тоді, коли для кожного рядка  $s = \{(X_1, x_1), \dots, (X_p, x_p)\} \in \bigcap_i \pi_X(T_i)$  існують такі значення  $o_1 \in D_{O_1, \bigcap_i T_i}, \dots, o_{k-p} \in D_{O_{k-p}, \bigcap_i T_i}$ , що  $s' = \{(X_1, x_1), \dots, (X_p, x_p), (O_1, o_1), \dots, (O_{k-p}, o_{k-p})\} \in \bigcap_i T_i$ .

У [4] доведено, що селекція та перейменування є дистрибутивними відносно перетину таблиць:  $\sigma_P(\bigcap_i T_i) = \bigcap_i \sigma_P(T_i)$  та  $Rn(T_1 \bigcap_R T_2)_\xi = Rn(T_1)_\xi \bigcap_{\xi(R)} Rn(T_2)_\xi$ .

Розглянемо взаємозв'язки між перетином та насиченням. Операція насичення не є дистрибутивною відносно перетину таблиць, проте виконується включення  $C(T_1 \bigcap_R T_2) \subseteq C(T_1) \bigcap_R C(T_2)$ , що доведено у [4]. Знайдено критерій, за яким це включення перетворюється у рівність.

**Теорема 2.** Рівність  $C(T_1 \bigcap_R T_2) = C(T_1) \bigcap_R C(T_2)$  виконується при  $C(T_1) \bigcap_R C(T_2) = T_\emptyset$ . Якщо ж  $C(T_1) \bigcap_R C(T_2) \neq T_\emptyset$ , то ця рівність виконується, якщо виконується хоча б одна з умов:

1) виконуються рівності  $\forall i \ D_{A_i, T_1 \cap_R T_2} = D_{A_i, T_1} \cap D_{A_i, T_2}$ ;

2) для будь-якого  $i$  та кожного елемента  $x$  з множини  $D_{A_i, T_1} \cap D_{A_i, T_2}$  існує такий рядок  $s \in T_1 \cap_R T_2$ , що  $(A_i, x) \in s$ .

У якості взаємозв'язків між перетином та активним доповненням можна розглянути розповсюдження відомих з теорії множин та теорії булевих функцій законів де Моргана на табличні операції (тут можна говорити про взаємозв'язки між перетином, об'єднанням та активним доповненням).

У [4] доведено включення  $\tilde{T}_1 \cap_R \tilde{T}_2 \subseteq (T_1 \cup_R T_2)$  та  $(T_1 \cap_R T_2) \subseteq \tilde{T}_1 \cup_R \tilde{T}_2$ . Для кожного з цих включень знайдено критерії переходу їх у рівності.

**Теорема 3.** Рівність  $\tilde{T}_1 \cap_R \tilde{T}_2 = (T_1 \cup_R T_2)$  виконується при  $(T_1 \cup_R T_2) = T_\emptyset$ .

Якщо ж  $(T_1 \cup_R T_2) \neq T_\emptyset$ , то ця рівність виконується тоді та лише тоді, коли виконується хоча б одна з чотирьох умов, що виключають одна іншу:

1)  $\forall A (A \in R \Rightarrow D_{A, T_1} = D_{A, T_2})$ ;

2)  $\forall A (A \in R \Rightarrow D_{A, T_2} \subseteq D_{A, T_1})$ , та для всіх індексів  $q$ , значень  $x \in D_{A_q, T_1} - D_{A_q, T_2}$  та всіх значень  $d_1, \dots, d_{q-1}, d_{q+1}, \dots, d_k$ , що належать відповідно активним доменам  $D_{A_1, T_1}, \dots, D_{A_{q-1}, T_1}, D_{A_{q+1}, T_1}, \dots, D_{A_k, T_1}$ , рядок  $\{(A_1, d_1), \dots, (A_{q-1}, d_{q-1}), (A_q, x), (A_{q+1}, d_{q+1}), \dots, (A_k, d_k)\} \in T_1$ ;

3)  $\forall A (A \in R \Rightarrow D_{A, T_1} \subseteq D_{A, T_2})$ , та для всіх індексів  $q$ , значень  $x \in D_{A_q, T_2} - D_{A_q, T_1}$  та всіх значень  $d_1, \dots, d_{q-1}, d_{q+1}, \dots, d_k$ , що належать відповідно активним доменам  $D_{A_1, T_2}, \dots, D_{A_{q-1}, T_2}, D_{A_{q+1}, T_2}, \dots, D_{A_k, T_2}$ , рядок  $\{(A_1, d_1), \dots, (A_{q-1}, d_{q-1}), (A_q, x), (A_{q+1}, d_{q+1}), \dots, (A_k, d_k)\} \in T_2$ ;

4) Існує такий атрибут  $A_q$ , для якого існують значення  $x \in D_{A_q, T_1} - D_{A_q, T_2}$ ,  $y \in D_{A_q, T_2} - D_{A_q, T_1}$ , причому  $D_{A_q, T_1} \cap D_{A_q, T_2} \neq \emptyset$ ; крім того, для всіх  $i \neq q$  виконуються рівності  $D_{A_i, T_1} = D_{A_i, T_2}$ ; а також, для всіх  $z_1 \in D_{A_q, T_1} - D_{A_q, T_2}$ , всіх  $z_2 \in D_{A_q, T_2} - D_{A_q, T_1}$  та всіх  $d_1, \dots, d_{q-1}, d_{q+1}, \dots, d_k$ , що належать відповідно активним доменам  $D_{A_1, T_1}, \dots, D_{A_{q-1}, T_1}, D_{A_{q+1}, T_1}, \dots, D_{A_k, T_1}$ , рядок  $\{(A_1, d_1), \dots, (A_{q-1}, d_{q-1}), (A_q, z_1), (A_{q+1}, d_{q+1}), \dots, (A_k, d_k)\} \in T_1$ , а рядок  $\{(A_1, d_1), \dots, (A_{q-1}, d_{q-1}), (A_q, z_2), (A_{q+1}, d_{q+1}), \dots, (A_k, d_k)\} \in T_2$ .

**Теорема 4.** Рівність  $(T_1 \cap_R T_2) = \tilde{T}_1 \cup_R \tilde{T}_2$  виконується тоді та лише тоді, коли виконується хоча б одна з двох умов, що виключають одна іншу:

1) для кожного атрибута  $A \in R$  виконуються рівності  $D_{A, T_1} = D_{A, T_1 \cap_R T_2}$  та  $D_{A, T_2} = D_{A, T_1 \cap_R T_2}$ ;

2) для кожного атрибута  $A_q \in R$  та кожного такого значення  $x$ , що  $x \in D_{A_q, T_1} - D_{A_q, T_1 \cap_R T_2}$  та всіх зна-

чень  $d_1, \dots, d_{q-1}, d_{q+1}, \dots, d_k$ , що належать відповідно активним доменам  $D_{A_1, T_1}, \dots, D_{A_{q-1}, T_1}, D_{A_{q+1}, T_1}, \dots, D_{A_k, T_1}$ , рядок  $\{(A_1, d_1), \dots, (A_{q-1}, d_{q-1}), (A_q, x), (A_{q+1}, d_{q+1}), \dots, (A_k, d_k)\} \in T_1$ . Для кожного атрибута  $A_w \in R$  та кожного такого значення  $y$ , що  $y \in D_{A_w, T_2} - D_{A_w, T_1 \cap_R T_2}$ , та всіх значень  $d_1, \dots, d_{w-1}, d_{w+1}, \dots, d_k$ , що належать відповідно активним доменам  $D_{A_1, T_2}, \dots, D_{A_{w-1}, T_2}, D_{A_{w+1}, T_2}, \dots, D_{A_k, T_2}$ , рядок  $\{(A_1, d_1), \dots, (A_{w-1}, d_{w-1}), (A_w, y), (A_{w+1}, d_{w+1}), \dots, (A_k, d_k)\} \in T_2$ .

Розглянемо взаємозв'язки між перетином та з'єднанням. Оскільки обидві ці операції є бінарними, то можна розглядати два варіанти дистрибутивності між ними. Оскільки перетином є частковою операцією, яка визначена лише у випадку, коли схеми обох таблиць-аргументів є однаковими, а схема таблиці-з'єднання є об'єднанням схем вихідних таблиць, при дослідженні можливої дистрибутивності існують певні обмеження на схеми вихідних таблиць. У кожного з двох наведених нижче виразів ліва та права частина одночасно будуть визначеними лише за умови рівності схем вихідних таблиць. У цьому випадку кожна з цих операцій є дистрибутивною відносно іншої:  $T \otimes (\bigcap_i T_i) = \bigcap_i (T \otimes T_i)$  та  $T \bigcap_R (\bigotimes_i T_i) = \bigotimes_i (T \bigcap_R T_i)$ , що доведено у [4].

## Властивості об'єднання таблиць

Відомо, що об'єднання таблиць є комутативною, асоціативною та ідемпотентною операцією, тобто виконуються рівності  $T_1 \bigcup_R T_2 = T_2 \bigcup_R T_1$ ,  $T_1 \bigcup_R (T_2 \bigcup_R T_3) = (T_1 \bigcup_R T_2) \bigcup_R T_3$  та  $T \bigcup_R T = T$ .

Розглянемо взаємозв'язки між об'єднанням та різницею таблиць. Операція об'єднання таблиць не є дистрибутивною відносно різниці. Для загального випадку справедлива

**Лема 5.** Виконується включення  $T_1 \bigcup_R (T_2 - T_3) \supseteq (T_1 \bigcup_R T_2) - (T_1 \bigcup_R T_3)$ .

Знайдено критерій перетворення цього включення у рівність.

**Лема 6.** Рівність  $T_1 \bigcup_R (T_2 - T_3) = (T_1 \bigcup_R T_2) - (T_1 \bigcup_R T_3)$  виконується тоді та лише тоді, коли  $T_1 = T_\emptyset$ .

Оскільки різниця таблиць (на відміну від їх об'єднання) не є комутативною операцією, можна розглянути два варіанти дистрибутивності між різницею та об'єднанням таблиць.

У першому варіанті різниця не є дистрибутивною відносно

об'єднання. Для загального випадку справедлива

**Лема 7.** Виконується включення  $T_1 \underset{R}{-} (T_2 \underset{R}{\cup} T_3) \subseteq (T_1 \underset{R}{-} T_2) \underset{R}{\cup} (T_1 \underset{R}{-} T_3)$ .

Знайдено критерій перетворення цього включення у рівність.

**Лема 8.** Рівність  $T_1 \underset{R}{-} (T_2 \underset{R}{\cup} T_3) = (T_1 \underset{R}{-} T_2) \underset{R}{\cup} (T_1 \underset{R}{-} T_3)$  виконується тоді та лише тоді, коли  $T_1 \underset{R}{\cap} T_2 = T_1 \underset{R}{\cap} T_3$ .

У другому варіанті різниця є дистрибутивною відносно об'єднання.

**Лема 9.** Виконується рівність  $(T_1 \underset{R}{\cap} T_2) \underset{R}{-} T_3 = (T_1 \underset{R}{-} T_3) \underset{R}{\cap} (T_2 \underset{R}{-} T_3)$ .

Доведення леми 9 наведено у [6], доведення всіх інших тверджень цього підрозділу наведено у [5].

У [4] доведено дистрибутивність проєкції, селекції та перейменування відносно об'єднання таблиць:  $\pi_X(\bigcup_i T_i) = \bigcup_i \pi_X(T_i)$ ;  $\sigma_P(\bigcup_i T_i) = \bigcup_i \sigma_P(T_i)$  та  $Rn(T_1 \underset{R}{\cup} T_2)_\xi = Rn(T_1)_\xi \cup Rn(T_2)_\xi$ .

Розглянемо взаємозв'язки між об'єднанням та насиченням таблиць. У [4] доведено включення  $C(T_1 \underset{R}{\cup} T_2) \supseteq C(T_1) \underset{R}{\cup} C(T_2)$ . Знайдено критерії переходу цього включення в рівність.

**Теорема 5.** Рівність  $C(T_1 \underset{R}{\cup} T_2) = C(T_1) \underset{R}{\cup} C(T_2)$  виконується тоді та лише тоді, коли виконується хоча б одна з умов:

- 1) існує не більше одного атрибута, для якого значення активного домену відносно таблиць  $T_1$  та  $T_2$  відрізняються;
- 2) виконується хоча б одне з включень  $\forall i \ D_{A_i, T_1} \subseteq D_{A_i, T_2}$  або  $\forall i \ D_{A_i, T_2} \subseteq D_{A_i, T_1}$ .

Розглянемо взаємозв'язки між об'єднанням та з'єднанням. Оскільки обидві ці операції є бінарними, то можна розглядати два варіанти дистрибутивності між ними. так саме, як і для перетину, одночасна визначеність лівих та правих частин кожного з двох наведених нижче виразів можлива лише за випадку рівності схем вихідних таблиць. У цьому випадку кожна з цих операцій є дистрибутивною відносно іншої:  $T \otimes (\bigcup_i T_i) = \bigcup_i (T \otimes T_i)$  та  $T \underset{R}{\cup} (\otimes_i T_i) = \otimes_i (T \underset{R}{\cup} T_i)$ , ці рівності доведено у [4].

## Властивості різниці таблиць

Доведення всіх тверджень цього підрозділу наведено у [6].

Неважко бачити, що різниця не є ані комутативною, ані асоціативною операцією, проте справедливе наступне твердження.

**Лема 10.** Виконується включення  $(T_1 - T_2) - T_3 \subseteq T_1 - (T_2 - T_3)$ . Це включення перетворюється у рівність тоді і лише тоді, коли  $T_1 \bigcap_R T_3 = T_\emptyset$ .

Виконується включення  $\pi_X(T_1) - \pi_X(T_2) \subseteq \pi_X(T_1 - T_2)$ , що доведено у [4]. Знайдено необхідні та достатні умови, за яких це включення перетворюється у рівність. Далі, у теоремі 6 покладемо  $T_1, T_2$  – таблиці схеми  $R$ ,  $X = \{X_1, \dots, X_m\}$ ; а елементи множини  $R - X$  позначимо  $O_1, \dots, O_p$ .

**Теорема 6.** Рівність  $\pi_X(T_1) - \pi_X(T_2) = \pi_X(T_1 - T_2)$  виконується тоді та лише тоді, коли для будь-якого рядка  $s = \{(X_1, x_1), \dots, (X_m, x_m)\} \in \pi_X(T_1) \bigcap \pi_X(T_2)$  та всіх значень  $o_1 \in D_{O_1, T_1 \bigcup_R T_2}, \dots, o_p \in D_{O_p, T_1 \bigcup_R T_2}$ , з  $s' = \{(X_1, x_1), \dots, (X_m, x_m), (O_1, o_1), \dots, (O_p, o_p)\} \in T_1$  випливає  $s' \in T_2$ .

У [4] доведено дистрибутивність селекції та перейменування відносно різниці:  $\sigma_P(T_1 - T_2) = \sigma_P(T_1) - \sigma_P(T_2) = \sigma_P(T_1) - T_2$  та  $Rn(T_1 - T_2)_\xi = Rn(T_1)_\xi - Rn(T_2)_\xi$ .

Знайдемо взаємозв'язки між різницею та насиченням. Неважко побачити, що насичення не є дистрибутивним відносно різниці, більш того, для загального випадку  $C(T_1 - T_2) \not\subseteq C(T_1) - C(T_2)$  та  $C(T_1 - T_2) \not\supseteq C(T_1) - C(T_2)$ . Знайдемо критерії виконання включення  $C(T_1 - T_2) \subseteq C(T_1) - C(T_2)$ .

**Лема 11.** Включення  $C(T_1 - T_2) \subseteq C(T_1) - C(T_2)$  виконується тоді та лише тоді, коли існує такий індекс  $i$ , що  $D_{A_i, T_1 - T_2} \bigcap D_{A_i, T_2} = \emptyset$ .

Знайдемо критерії виконання включення  $C(T_1) - C(T_2) \subseteq C(T_1 - T_2)$ .

**Лема 12.** Включення  $C(T_1) - C(T_2) \subseteq C(T_1 - T_2)$  виконується тоді та лише тоді, коли виконується одна з двох взаємовиключних умов:

- 1)  $\forall i D_{A_i, T_1 - T_2} = D_{A_i, T_1}$ ;
- 2)  $\forall i (D_{A_i, T_1 - T_2} \neq D_{A_i, T_1} \rightarrow (\forall j \neq i \ D_{A_j, T_1} \subseteq D_{A_j, T_2}))$ .

Об'єднаємо результати, що одержано у лемах 11 та 12, у наступному твердженні.

**Теорема 7.** Рівність  $C(T_1) - C(T_2) = C(T_1 - T_2)$  виконується тоді та лише тоді, коли існує такий індекс  $i$ , що  $D_{A_i, T_1 - T_2} \bigcap D_{A_i, T_2} = \emptyset$ , та, крім того, виконується одна з двох взаємовиключних умов:

- 1)  $\forall i D_{A_i, T_1 - T_2} = D_{A_i, T_1}$ ;

$$2) \forall i (D_{A_i, T_1 - T_2} \neq D_{A_i, T_1} \rightarrow (\forall j \neq i \quad D_{A_j, T_1} \subseteq D_{A_j, T_2})).$$

У якості взаємозв'язків між різницею та активним доповненням можна розглянути розповсюдження відомої з теорії множин рівності  $A - B = A \wedge \bar{B}$  на табличні операції. У [4] доведено включення  $T_1 - T_2 \supseteq T_1 \bigcap_R \tilde{T}_2$ . Знайдемо критерії перетворення цього включення у рівність.

**Лема 13.** Рівність  $T_1 - T_2 = T_1 \bigcap_R \tilde{T}_2$  виконується тоді та лише тоді, коли для будь-якого  $A \in R$  виконується включення  $D_{A, T_1} \subseteq D_{A, T_2}$ .

Розглянемо взаємозв'язки між різницею та з'єднанням. Оскільки різниця є частковою операцією, яка визначена лише у випадку, коли схеми обох таблиць-аргументів є однаковими, а схема таблиці-з'єднання є об'єднанням схем вихідних таблиць, при дослідженні можливої дистрибутивності існують певні обмеження на схеми вихідних таблиць. У [4] доведено дистрибутивність з'єднання відносно різниці: нехай  $R_1$  – схема таблиці  $T_1$ ,  $R$  – схема таблиці  $T_2$ . Тоді, якщо  $R$  – схема таблиці  $T_3$ , виконується рівність:  $T_1 \otimes (T_2 - T_3) = (T_1 \otimes T_2) -_{R \cup R_1} (T_1 \otimes T_3)$ . Зауважимо, що у випадку, за яким схеми таблиць  $T_2$  та  $T_3$  не збігаються, ліва та права частини наведеної вище рівності не можуть одночасно бути визначеними, оскільки ліва частина обов'язково буде невизначеною.

Для дослідження можливої дистрибутивності різниці відносно з'єднання (зважаючи на некомутативність різниці будемо розглядати два варіанти такої дистрибутивності), так саме, як і для перетину та об'єднання, одночасна визначеність лівих та правих частин кожного з наведених нижче виразів можлива лише за випадку рівності схем вихідних таблиць. Справедливе твердження.

**Лема 14.** Нехай  $R$  – схема таблиць  $T_1, T_2, T_3$ . Тоді:

- а) виконується рівність  $(T_1 \otimes T_2) -_{R} T_3 = (T_1 -_{R} T_3) \otimes (T_2 -_{R} T_3)$ ;
- б) виконується включення  $T_1 -_{R} (T_2 \otimes T_3) \supseteq (T_1 -_{R} T_2) \otimes (T_1 -_{R} T_3)$ ;
- в) рівність  $T_1 -_{R} (T_2 \otimes T_3) = (T_1 -_{R} T_2) \otimes (T_1 -_{R} T_3)$  виконується тоді та лише тоді, коли  $T_1 \bigcap_R T_2 = T_1 \bigcap_R T_3$ .

## Висновки

Таким чином, у роботі розглянуто та систематизовано взаємозв'язки перетину, об'єднання та різниці таблиць між собою та з іншими сигнатурними операціями табличних алгебр: та проекцією, селекцією, перейме-

нуванням, насиченням, активним доповненням та з'єднанням. Знайдено критерії, за яких деякі взаємозв'язки, що у загальному випадку є включеннями, стають рівностями. Ці результати можуть бути використаними для перетворення виразів, що містять табличні операції, задля їх мінімізації або переходу до стандартного вигляду. У подальшому планується розглянути взаємозв'язки між іншими табличними операціями, в першу чергу операції ділення з рештою сигнатурних табличних операцій, а також знаходження ймовірностей виконання знайдених у цій роботі критеріїв для довільних таблиць та таблиць із певними фіксованими параметрами.

## Список використаних джерел

- [1] Codd E.F. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*. 1970. Vol. 13, No. 6. P. 377–387.
- [2] Red'ko V.N., Bui D.B. Foundations of the theory of relational database models. *Cybernetics and Systems Analysis*. 1996. Vol. 32, Issue 4. P. 471–478.
- [3] Garcia-Molina H., Ullman J.D., Widom J. Database Systems. The Complete Book, 2th ed. Pearson Prentice Hall, 2009.
- [4] Редько В.Н., Брона Ю.Й., Буй Д.Б., Поляков С.А. Реляційні бази даних: табличні алгебри та SQL-подібні мови. Київ: Академперіодика, 2001.
- [5] Сенченко А.С. Взаимосвязи между пересечением, объединением и другими сигнатурными операциями в табличных алгебрах. *Праці Інституту прикладної математики і механіки НАН України*. 2018. Т. 32. С. 121–132.
- [6] Сенченко О.С. Взаємозв'язки різниці та проекції з іншими сигнатурними операціями табличних алгебр. *Праці Інституту прикладної математики і механіки НАН України*. 2021. Т. 35, №2.

Сенченко О.С., Серeda О.А., Притула М.І.

<sup>1</sup> к. ф-м. н., старший науковий співробітник відділу теорії керуючих систем Інституту прикладної математики і механіки НАНУ, Слов'янськ

<sup>2</sup> аспірант відділу теорії керуючих систем Інституту прикладної математики і механіки НАНУ, Слов'янськ

<sup>3</sup> аспірант відділу теорії керуючих систем Інституту прикладної математики і механіки НАНУ, Слов'янськ

e-mail: <sup>1</sup> senchenko.a76@gmail.com, <sup>2</sup> 0955174042@ukr.net, <sup>3</sup> elanir358@gmail.com

## ВИЗНАЧАЛЬНА ПАРА ДЛЯ ДЕТЕРМІНОВАНИХ ГРАФІВ

В роботі розглядається задача знаходження ефективного лінгвістичного представлення детермінованих графів, у яких в околі кожної вершини всі інші вершини мають попарно різні мітки. Запропоновано представлення таких графів двома множинами, що утворюють визначальну пару: перша компонента цієї пари описує цикли графа, друга – його висячі вершини. Розглянуто структуру мінімальної за кількістю слів так званої канонічної визначальної пари, наведено алгоритм її побудови, знайдено її властивості та чисельні характеристики.

**Ключові слова:** *графи з розміченими вершинами, визначальна пара.*

## Вступ

Теорія графів є важливим розділом дискретної математики, який надає потужний апарат для розв'язання задач різного ступеня складності в найрізноманітніших галузях наукової та практичної діяльності. Широке застосування графів пов'язане з тим, що вони є природним засобом пояснення складних ситуацій на інтуїтивному рівні. Ці переваги зображення складних структур і процесів у вигляді графів стають ще більш відчутними за наявності оптимальних засобів їх представлення. Одним із сучасних напрямків теорії графів є дослідження розмічених (зважених) графів, тобто графів, елементи яких – вершини, дуги, інцидентори тощо – мають мітки з попередньо визначеного алфавіту. Такі графи широко використовуються в інформатиці та кібернетиці для описання та моделювання різноманітних обчислювальних процесів. Серед розмічених графів найбільш дослідженими є скінченні орієнтовані графи з розміченими ребрами (LTS, зважені автомати, скінченні автомати тощо). Також існує багато обчислювальних процесів (наприклад, у програмуванні, робототехніці, верифікації моделей) [1, 2], які можна природньо представити графами з розміченими вершинами.

У роботі [3] запропоновано представлення скінченних, всюди визначених автоматів без виходів системами визначальних співвідношень – множинами пар слів спеціального вигляду, що задають еквівалентні траєкторії в автоматі. За допомогою такого представлення було розв'язано задачу характеристики – визначення, чи є дана множина пар слів системою визначальних співвідношень для заданого автомата без безпосередньої побудови самого автомата. На нашу думку, природним є розповсюдження такого представлення на деякий підклас розмічених графів. Доцільність такого розповсюдження впливає з природнього взаємозв'язку між розміченими графами та мовами в алфавіті міток вершин графів: траєкторія переміщень у графі може бути задана послідовністю міток вершин, а аналіз таких переміщень може дати важливу інформацію про структуру графа. Цей взаємозв'язок може дозволити застосувати методи теорії автоматів до аналізу розмічених графів.

Найбільш близькими до класу автоматів без виходів є так звані детерміновані графи – клас графів з розміченими вершинами, у яких в околі кожної вершини всі вершини мають попарно різні мітки [4]. Графи з такою розміткою мають низку важливих застосувань. У детермінованих графах траєкторія руху однозначно відновлюється за послідовністю міток вершин, тому у якості об'єкта розповсюдження понять та апарату систем визначальних співвідношень було обрано саме клас детермінованих графів. Для такого розповсюдження у якості основної було обрано задачу лінгвістичного представлення детермінованого графа: співставленню графу однієї чи декілька скінченних множин слів у алфавіті міток вершин графа, що описують можливі (та/або неможливі) траєкторії руху в графі. Для такого представлення треба визначитися з кількістю таких множин слів (бажано, щоб така кількість була якомога меншою) та роллю, яку відіграє кожна з цих множин слів у графі (які структурні елементи графа описує кожна множина). При цьому, ці множини треба обрати у такий спосіб, щоб можна було винайти універсальний алгоритм, який кожному набору цих множин або однозначно співставляє детермінований граф, або сповіщає, що це зробити неможливо. Також треба винайти алгоритм переходу від будь-якого детермінованого графа до сказаного вище набору множин.

## Означення

Розглядаються неорієнтовані, скінченні, непорожні, зв'язні, звичайні (такі, що не містять петлі та кратні ребра) детерміновані графи з розмічени-

ми вершинами  $G = (V, E, X, \xi(V))$ , де  $V$  – множина вершин графа,  $E$  – множина його ребер,  $\xi(V) : V \rightarrow X$  – всюди визначена функція розмітки вершин графа символами скінченного алфавіту  $X = \{x_1, \dots, x_p\}$ , у яких для будь-яких вершин  $v, v', v'' \in V$  з  $(v, v'), (v, v'') \in E$  та  $\xi(v') = \xi(v'')$  випливає рівність  $v' = v''$  (далі Д-графи). Змістовно кажучи, наведене означає, що у Д-графів будь-які дві вершини, що одночасно суміжні деякій вершині, мають різні мітки. Множину всіх слів скінченної довжини в алфавіті  $X$  будемо позначати через  $X^*$ . Д-графи є дуже зручними для дослідження їх структури агентами – графохідними автоматами, оскільки, якщо спостерігач має карту графа (множини  $V, E, X$  та функцію  $\xi(V)$ ) та знає, на яку вершину досліджуваного Д-графа був поміщений агент на початку дослідження (тобто, ініціальну вершину), то за послідовністю міток вершин, у яких перебував агент, може бути однозначно відновлена траєкторія переміщення агента по графу. У випадку, коли спостерігачу не відома карта досліджуваного графа, переміщення агентів можуть бути організовані в такий спосіб, щоб на основі їх аналізу спостерігач одержав шукану інформацію про структуру графа (наприклад, складення карти графа, пошуку у ньому найкоротших шляхів, порівняння досліджуваного графа з графом-еталоном).

Нехай  $G = (V, E, X, \xi(V))$  – довільний розмічений граф. Детермінізацією  $G$  назвемо процедуру, яка продовжується до тих пір, поки в графі існують такі різні вершини  $v, v', v'' \in V$ , що одночасно виконуються умови  $(v, v'), (v, v'') \in E$  та  $\xi(v') = \xi(v'')$ , і складається з ототожнення вершин  $v'$  та  $v''$ , заміни кратних ребер, що виникають в результаті цього ототожнення, одним ребром та вилучення ребра  $(v', v'')$  (якщо воно існувало в  $G$ ). Зі скінченності вершин у графі випливає скінченність його детермінізації.

Зафіксуємо деяку вершину  $v_0 \in V$  Д-графа  $G = (V, E, X, \xi(V))$ , яку далі будемо називати ініціальною, цю вершину за необхідності ми будемо виділяти у позначенні Д-графа:  $G = (V, E, X, \xi(V), v_0)$ . Ізоморфізмом Д-графів  $G_1 = (V_1, E_1, X, \xi_1(V_1), v')$  і  $G_2 = (V_2, E_2, X, \xi_2(V_2), v'')$  ( $G_1 \cong G_2$ ) з однаковою множиною міток вершин  $X$  називаємо взаємно однозначну відповідність  $\varphi : V_1 \leftrightarrow V_2$  між множинами їх вершин, що зберігає їх ініціальні вершини, відношення суміжності вершин та функцію розмітки вершин:  $\varphi(v') = v'', \forall v', v'' \in V_1 \quad ((v', v'') \in E_1 \leftrightarrow (\varphi(v'), \varphi(v'')) \in E_2)$  та  $\forall v \in V_1 \quad \xi_1(v) = \xi_2(\varphi(v))$ .

Оскільки шляху  $p = v'_0 \dots v'_k$  однозначно відповідає слово в алфавіті міток  $\xi(p) = \xi(v'_1) \xi(v'_2) \dots \xi(v'_k)$ , то надалі шлях  $p$  розглядатимемо як слово  $\xi(p)$  та позначатимемо рівністю  $v'_1 p = v'_k$ . Шлях  $p = x'_1 x'_2 \dots x'_k$

назвемо припустимим для вершини  $v'_1$ , якщо  $\xi(v'_1) = x'_1$ , та існують такі вершини  $v'_2, \dots, v'_k \in V$ , що  $\xi(v'_2) = x'_2, \dots, \xi(v'_k) = x'_k$ , і  $(v'_1, v'_2), \dots, (v'_{k-1}, v'_k) \in E$ . Слово  $x'_k \dots x'_1$  будемо називати реверсом слова  $p = x'_1 \dots x'_k$  та позначати його через  $p^{-1}$ . Слово  $q = x''_1 \dots x''_m$  назвемо початковим відрізком слова  $p = x'_1 \dots x'_k$  у випадку, якщо  $m \leq k$  та виконуються рівності  $x''_1 = x'_1, \dots, x''_m = x'_m$ , цей факт позначатимемо через  $q \subseteq p$ .

## Визначальна пара для детермінованих графів

Всі вершини заданого Д-графа  $G = (V, E, X, \xi(V), v_0)$  можна розбити на два класи за такою процедурою: видалимо з  $G$  усі висячі вершини, відмінні від ініціальної, разом із ребрами, що є інцидентними цим вершинам. Цю процедуру будемо повторювати до тих пір, поки це можливо. Одержаний у такий спосіб граф  $B(G)$  назвемо базою графа  $G$ , вершини, що входять до  $B(G)$ , назвемо базовими, а ті вершини  $G$ , що не входять до  $B(G)$ , – вільними. Неважко бачити, що база графа будується однозначно за скінченну кількість кроків, база  $B(G)$  може складатися з однієї ініціальної вершини (у випадку, коли  $G$  є деревом), а у випадку, коли  $B(G)$  містить хоча б одну вершину, яка відмінна від ініціальної, то вона повинна містити хоча б один простий цикл. Звідси випливає, що кожна базова вершина, яка є відмінною від ініціальної вершини  $v_0$ , суміжна щонайменше двом базовим вершинам. Вільну вершину  $v$  назвемо прямим предком вільної вершини  $v'$ , якщо для будь-якого шляху  $p'$ , такого, що  $v_0 p' = v'$ , виконується  $v_0 p = v$  для деякого початкового відрізка  $p \subseteq p'$ .

Далі вважаємо, що всі шляхи в графі починаються з ініціальної вершини. Будемо робити представлення Д-графа  $G = (V, E, X, \xi(V), v_0)$  парою  $\{C, L\}(x')$  скінчених множин слів  $C, L \in X^*$ , для якої всі слова з  $C$  та  $L$  є припустимими для  $v_0$ , слова множини  $C$  описують цикли графа  $G$ , а слова  $L$  описують його висячі вершини. Далі в роботі під терміном «пара»  $\{C, L\}(x')$  розуміємо саме такі дві скінченні множини  $C$  та  $L$ , що всі слова множини  $C$  починаються та закінчуються символом  $x'$ , а всі слова множини  $L$  починаються з цього символу. При цьому множини  $C$  та  $L$  будемо називати компонентами пари  $\{C, L\}(x')$ .

У [5, 6] визначено процедуру, яка за заданою парою  $\{C, L\}(x')$  або будує Д-граф  $G(\{C, L\}(x'))$ , або показує, що за цією парою Д-граф побудувати неможливо.

Покладемо, що спочатку граф  $G(\{C, L\}(x'))$  складається з однієї вершини  $v_0$  з міткою  $\xi(v_0) = x'$ .

*Етап 1.* Для кожного слова  $p_i = x'_1 x^i_1 \dots x^i_{n-1} x' \in C$  додаємо  $n - 1$  вершину  $v^i_1, \dots, v^i_{n-1}$  з мітками відповідно  $x^i_1, \dots, x^i_{n-1}$  та  $n$  ребер  $(v_0, v^i_1), (v^i_1, v^i_2), \dots, (v^i_{n-2}, v^i_{n-1}), (v^i_{n-1}, v_0)$ . Після кожного такого додавання детермінізуємо одержаний граф.

*Етап 2.* У графі, який ми одержали після виконання етапу 1, можуть існувати висячі вершини. Кожну таку вершину  $v$ , що є відмінною від ініціальної вершини  $v_0$ , поміщаємо для подальшого аналізу у множину  $Q$ , яка на початку була порожня.

*Етап 3.* Для кожного слова  $p_j = x'_1 x^j_1 \dots x^j_{n-1} \in L$  виконуємо таку послідовність дій: додаємо вершини  $v^j_1, \dots, v^j_{n-1}$  з мітками відповідно  $x^j_1, \dots, x^j_{n-1}$ , ребра  $(v_0, v^j_1), \dots, (v^j_{n-2}, v^j_{n-1})$  та детермінізуємо одержаний граф. Якщо в результаті детермінізації вершина  $v_0 p_j$  не є висячею, то вважаємо, що граф  $G(\{C, L\}(x'))$  не існує, і процедура завершується. Якщо ж в результаті детермінізації з'явилась висяча вершина  $v$ , відмінна від вершини  $v_0 p_j$ , то цю вершину поміщаємо для подальшого аналізу у множину  $Q$ . Після цього переходимо до наступного слова компоненти  $L$ .

*Етап 4.* Для кожної вершини  $v \in Q$  розглядаємо всі слова компоненти  $L$ : якщо не існує такого слова  $p \in L$ , що вершина  $v$  є прямим предком вершини  $v_0 p$ , то вважаємо, що граф  $G(\{C, L\}(x'))$  не існує.

*Етап 5.* Розглядаємо всі слова множини  $L$ : якщо існує таке  $p \in L$ , що вершина  $v_0 p$  не є висячею, то вважаємо, що граф  $G(\{C, L\}(x'))$  не існує.

Якщо в результаті виконання цієї процедури за парою  $\{C, L\}(x')$  можливо побудувати граф  $G(\{C, L\}(x'))$ , то таку пару назвемо правильною. Правильну пару  $\{C, L\}(x')$  назвемо визначальною для Д-графа  $G$ , якщо  $G(\{C, L\}(x')) \cong G$ . Зауважимо, що таке завдання Д-графа визначальною парою в деякій мірі (етапами 1 та 3) є аналогічним представленням автоматів системою визначальних співвідношень, запропонованому у [3].

Доцільність введення другого, четвертого та п'ятого етапів процедури побудови Д-графа за парою обумовлено намаганням чітко розділити функції компонентів пари: передбачається, що слова компоненти  $C$  описують базові вершини графа, а слова компоненти  $L$  – його вільні вершини, причому кожне слово  $p \in L$  описує висячу вершину  $v_0 p$ , а для кожної вільної висячої вершини  $v$  потрібне існувати хоча б одне таке слово  $p \in L$ , що  $v_0 p = v$  (зауважимо, що у випадку, коли вершина  $v_0$  є висячею, то її не потрібно описувати у компоненті  $L$ ). У випадку, коли слова множини  $C$  задають деякі вільні вершини графа (тобто після виконання етапу 2 множина  $Q$  не є порожньою), ці вільні вершини обов'язково повинні бути

описаними словами компоненти  $L$  (див. етап 4), а якщо такого опису нема, то вважаємо, що за парою  $\{C, L\}(x')$  граф  $G(\{C, L\}(x'))$  побудувати неможливо.

Розглянемо дію окремих етапів наведеної вище процедури на такому прикладі.

**Приклад 1.** Нехай  $C = \{134321, 1245231\}$ ,  $L = \{124143, 12342, 132541, 12435\}(1)$ . На рис. 1 а зображено граф, який одержуємо після виконання етапів 1 та 2. Вершина  $v'$ , що виділена, є висячею та відмінною від ініціальної вершини  $v_0$ , тому цю вершину поміщаємо до множини  $Q$ . На рис. 1 б зображено граф, який одержуємо після виконання етапу 3. Після розгляду слова 124143 вершина  $v''$ , що виділена, є висячею та відмінною від вершини  $v_0$ , тому цю вершину поміщаємо до множини  $Q$ . Далі робимо аналіз вершин  $v'$  та  $v''$ , що входять до множини  $Q$ . Оскільки вершина  $v'$  є прямим предком вершини  $u'$ ,  $12342 \in L$  та  $v_012342 = u'$ , а також вершина  $v''$  є прямим предком самої себе,  $132541 \in L$  та  $v_0132541 = v''$  (рис. 1 в), то переходимо до виконання етапу 5. Оскільки вершина  $v$  (рис. 1 г) не є висячею,  $v_0124143 = v$  та  $124143 \in L$ , то, за умовою етапу 5, граф  $G(\{C, L\}(1))$  не існує.

Зауважимо, що для пари  $\{C, L'\}(1)$ , де  $L' = L - \{124143\}$  граф  $G(\{C, L'\}(1))$  існує, він зображений на рис. 1 г.

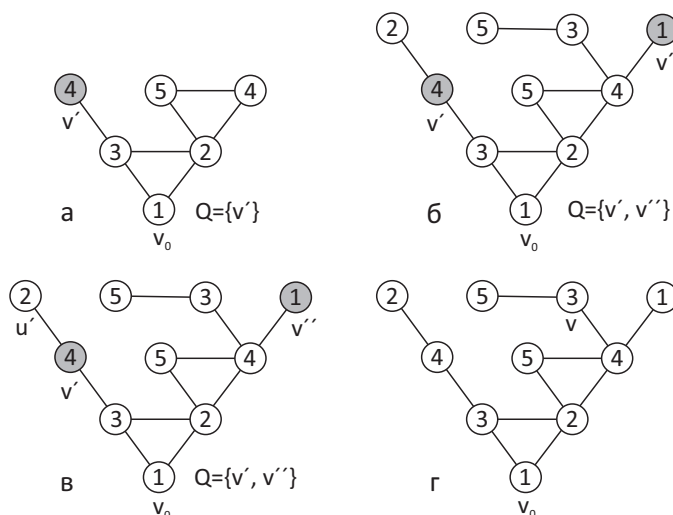


Рис. 1: Ілюстрація процедури побудови Д-графа за заданою парою.

## Визначальна пара для детермінованих графів

Нехай  $G = (V, E, X, \xi(V), v_0)$  – деякий Д-граф. Зафіксуємо на  $X$  лінійний порядок  $<: x_1 < x_2 < \dots < x_p$ . Визначимо на  $X^*$  порядок  $\leq$ :

- 1) для будь-якого  $x_i \in X$ , де  $1 \leq i \leq p$ , покладемо  $x_i \leq x_i$ ;
- 2) для будь-яких  $p, q \in X^*$ , якщо  $d(p) < d(q)$ , то  $p \leq q$ ;
- 3) для будь-яких  $p, q \in X^*$ , якщо  $p = x'_1 \dots x'_s$ ,  $q = x''_1 \dots x''_s$ ,  $x'_1 = x''_1$ ,  $\dots$ ,  $x'_{k-1} = x''_{k-1}$  і для деякого  $k \leq s$  виконується  $x'_k < x''_k$ , то  $p \leq q$ .

Неважко бачити, що порядок  $\leq$  на множині  $X^*$  визначається лінійним порядком  $<$  на множині  $X$ .

Визначимо допоміжну множину слів в алфавіті  $X$ . Базис досяжності  $\mathcal{V}_G$  це довільна множина слів  $\{w_1, w_2, \dots, w_n\} \subseteq X^*$ , яка задовольняє умовам:

- 1)  $\{v_0 w_i | i = 1, 2, \dots, n\} = V$ ;
- 2) якщо  $i \neq j$ ,  $i, j \in \{1, 2, \dots, n\}$ , то  $v_0 w_i \neq v_0 w_j$ ;
- 3) якщо  $w_i = w'_i x$ ,  $x \in X$ , то  $w'_i \in \{w_1, \dots, w_n\}$ ,  $i = 1, \dots, n$ .

Якщо множина  $\mathcal{V}_G$  складається із слів, найменших за порядком  $\leq$ , то називатимемо її найкоротшим базисом досяжності. Зрозуміло, що такий базис визначено однозначно. Далі, якщо не обумовлено інше, вважатимемо, що базис  $\mathcal{V}_G$  найкоротший. Кістякове дерево графа  $G$ , яке природним чином визначається базисом  $\mathcal{V}_G$ , позначимо  $T_V$ . Показано, що за умови проведення пошуку вершин, що можуть підлягати ототожненню в процедурі детермінізації графа за зростанням відповідних ним слів з  $\mathcal{V}_G$ , результатом виконання цієї процедури є Д-граф, що визначається однозначно.

Опишемо алгоритм побудови так званої канонічної визначальної пари  $\{\Sigma_G, \Lambda_G\}$ . Спочатку покладемо  $\Sigma_G = \emptyset$  і  $\Lambda_G = \emptyset$ . Якщо граф  $G$  складається з однієї вершини  $v_0$ , то покладемо  $\Sigma_G = \emptyset$ ,  $\Lambda_G = \emptyset$  і  $\mathcal{V}_G = \{\xi(v_0)\}$ . Нехай граф  $G$  містить більше ніж одну вершину. Спочатку до множини  $\Lambda_G$  додаємо усі слова  $w \in \mathcal{V}_G$  такі, що вершина  $v_0 w$  є висячою вершиною графа  $G$ . Після цього для кожної двійки слів  $p, q \in \mathcal{V}_G - \Lambda_G$ , якщо жодне з них не є початковим відрізком іншого і  $v_0 p q^{-1} = v_0$ , то додаємо до множини  $\Sigma_G$  менше за порядком  $\leq$  із слів  $p q^{-1}$  або  $q p^{-1}$ . Далі прибираємо усі повтори слів у  $\Sigma_G$ , залишаючи по одному екземпляру.

Сформулюємо властивості слів, що належать множині  $\Sigma_G$ .

**Лема 1.** Нехай  $\Sigma_G \neq \emptyset$ . Тоді виконуються такі твердження:

- а) для будь-якого  $\sigma \in \Sigma_G$  існують такі  $p, q \in X^*$ , що  $p q = \sigma$  та  $p \in \mathcal{V}_G$  і  $q^{-1} \in \mathcal{V}_G$ ;
- б) якщо  $\sigma \in \Sigma_G$ , то  $d(\sigma) \leq 4$ ;
- в) для кожного  $\sigma \in \Sigma_G$  знайдуться такі  $p, q \in X^*$ ,  $x, y \in X$ ,  $x \neq y$ , що  $\sigma = p x q y p^{-1}$ .

Для будь-якої вершини  $v \in V$  через  $sp_v$  позначатимемо найкоротше

за порядком  $\leq$  слово, для якого  $v_0 sp_v = v$ .

**Лема 2.** Нехай  $(v_1, v_2)$  – деяке ребро графа  $G = (V, E, X, \xi(V), v_0)$ ,  $\xi(v_1) = x'_1$  та  $\xi(v_2) = x'_2$ . Тоді існують такі  $z_1, z_2 \in X^*$ , що виконується хоча б одне з тверджень:

- а)  $sp_{v_1} x'_2 z_1 \in \Sigma_G \cup \Lambda_G$ ;
- б)  $sp_{v_2} x'_1 z_2 \in \Sigma_G \cup \Lambda_G$ .

**Лема 3.** Якщо хоча б одна компонента  $\{\Sigma_G, \Lambda_G\}$  графа  $G = (V, E, X, \xi(V), v_0)$  не є порожньою множиною, то для кожної вершини  $v \in V$  існують такі  $z_1, z_2, z_3 \in X^*$ , що виконується хоча б одне з тверджень:

- а)  $sp_v z_1 \in \Lambda_G$ ;
- б)  $sp_v z_2 \in \Sigma_G$ ;
- в)  $z_3 (sp_v)^{-1} \in \Sigma_G$ .

З процедури побудови графа за його визначальною парою випливає наступне твердження.

**Теорема 1.**  $G(\{\Sigma_G, \Lambda_G\}) \cong G$ .

Було знайдено чисельні оцінки компонентів канонічної визначальної пари Д-графа  $G = (V, E, X, \xi(V), v_0)$ . Нехай далі  $G$  містить  $n$  вершин та  $m$  ребер.

**Теорема 2.**  $|\Sigma_G| = m - n + 1$ .

Позначимо  $\lceil x \rceil$  – найменше натуральне число, що не менше за  $x$ .

**Теорема 3.** Якщо  $m = n - 1$ , то  $1 \leq |\Lambda_G| \leq n - 1$ . Якщо ж  $m > n - 1$ , то  $0 \leq |\Lambda_G| \leq n - \lceil \frac{3}{2} + \sqrt{\frac{9}{4} - 2 \cdot n + 2 \cdot m} \rceil$ , причому всі оцінки є досяжними (рис. 2).

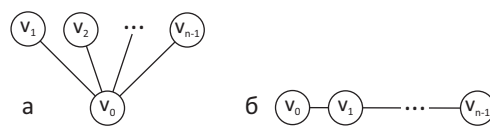


Рис. 2: Древа, для яких максимальна (а) та мінімальна (б) оцінки потужності другої компоненти канонічної визначальної пари є досяжними.

Далі знайдемо оцінки значень  $\|\Sigma_G\|$  та  $\|\Lambda_G\|$  – сумарної довжини усіх слів кожної компоненти визначальної пари  $\{\Sigma_G, \Lambda_G\}$ .

**Теорема 4.**

$4(m - n + 1) \leq \|\Sigma_G\| \leq 4(m - n + 1) \left( n - \lceil \frac{3}{2} + \sqrt{\frac{9}{4} - 2 \cdot n + 2 \cdot m} \rceil + 2 \right)$ , причому ці оцінки є досяжними.

**Теорема 5.** Якщо Д-граф  $G$  є деревом, то  $n \leq \|\Lambda_G\| \leq \lceil \frac{n^2 + 2n}{4} \rceil$ , причому всі оцінки є досяжними.

## Висновки

Таким чином, у роботі наведено лінгвістичне представлення детермінованих графів з ініціальною вершиною. Описано алгоритм побудови канонічної визначальної пари. Знайдено окремі властивості та чисельні оцінки канонічної визначальної пари для детермінованих графів із відомою кількістю вершин та ребер – кількість слів у першій компоненті пари та мінімальні й максимальні досяжні оцінки кількості слів у другій компоненті пари, а також мінімальні й максимальні досяжні оцінки об'єму слів у першій компоненті пари та такі самі оцінки для другої компоненти пари за випадку коли граф є деревом. Подальші дослідження пов'язані з дослідженням окремих класів детермінованих графів – так званих сильно- та слабо-детермінованих.

## Список використаних джерел

- [1] Dudek G., Jenkin M. Computational Principles of Mobile Robotics, 2nd ed. Cambridge: Cambridge Univ. Press, 2010.
- [2] Baier C., Katoen J.-P. Principle of Model Checking. The MIT Press, 2008.
- [3] Grunskii I.S., Senchenko A.S. Properties of systems of defining relations for automata. *Discrete Mathematics and Applications*. 2004. Vol. 14, Issue 6. P. 593–601.
- [4] Grunskii I.S, Mikhaylova I.A., Sapunov S.V. Domination on the vertices of labeled graphs. *Algebra and Discrete Mathematics*. 2012. Vol. 15, Issue 2. P. 174–184.
- [5] Сапунов С.В., Сенченко А.С. Лингвистическое представление графов с помеченными вершинами. *Доповіді Національної академії наук України*. 2019. № 11. С. 17–24.
- [6] Сапунов С.В., Сенченко О.С., Серeda О.А. Метричні властивості канонічної визначальної пари для детермінованих графів. *Праці Інституту прикладної математики і механіки НАН України*. 2020. Т. 34. С. 132–143.

**Хусаїнов Д.Я., Шатирко А.В., Бичков О.С., Шакотько Т.І.**

<sup>1</sup> д. ф.-м. н., професор кафедри моделювання складних систем Київського національного університету імені Тараса Шевченка

<sup>2</sup> д. ф.-м. н., доцент кафедри моделювання складних систем Київського національного університету імені Тараса Шевченка

<sup>3</sup> д. т. н., завідувач кафедри програмних систем і технологій Київського національного університету імені Тараса Шевченка

<sup>4</sup> асистент кафедри дослідження операцій Київського національного університету імені Тараса Шевченка

*e-mail:* <sup>1</sup> d.y.khusainov@gmail.com, <sup>2</sup> shatyrko.a@gmail.com, <sup>3</sup> bos.knu@gmail.com,

<sup>4</sup> trachuk\_85@ukr.net

## **СТІЙКІСТЬ ТА ЗБІЖНІСТЬ В МОДЕЛЯХ КЕРУЮЧИХ НЕЙРОДИНАМІЧНИХ СИСТЕМ**

На основі застосування апарату прямого методу Ляпунова досліджено питання стійкості та збіжності розв'язків в керуючих моделях нейродинаміки (сіток Хопфілда), що описані в термінах функціонально-диференціальних рівнянь з відхиленням аргументу.

**Ключові слова:** *нейронні мережі, стійкість, метод Ляпунова, відхилення аргументу.*

### **Вступ**

З упевненістю можна сказати, що предметом сучасної теорії керування є проектування систем, які, крім того, що у певному сенсі оптимізують заданий критерій якості, ще й (бажано) працюють без втручання людини [1]. Це відповідає одному з напрямків штучного інтелекту, як методу, підходу, способу проектування складних оптимальних систем, на основі використання комп'ютерних систем та технологій [2]. Враховуючи складність сьогоденних реальних завдань (велика розмірність, багатокритеріальність, різноманітна невизначеність), а також необхідність прийняття рішень в інтерактивному режимі, зростає значення використання штучних нейронних мереж при їх вирішенні.

Область знань, у якій нейронні мережі розглядаються як нелінійні динамічні системи й основна увага приділяється проблемам стійкості та збіжності називається нейродинамікою [3-8]. Класична теорія стійкості має на увазі стійкість окремо взятої траєкторії (як правило положення рівноваги). Тому розглядаються моделі, які описуються динамічними системами (диференціальними або різницевиими), що мають одне положення

рівноваги. Якщо розглядати системи, що мають кілька положень рівноваги, то існує поняття «грубі системи». На жаль, якщо для нелінійних систем на площині існують необхідні та достатні умови «грубості», то для систем великої розмірності ці умови відсутні.

Системи рівнянь, що описують динамічні процеси нейродинаміки, мають дуже велику розмірність. Тому пряме використання класичних методів дослідження стійкості тут неможливе. Класичні способи дослідження наштовхуються на важкі проблеми великої розмірності. І в системах, які описують моделі нейродинаміки, часто використовують поняття «системи зі слабкою нелінійністю», тобто системи з виділеною лінійною частиною та нелінійністю, що знаходиться в деякому досить малому лінійному секторі [9-12]. Вочевидь, що з малої величини «слабкої нелінійності» стійкість становища рівноваги нелінійної системи впливатиме з асимптотичної стійкості системи першого лінійного наближення. Звідси, зокрема, витікатиме збіжність процесів навчання штучних нейромереж.

Ще одним підходом дослідження динамічних систем великої розмірності є метод порівняння, який успішно розробляли В.М. Матросов, А.А. Мартинюк та їхні учні. Система розбивається на підсистеми та будуються окремі функції Ляпунова. Висновок про стійкість вихідної системи виробляється виходячи з дослідження «системи порівняння», отриманої з функцій Ляпунова окремих підсистем. Однак, наскільки відомо авторам, цей підхід при дослідженні процесів стійкості моделей нейродинаміки не використовувався.

Також у якості підходу до подолання фактора «великої розмірності» можливо використання поняття «колективна динаміка», яке з'явилося останнім часом. Воно дозволяє досліджувати групову поведінку систем великої розмірності. Однак у цьому напрямі робляться лише перші кроки. Ще однією особливістю, яка виникає при дослідженні динамічних процесів у нейродинаміці, є врахування часу післядії. Реальні процеси неспроможні відбуватися миттєво. Тому ігнорування цього чинника може призвести до помилкових висновків щодо динаміки навчання чи розпізнавання [6,8].

## Постановка задачі

У цій роботі проведено спробу зробити огляд робіт з дослідження стійкості та збіжності розв'язків систем диференціальних рівнянь, які моделюють процеси нейродинаміки. Необхідно провести дослідження неперервно-дискретних динамічних систем із запізненням у задачах ди-

наміки штучного інтелекту при розробці природомовних інформаційних процесів та систем. Зокрема, провести дослідження стійкості нейромереж Хопфілда та побудову оцінки збіжності процесів навчання нейромереж з урахуванням післядії, як у вигляді запізнення аргументу, так і нейтрального типу.

Математичні моделі динаміки нейронних мереж, що описуються нелінійними диференціальними рівняннями, з виділеною асимптотично стійкою діагональною лінійною частиною, розглянуті в [8-10]. Слід зазначити, що при дослідженні динаміки потрібно враховувати час обробки та передачі сигналів. Тому більш адекватними моделями є моделі, що враховують фактор запізнення. Про це було зазначено зокрема в роботі [4].

Як було зазначено в [6], динаміку сукупності елементів нейромережі можна описати системою диференціальних рівнянь

$$C_i \frac{dv_i(t)}{dt} = -\frac{v_i(t)}{R_i} + \sum_{j=1}^n \omega_{ij} x_j(t) + I_j$$

Для заданого індукованого локального поля  $v_i(t)$  можна визначити вихід нейрона  $i$  за допомогою відношення  $x_j(t) = \varphi(v_j(t))$ . Функція активації  $\varphi(\bullet)$ , яка визначає відношення виходу  $x_j(t)$  нейрона  $j$  до його вхідного індукованого локального поля  $v_j(t)$ , є неперервно-диференційовною функцією. Частіше за все в якості функції активації використовують логістичну функцію

$$\varphi(v_j) = \frac{1}{1 + \exp\{-v_j\}}, j = \overline{1, n}.$$

Тоді, використаши заміну

$$a_i = \frac{1}{R_i C_i}, w_i = \frac{\omega_{ij}}{C_i},$$

отримаємо систему

$$\frac{dv_i(t)}{dt} = -a_i v_i(t) + \sum_{j=1}^n w_{ij} \varphi_j(v_j(t)) + K_j.$$

Ще одна модель нейродинаміки може бути описана такою системою диференціальних рівнянь

$$\frac{dx_i(t)}{dt} = -a_i x_i(t) + \varphi \left( \sum_{j=1}^n w_{ij} x_j(t) \right).$$

Мережа Хопфілда складається з багатьох нейронів, що формують систему з множиною зворотних зв'язків. Кількість зворотних зв'язків дорівнює кількості нейронів. Вихід кожного нейрона замикається через елемент одиничної затримки на решту нейронів мережі. Але кожний нейрон цієї мережі не має зворотних зв'язків із собою. У цьому випадку рівняння динаміки можна переписати у вигляді такої системи диференціальних рівнянь

$$C_i \frac{dv_i(t)}{dt} = -\frac{v_i(t)}{R_i} + \sum_{j=1}^n \omega_{ij} \varphi_j(v_j(t)) + I_j.$$

Мережа Хопфілда матиме динамічну рівновагу, якщо на її вхід подається образ, який відображається сам у себе. З погляду фізики повна енергія системи у цій точці буде мінімальною. Енергетична функція (по суті, функція Ляпунова) може бути задана у вигляді суми кінетичної та потенційної енергії

$$V(y) = \frac{1}{2} \sum_{i,j=1}^n w_{ij} y_i y_j + \sum_{k=1}^n \delta_k y_k.$$

І, як випливає з теорем другого методу Ляпунова, стаціонарний стан мережі буде асимптотично стійким, якщо функція Ляпунова буде додатно визначеною, а повна похідна в силу системи буде від'ємно визначеною.

Основною, що розглядається, є мережа Хопфілда, яку описано такою системою звичайних диференціальних рівнянь

$$C_i \frac{dy_i(t)}{dt} = -\frac{1}{R_i} y_i(t) + \sum_{j=1}^n \nu_{ij} \varphi_j(y_j(t)) + I_i, t \geq 0, i = \overline{1, n}.$$

З урахуванням часу обробки сигналів, необхідно використати системи диференціально-різницевого рівняння з запізнюванням аргументу

$$C_i \frac{dy_i(t)}{dt} = -\frac{1}{R_i} y_i(t) + \sum_{j=1}^n \nu_{ij} \varphi_j(y_j(t - \tau)) + I_i, t \geq 0, i = \overline{1, n}.$$

Тут  $n$  – число нейронів мережі,  $y(t) = (y_1(t), y_2(t), \dots, y_n(t))^T$  – вектор стану мережі в момент часу  $t > 0$ ;  $R_i$ ,  $C_i$ ,  $I_i$ ,  $i = \overline{1, n}$  – опір, ємність й зовнішній струм відповідно.

Заміною  $a_i = \frac{1}{R_i C_i}$ ,  $\omega_{ij} = \frac{\nu_{ij}}{C_i}$ ,  $I_i^0 = \frac{I_i}{C_i}$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, n}$  системи можна звести до систем у, так званому, стандартному вигляді

$$\frac{dy_i(t)}{dt} = -a_i y_i(t) + \sum_{j=1}^n \omega_{ij} \varphi_j(y_j(t)) + I_i^0, t \geq 0, i = \overline{1, n}.$$

А з урахуванням запізнювання, остаточно матимемо систему яка реально описує нейродинамічні процеси у мережі Хопфілда

$$\frac{dy_i(t)}{dt} = -a_i y_i(t) + \sum_{l=1}^n \omega_{il} \varphi_j(y_j(t - \tau)) + I_i^0, t \geq 0, i = \overline{1, n}.$$

## Метод дослідження

У даній роботі з метою дослідження стійкості використовується метод функцій Ляпунова з додатковою умовою типу Б.С. Разуміхіна. У роботах [7-9] були отримані умови асимптотичної стійкості положення рівноваги систем із запізненням. При їх побудові використовувався метод автономних функцій Ляпунова з умовою Разуміхіна. Функція будувалася у вигляді суми квадратів фазових координат.

Проте, якщо необхідно отримати не тільки самі умови стійкості, а й оцінки збіжності розв'язків, то автономна функція Ляпунова не підходить. Треба використовувати неавтономну функцію Ляпунова, поверхня рівня якої являє собою «поверхню, що стискається, та містить всі розв'язки всередині себе».

## Висновки

Розглянуті системи диференціальних рівнянь із виділеною негативною лінійною діагональною частиною та нелінійністю спеціального виду. Такі системи зустрічаються при вивченні процесі нейродинаміки, зокрема динаміки нейронних мереж. Отримано умови асимптотичної стійкості положення рівноваги. Розглянуто також системи із запізненням аргументу. Дослідження стійкості проведено з використанням як функціоналів Ляпунова-Красовського так й функцій Ляпунова. Завдяки останньому та з використанням підходу LMI результати є конструктивними, бо всі умови мають вигляд матричних нерівностей, що легко чисельно перевіряються. В подальшому можливе узагальнення на випадок складних нейродинамічних систем, які описуються у вигляді функціонально-диференціальних рівнянь нейтрального типу.

## Список використаних джерел

- [1] Wiener N. Cybernetics or Control and Communication in the Animal and the Machine. Technology Press, John Wiley, 1948.

- [2] Russel S., Norvig P. Artificial Intelligence. A Modern Approach. 3-rd Edition. Prentice Hall, 2009.
- [3] Борисюк Г.Н., Борисюк Р.М., Казанович Я.Б., Иваницкий Г.Р. Модели динамики нейронной активности при обработке информации мозгом – итоги «десятилетия». *Успехи физических наук*. 2002. Т. 172, № 10. С. 1189–1214.
- [4] Галушкин А.И., Цыпкин Я.З.. Нейронные сети: История развития теории. Москва: ИПРЖР, 2001.
- [5] Гитис В.Б. Теория и практика применения нейронных сетей: пособие. Краматорск: ДГМА, 2016.
- [6] Писаренко В.Г. Новая модель функционирования живой нейросети, учитывающая запаздывающее взаимодействие нейронов. *Кибернетика и системный анализ*. 2016. Т. 52, № 6. С. 181–192.
- [7] Розенблатт Ф. Принципы нейродинамики. Москва: «Мир», 1965.
- [8] Haykin S. Neural Networks. A Comprehensive foundation. Prentice Hall, 1998.
- [9] Хусаинов Д.Я., Диблик Й., Баштинец Я., Сиренко А.С. Устойчивость, неравномерная по запаздыванию, одной слабонелинейной системы с последствием. *Труды института прикладной математики и механики НАН Украины*. 2015. Т. 29, № 6. С. 129–146.
- [10] Шатырко А.В., Диблик Й., Хусаинов Д.Я., Баштинец Я. Сходимость процессов нейродинамики в модели Хопфилда. *Штучний інтелект*. 2017. № 3–4. С. 139–148.
- [11] Khusainov D.Ya., Diblik Y., Bastinec Ja., Shatyрко A.V. Investigating Dynamics of One Weakly Nonlinear System with Delay Argument. *Journal of Automation and Information Sciences*. 2018. Vol. 50, Issue 1. P. 20–38.
- [12] Хусаинов Д.Я., Шатырко А.В., Пужа Б., Новотна В., Пилипенко В.А. Исследование динамики класса нейросетей представимых слабо-нелинейными разностными системами. *Штучний інтелект*. 2019. № 1–2. С. 49–58.

<sup>1</sup> аспірант кафедри теоретичної кібернетики Київського національного університету імені Тараса Шевченка

<sup>2</sup> ч-к. НАНУ, д. ф.-м. н., п.н.с. Інституту кібернетики імені В.М. Глушкова НАН України, завідувач кафедри теоретичної кібернетики Київського національного університету імені Тараса Шевченка

*e-mail:* <sup>1</sup> yaserg555@gmail.com, <sup>2</sup> krak@univ.kiev.ua

## **АЛГОРИТМИ СТВОРЕННЯ ІНТЕРАКТИВНОГО ТРЕНАЖЕРА ДЛЯ НАВЧАННЯ ТА ТРЕНУВАННЯ НАВИЧОК СТРІЛЬБИ**

Наводяться результати по створенню тренажер для навчання та тренування одиночному і груповому веденню бою в складі підрозділу. Тренажер є моделлю реальних об'єктів та процесів, тому повинні застосовуватися ті ж самі критерії оцінювання пострілу та дій стрільків з урахуванням погодних умов стрільби та поведінки мішеней (цілей) та вимог до припустимої похибки вимірювань. Удосконалено відомі і розроблено нові алгоритми обробки сигналів: виявлення всіх світлових засвічень, утворених лазерними модулями навчальної зброї та визначення їх координат; збір інформації від апаратної складової тренажерного комплексу; обробка масиву даних, отриманих від кожного макету зброї.

**Ключові слова:** *модель, камера, проектор, тренажер, мультимедійний тип.*

### **Вступ**

Сучасні умови ведення бою вимагають від особового складу стійких індивідуальних практичних навичок використання озброєння та військової техніки в різних умовах обстановки і високої підготовленості до дій у складі тактичних груп. Оснащення збройних сил провідних країн новітніми зразками техніки вимагає зміни поглядів на бойову підготовку, тому надзвичайно актуальними є тренування особового складу за допомогою моделюючих тренажерів, які діють у віртуальному просторі [1].

### **Аналіз зображень влучань на тлі мішені**

Алгоритм визначення положення влучень у ціль базується на аналізі потокового зображення з фотографічного сенсора, із застосуванням оптимізації, яка полягає у розгляді пікселів з певним кроком. Це дозволяє аналізувати не всі пікселі підряд, а тільки у обмеженій зоні пошуку засвічень [2]. Для виявлення корисного сигналу виділяються засвічення на

екрані, що сформовані лазерним модулем навчальної зброї, з урахуванням впливів від сусідніх засвічень, дистанції до мішені та кута влучання.

## **Відтворення реалістичних зображень на тлі мішені**

Реалістичність досягається шляхом візуальної відповідності генерованого зображення реальному об'єкту. Тому повинна забезпечуватися реалістичність окремих об'єктів (фону, мішеней, перешкод тощо) – кутові розміри мішеней повинні відповідати імітованій відстані до цілей, вигляд мішеней повинен відповідати реальним об'єктам, положення кожного об'єкту на екрані повинне відповідати модельованому положенню об'єкта у просторі. Обов'язково забезпечується можливість фіксації точки влучення з урахуванням балістичних характеристик. Одним з ключових моментів є розробка бази мішеней. База мішеней містить мішені різної типології. Реалістичні розміри мішеней, розраховуються програмою з урахуванням зазначених розмірів екрану, відстані до екрану та дистанції стрільби.

## **Результати і подальші дослідження**

Розроблені алгоритми обробки сигналів, що використовуються у тренажері, забезпечують достатню швидкодію для досягнення мінімального проміжку між двома сусідніми пострілами кожного макету та високу точність визначення координат кожного засвічення. При створенні макету тренажера були застосовані методи та алгоритми машинної графіки, що дозволяють масштабувати зображення, вирівнювати краї дискретних зображень, методи та алгоритми моделювання місцевості, мішенної обстановки та стрільби. Подальші дослідження спрямовані на удосконалення запропонованого підходу, проведення експериментів та реалізацію інформаційної технології.

## **Список використаних джерел**

- [1] Крак Ю.В. Яременко С.В., Попова Д.В. Програма формування інтерактивної мішенної обстановки тренажерної системи та формування засад інтелектуалізації менеджменту та управління. Свідectvo про реєстрацію авторського права на твір №95050 від 26.12.2019.
- [2] Yaremenko S.V., Krak, I.V. Determination of the position of the laser spot in the plane of the photo sensor of the multimedia shooting gallery. In *IEEE EUROCON 2021 – 19th International Conference on Smart Technologies : Conference Proceedings*. 2021. P. 1–6.