**Міністерство освіти і науки України**
**Харківський національний**
**університет імені В. Н. Каразіна**

# КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ В НАУКОЄМНИХ ТЕХНОЛОГІЯХ


## ЗБІРНИК НАУКОВИХ ПРАЦЬ МІЖНАРОДНОЇ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**Харків, 22-24 квітня 2020 року**

**Харків**
**2020**

**Министерство образования и науки Украины**
**Харьковский национальный университет**
**имени В. Н. Каразина**

# КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ В НАУКОЕМКИХ ТЕХНОЛОГИЯХ


## СБОРНИК НАУЧНЫХ ТРУДОВ МЕЖДУНАРОДНОЙ НАУЧНО-ТЕХНИЧЕСКОЙ КОНФЕРЕНЦИИ

**Харьков, 22-24 апреля 2020 года**

**Харьков**
**2020**

Затверджено до друку рішенням Вченої ради Харківського національного університету імені В. Н. Каразіна (протокол № 8 від 27.04.2020 р.).

Реєстраційне посвідчення в ДНУ «Український інститут науково-технічної експертизи та інформації» (посвідчення № 800 від 18.12.2019 р.).

Комп'ютерне моделювання в наукоємних технологіях: Збірник наукових праць міжнародної науково-технічної конференції (м. Харків, 22-24 квітня 2020 року) – Х.: ХНУ ім. В.Н. Каразіна, 2020. – 312 с.



**Адреса редакційної колегії:** 61022, м. Харків, майдан Свободи, 6,
ХНУ імені В. Н. Каразіна, к. 534.
Тел. +380 (57) 705-42-81, email: kmht@karazin.ua.

Доповіді, що увійшли до збірника, висвітлюють такі напрямки: математичне моделювання фізичних процесів, моделювання інформаційних процесів в складних та розподілених системах, системи автоматизованого збору та когнітивного подання наукових даних, аналіз процесів в радіаційних, плазмових та інших сучасних технологіях, моделювання транспортних процесів і систем, безпека інформаційних систем і технологій, верифікація та оцінка надійності програмного забезпечення.

Для викладачів, наукових працівників, аспірантів, студентів вишів.

Доклады, включенные в сборник, отражают следующие направления: математическое моделирование физических процессов, моделирование информационных процессов в сложных и распределенных системах, системы автоматизированного сбора и когнитивного представления научных данных, анализ процессов в радиационных, плазменных и других современных технологиях, моделирование транспортных процессов и систем, безопасность информационных систем и технологий, верификация и оценка надежности програмного обеспечения.

Для преподавателей, научных работников, аспирантов, студентов вузов.

ISBN 978-966-285-639-2

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В. Н. КАРАЗІНА**
**ННЦ ХАРКІВСЬКИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**
**MAX PLANCK INSTITUTE OF MICROSTRUCTURE PHYSICS**
**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА**
**INSTITUTE OF NUCLEAR CHEMISTRY AND TECHNOLOGY (Warsaw, Poland)**
**РІВНЕНСЬКИЙ ДЕРЖАВНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ**
**НАЦІОНАЛЬНИЙ АЕРОКОСМІЧНИЙ УНІВЕРСИТЕТ**
**ІМЕНІ М. Є. ЖУКОВСЬКОГО (ХАРКІВ)**
**ЗАТ « ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ » (ХАРКІВ)**
**ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**
**ТОВ "БЮРО ІРІС" ( КИЇВ)**
**TEAM INTERNATIONAL SERVICES, INC. (Lake Mary, USA)**

## ПРОГРАМНИЙ КОМІТЕТ:

Азарєнков М.О., акад. НАНУ, проф., д.ф.-м.н., Харків, голова
Бардачов Ю. М., проф., д.т.н., Херсон
Бомба А.Я., проф., д.т.н., Рівне
Буй Д. Б., проф., д.ф.-м.н., Київ
Ванін В .А., проф., д.т.н., Харків
Горбенко І.Д., проф., д.т.н., Харків
Доля Г.М., проф., д.т.н., Харків
Жолткевич Г.М., проф., д.т.н., Харків
Куклін В.М., проф., д.ф.-м.н., Харків
Лазурик В.Т., проф., д.ф.-м.н., Харків
Рассомахін С.Г., проф., д.т.н., Харків
Савула Я. Г, проф., д.ф.-м.н., Львів
Споров О. Є., доц., к.ф.-м.н., Харків
Стєрвоєдов М.Г., доц., к.т.н., Харків
Styervoyedov A. Dr., Halle, Germany
Толстолузька О.Г., проф., д.т.н., Харків
Ткачук М.В., проф., д.т.н., Харків
Харченко В.С., проф., д.т.н., Харків
Хомченко А.Н. проф., д.ф.-м.н., Миколаїв
Шматков С.І., проф., д.т.н., Харків
Шульга М.Ф., акад. НАНУ, проф., д.ф.-м.н., Харків
Zimek Z., Ph.D., Warsaw, Poland
Яновський В.В., проф., д.ф.-м.н., Харків

## ОРГАНІЗАЦІЙНИЙ КОМІТЕТ:

*Лазурик В.Т.,* д.ф.-м.н., проф., декан ФКН ХНУ імені В.Н. Каразіна, голова,
*Споров О.Є.,* к.ф.-м.н., доц. ХНУ імені В.Н. Каразіна, заст. голови,
*Толстолузька О.Г.*, д.т.н., проф. ХНУ імені В.Н. Каразіна, заст. голови,
*Ткачук М.В.*, д.т.н., проф., зав. каф. МСіТ ХНУ імені В.Н. Каразіна,
*Куклін В.М.*, д.ф.-м.н., проф., зав. каф. ШІ та ПЗ ХНУ імені В.Н. Каразіна,
*Дюльдя С.В.,* к.ф -м.н., ХФТІ,
*Єсін В.І.*, д.т.н., проф., ХНУ імені В.Н. Каразіна,
*Артюх О.А.*, зав. лаб. ХНУ імені В.Н. Каразіна,
*Шевцов С. О.*, директор ТОВ Бюро ІРІС, (Київ)
*Жолткевич Г.М.*, д.т.н., проф., декан ФМІ ХНУ імені В.Н. Каразіна,
*Ванін В .А.*, д.т.н., проф., НТУ «ХПІ» (Харків),
*Зінов'єв Д.В.,* ст. викл. ХНУ імені В.Н. Каразіна,
*Рассомахін С.Г.,* д.т.н., проф., зав. каф. БІСТ ХНУ імені В.Н. Каразіна,
*Styervoyedov A. Dr.*, Max Planck Institute of Microstructure Physics (Germany),
*Петерсен С.,* виконавчий директор TEAM International (Харків),
*Стєрвоєдов М.Г.,* к.т.н., доц., зав. каф. ЕУС ХНУ імені В.Н. Каразіна,
*Шматков С.І.*, д.т.н., проф., зав. каф. ТПС ХНУ імені В.Н. Каразіна,
*Кругол М.М.*, асистент НТУ «ХПІ».

**http:// www.univer.kharkov.ua**
**http:// www-csd.univer.kharkov.ua**

**ЗМІСТ**

**УДК 004.8**

АЛЬОШИНА М. В.

# МОДЕЛЮВАННЯ ПРОЦЕСІВ ПРИЙНЯТТЯ РІШЕННЯ ЗА ДОПОМОГОЮ НЕЧІТКИХ НЕЙРОНІВ ДЛЯ МУЛЬТИАГЕНТНОЇ СИСТЕМИ

**Нечітка логіка**

Математична теорія нечітких множин і нечітка логіка є узагальненнями класичної теорії множин та класичної формальної логіки. Поняття нечіткої логіки було впроваджено американським ученим Лотфі Заде (Lotfi Zaden) в 1965 р і причиною зародження даної науки була наявність нечітких та приблизних міркувань людини при спробі описати деякі явища, процеси, системи та інше.

Нечітка логіка являє собою перш за все логіку і, як відомо, будь-яка логічна функція може бути представлена диз'юнктивною або кон'юнктивною нормальною формою, з цього маємо висновок, що для реалізації обчислення висловлювань достатньо лише трьох операцій: кон'юнкції (&&), диз'юнкції (||) та заперечення (!). В класичній логіці кожна з цих операцій може бути представлена у вигляді таблиці істиності:

*Табл.1 Таблиці істиності, приклад кон'юнкції та диз'юнкції*

| $a$ | $b$ | && | $a$ | $b$ | \|\| |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

Але в нечіткій логіці замість величин *істина* та *брехня* використовується таке поняття як *ступінь істинності*, яке може приймати любі значення з нескінченної множини від 0 до 1 або іншої величини – верхньої границі. З цього можна зробити висновок, що логічні операції вже не можна представити таблично, в нечіткій логіці вони задаються у вигляді характеристичних функцій (функцій належності) - $\mu_{\mathrm{A}}(x)$, яка приймає значення на множині *M*. В свою чергу нечіткі (fuzzy) множини – це множини елементів, які описані характеристичною функцією.

Нечіткою множиною *A* називається сукупність пар

$$A = \{\langle x, \mu_A(x)\rangle \,|x \in U\} \tag{1}$$

де $\mu_{\mathrm{A}}(x)$ – функція належності, тобто $\mu_{\mathrm{A}}(x)$: U → [0, 1].
Нехай, наприклад,

$$U = \{a, b, c, d, e\},$$
$$A = \{\langle a, 0\rangle, \langle b, 0.1\rangle, \langle c, 0.5\rangle, \langle d, 0.9\rangle, \langle e, 1\rangle\}.$$

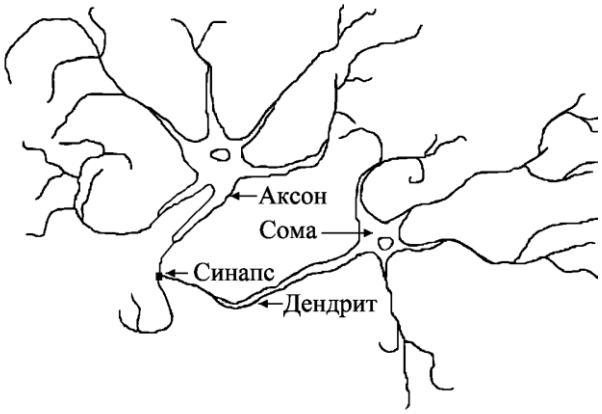
Будемо казати, що елемент $a$ не належить множині *A*, елемент $b$ належить їй в малому ступені, елемент $c$ більш менш належить, елемент $d$ належить в значному степені, $e$ є елементом множини *A*.

За допомогою нечіткої логіки стало можливим вирішення безлічі проблем не лише в математичній сфері, великого поширення нечітка логіка здобула в вирішенні проблем штучного інтелекту, зокрема нейронних мереж, адже задля того, щоб навчити систему приймати рішення як людина, треба навчити її мислити як людина, та в багатох випадках неможливо стовідсотково зазначити правильну відповіть, але можна визначити ступінь приналежності до певної характеристики. Простим прикладом може виступати визначення «гарячого чаю»: перед нами стоїть задача визначити є чай гарячим, чи ні, отже класична логіка описана лише двома

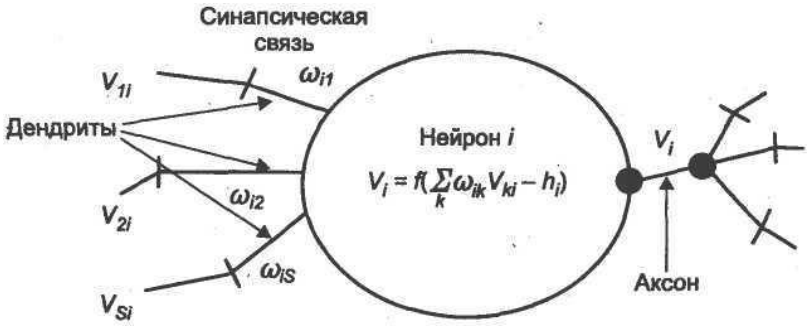
значеннями – 0 або 1 (чай або гарячий, або холодний), але для різних експертів поняття «гарячий» є різним, отже для одного єксперта чай з температурою 60 С буде гарячим, а для іншого з температурою 80 С, і ми можемо бачити, що для цього випадку класична логіка не пасує, але нечітка логіка може вирішити дану проблему. Нехай температура води змінюється від 0 до 100 градусів, можемо зазначити нечітку множину M = {0/0; 0,25/20; 0,5/50; 0,75/80; 1/100}, отже чай з температурою 80 С принадлежить до множини «гарячий» зі ступенем приналежності 0,75. Так для однієї людини чай такої температури буде гарячим, а для іншої – не дуже. В цьому і проявляється нечіткість завдання відповідної множини.

**Нейронна мережа в поданні нечіткої логіки**

Ідея об'єднати природну мову і сформовані людиною поняття системи нечіткої логіки зі штучними нейронними мережами належить Ж. -С. Р. Чангу *(J. -S. Roger Jang)* з Тайванського університету. Нечіткі нейронні мережі побудовані за наступним принципом: виводи робляться на основі апарату нечіткої логіки, а функції належності підлаштовуються з використанням алгоритмів навчання нейронних мереж. Штучні нейронні мережі будуються за принципами організації й функціонування їх біологічних прототипів. Вони здатні вирішувати широкий круг задач роспізнавання образів, ідентифікацій, прогнозування, оптимізації, управління складними об'єктами.




*Мал. 1 Біологічний нейрон*




*Мал. 2 Математична модель нейрона*

На вхід штучного нейрону поступає множина деяка сигналів, кожен з яких є виходом ішного нейрону. Кожен вхід множиться на відповідну вагу, аналогічну синаптичній силі, всі множення складаються, зазначаючи рівень активації нейрону.

Математична модель нейрону виглядає наступним чином:

$$S = \sum_{i=1}^{n} w_i x_i + b \qquad (2)$$

де $w_i$, - вага синапсу, i = 1...n; b – значення зміщення; s – результат складання; x – компонент вхідного вектору (вхідний сигнал), $x_i$ – 1…n; y – вихідний сигнал нейрону; n – число входів нейрону.

Нейронна мережа – це павутиння з нейронів, яку для того, щоб вона робила щось осмислене, треба попередньо навчити. На даний момент існує обмежена кількість можливостей навчити штучні нейронні мережі. Розрізняють алгоритми навчання з вчителем та без вчителя.

**Постановка задачі в предметній галузі**

Роздивимося рішення задачі про динаміку формування ціни на біржі методами нечіткої логіки. Проблема – створюванні крупними покупцями й продавцями біржі слугують для полегшення процесів торгівлі та знижують витрати на економічні відносини. Ці організації не є державними, але повинні слідувати встановленим законодатством вимогам. Велика кількість покупців представляють свої інтереси фіксацією ціни і необхідного ними обсягу товару. Товар фіксується і до продажу не може бути реалізований, також суми коштів покупців фіксуються і не можуть бути ними використані до покупки. Проблема в тому, що між мінімальною ціною продажу і максимальною ціною покупки є інтервал-спред, який треба подолати. Необхідність продажів і покупок змушує торговців шукати цю можливість, знижуючи або підвищуючи свої ціни. Процес, який організовують біржові брокери, є досить нервовим та супроводжується коливанням цін пропозиції і попиту. Якщо починаються покупки нервовий стан посилюється, всі хочуть приєднатися до покупців та продавців, що домовилися про ціну. При цьому вони вже готові змінювати ціни в більшому масштабі.

**Представлення моделі опису процесу**

Модель, яка описує даний процес наступна.

Нехай продавці хочуть отримати ціну

$$y_j = y_{j0} - g_j \frac{1}{1-G} f_y \quad (3)$$

де $y_{j0}$ – заявлена ціна продажу j-го учасника, $g_j$ – можливе змінення ціни, зв'язане з кількістю продукту, що продається, якщо продукту багато, тоді згода знизити ціну превалює більше, $\frac{1}{1-G}$ – коефіцієнт, якісно демонструючий як впливає об'єм вже проданого на змінення ціни, $f_y$ – випадкова величина, яка змінюється від 0 до 1, імітуючи нерішучисть або навпаки відчайдушну сміліть продавців.

Покупці в свою чергу хочуть отримати ціну

$$x_i = x_{i0} - q_i \frac{1}{1-G} f_x \quad (4)$$

де $x_{i0}$ – заявлена ціна продажу і-го учасника, $g_j$ – можливе змінення ціни, зв'язане з кількістю продукту, що купується, $f_y$ – випадкова величина, яка змінюється від 0 до 1, яка імітує змінення настрою покупців.

В динаміці тимчасової, при якій випадкові величини змінюються, слід зазначити, хто саме з покупців й продавців домовився про ціну. Умова цього

$$T[(y_1, y_2, \dots, y_N)] = S\,[(x_1, x_2, \dots, x_M)] \quad (5)$$

де

$$T[(y_1, y_2, \dots, y_N)] \equiv T[(y_1 \wedge y_2 \wedge \dots \wedge y_N)], \quad (6)$$

$$S[(x_1, x_2, \dots, x_M)] \equiv S[(x_1 \wedge x_2 \wedge \dots \wedge x_M)], \quad (7)$$

кон'юнкцію та диз'юнкцію можна обирати у вигляді

$$A \wedge B = min(A, B), \quad (8)$$

$$A \vee B = max(A, B). \quad (9)$$

З'ясувати конкретну пару покупця та продавця, які погодилися на взаємні узгодження умови можна розглянувши формулу процедури (3).

Тепер визначимо величину

$$G = \frac{\min\{\sum q_i^*; \sum g_j^*\}}{\{\sum g_j\}}, \quad (10)$$

тобто це ті значення реальних відносних продажів й купівль,які відповідають покупцям і продавцям, які досягли згоди в ціні (відмічені зірочкою). Інші можуть не погоджуватися. $\{\sum g_j\}$ – загальний об'єм продажів.

Повернемося до процедур продажів й купівль. Якщо $g_{j1} < q_{i1}$, тобто продавець має меньше товару, ніж покупець, тоді продавець $j_1$ зникає з ринку, а покупець, трохи задовільнившись, залишається, але змінюється

$$x_{i1} = x_{i10} + q_{i1}\frac{1}{1-G}f_x \Rightarrow x_{i1} = x_{i10} + (q_{i1} - g_{j1})\frac{1}{1-G}f_x, \quad (11)$$

Якщо ж $g_{j2} > q_{i2}$, тобто покупець хоче меньше товару, ніж має продавець, тоді покупець зникає з ринку, а продавець трохи змінюється

$$y_{j2} = y_{j20} + g_{j2}\frac{1}{1-G}f_y \Rightarrow y_{j2} = y_{j20} + (g_{i2} - q_{j2})\frac{1}{1-G}f_y, \quad (12)$$

Все що продане записується в (10) та враховується у виразах (3) і (4). Початкові умови

$$x_{i0} \le -a, y_{j0} \ge a, \quad (13)$$

$g_j$ та $q_i$ знаходяться в інтервалі $0.3 \div 1$ й позитивні, а випадкові величини $f_x$ та $f_y$ позитивно зазначені з максимальним значенням рівним одиниці, міняються з часом, перевіряється умова (5), частково, за вказаними вище правилами виключаються учасники, які досягли однакової ціни.

ЛІТЕРАТУРА

1. Асадуллаев Р. Г. Нечеткая логика и нейронніе сети: уч. пособие Белгород: НИУ «БелГУ», 2017. 309 с.
2. Куклин В. М. Представление знаний и операци над ними: уч. пособие Харьков: ХНУ им. В. Н. Каразина, 2019. 180 с.
3. Нильсон Н. Принципы искусственного интеллекта / Н. Нильсон ; пер. с англ. – Москва : Радио и связь, 1985. – 376 с.
4. Jang J. S. R. ANFIS: adaptive−network−based fuzzy inference system / J. S. R. Jang // IEEE transactions on systems, man, and cybernetics,1993. – Vol. 23. – № 3. – P. 665–685.
5. Zadeh Lotfi A. Fuzzy sets / Lotfi A. Zadeh // Information and Control.1965. – Vol. 8. – P. 338–353. Kosko B. Fuzzy systems as universal approximation / B. Kosko // IEEE Transactions on Computers, 1994. – Vol. 43. – № 11. – P. 1329–1333.
6. Яхъяева Г.Э. Нечеткие множества и нейронные сети: уч. пособие Москва: Интернет-университет Информационных технологий, 2006. 316 с.
7. Богатков В.Н., Дранишников Л.В., Пророков А.Е. Построение систем управления на основе нейронных сетей: уч.-методическое пособие Апатиты: Изд-во КФ ПетрГУ, 2011. 41 с.
8. Основные понятия и определения нечетких нейронных (гибридных) сетей. URL: https://studme.org/133137/informatika/osnovnye_ponyatiya_opredeleniya_nechetkih_neyronnyh_gibridnyh_setey
9. Устройство бирж. URL: https://siver-irk.ru/ustrojstvo-birzh/
10. Фондовый рынок: Как устроены биржи и зачем они нужны? URL: https://habr.com/ru/company/iticapital/blog/210570/

**АЛЬОШИНА Марія Володимирівна** – студентка факультету компьютерних наук; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків, Україна, 61022; e-mail: mariyaaloshyna34@gmail.com; ORCID: 0000-0003-3206-3551.

Наукові інтереси:

*– експертні системи, штучний інтелект.*

**УДК 004.272**

АНЖУРОВ В.Е., ТОЛСТОЛУЖСКАЯ Е.Г.

# КОМПЬЮТЕРНАЯ МОДЕЛЬ ПРЕПРОЦЕССИНГА ДАННЫХ В DATA MINING

**Краткие теоретические сведения**

Препроцессинг данных является важным этапом при работе с большими объемами данных. Как правило, первоначально данные собираются без структурирования и классификации, с целью экономии времени сбора. В результате получаем «сырые» и неструктурированные данные, с которыми в последствии крайне сложно работать. Задачей препроцессинга является как раз конвертирование «сырых» данных в форму данных, по которым можно производить вычисления, поиск и т. д.

Задачей препроцессинга является:

1) Очистка данных от так называемого «шума»;
2) Преобразование информации, то есть преобразование данных в структуру для последующей работы;
3) Сжатие объема данных.

**Метод препроцессинга данных**

В рамках работы был реализован этап пробразования данных в группы со схожими свойствами, именуемый кластеризацей. Для реализации был выбран метод «k-средних». Метод k-средних – это метод кластерного анализа, целью которого является разделение m наблюдений (из пространства $R^n$ )на k кластеров, при этом каждое наблюдение относится к тому или иному кластеру, к центру которого оно ближе всего. В качестве меры близости используется Евклидово расстояние:

$$\rho(x,y) = \left||x-y|\right| = \sqrt{\sum_{p=1}^{n}(x_p - y_p)^2}\,,\ x,y \in R^n \qquad (1)$$

Итак, рассмотрим ряд наблюдений $\left(x^{(1)}, x^{(2)}, \dots, x^{(m)}\right), x^{(j)} \in R^n$. Метод k-средних разделяет m наблюдений на k групп (или кластеров) (k <= m) $S = \{S_1, S_2, \dots, S_k\}$, чтобы минимизировать суммарное квадратичное отклонение точек кластеров от центродов этих кластеров.

Если мера близости до центроида определена, то разбиение объектов на кластеры сводится к определению центроидов этих кластеров. Число кластеров k задается исследователем заранее. [2].

Рассмотрим первоначальный набор k средних (центроидов) $\mu_1, \dots, \mu_k$ в кластерах $S_1, S_2, \dots, S_k$. На первом этапе центроиды кластеров выбираются случайно или по определенному правилу (например, выбрать центроиды, максимизирующие начальные расстояния между кластерами).

Относим наблюдения к тем кластерам, чье среднее (центроид) к ним ближе всего. Каждое наблюдение принадлежит только к одному кластеру, даже если его можно отнести к двум и более кластерам. Затем центроид каждого i-го кластера перевычисляется по следующему правилу [3]:

$$\mu_i = \frac{1}{S_i}\sum_{x^{(i)} \in S_i} x^{(i)} \qquad (2)$$

Таким образом, алгоритм k-средних заключается в перевычислении на каждом шаге центроида для каждого кластера, полученного на предыдущем шаге. Алгоритм останавливается, когда значения $\mu_i$ не меняются.

Неправильный выбор первоначального числа кластеров k может привести к некорректным результатам. Именно поэтому при использовании метода k-средних важно сначала провести проверку подходящего числа кластеров для данного набора данных.

**Сведения о разрабатываемой модели**

Разрабатываемая модель обрабатывает информацию о студентах. На вход программе подается Excel файл со следующей структурой:

*Табл.1 Структура входных данных*

| ФИО студента | Средний балл | Процент посещаемости |
|---|---|---|
| Петров Александр Викторович | 68,5 | 44,5% |
| Сизиков Игорь Валерьевич | 77,6 | 54,6% |
| Птушкина Светлана Степановна | 51,6 | 32% |
| Григорьев Олег Станиславович | 89,2 | 64,5% |
| Королькова Елена Игоревна | 95,66 | 60,1% |

Таким образом, у нас есть собранные данные о студентах, их посещаемости и успеваемости. Целью является оценить общую картину успеваемости/посещаемости студентов. Для этого, необходимо реализовать алгоритм кластеризации, то есть объединения студентов в группы не по каким-либо правилам, а исходя из того, что в группе должны находится студенты со схожими свойствами.

Схема модели представлена на рисунке ниже:

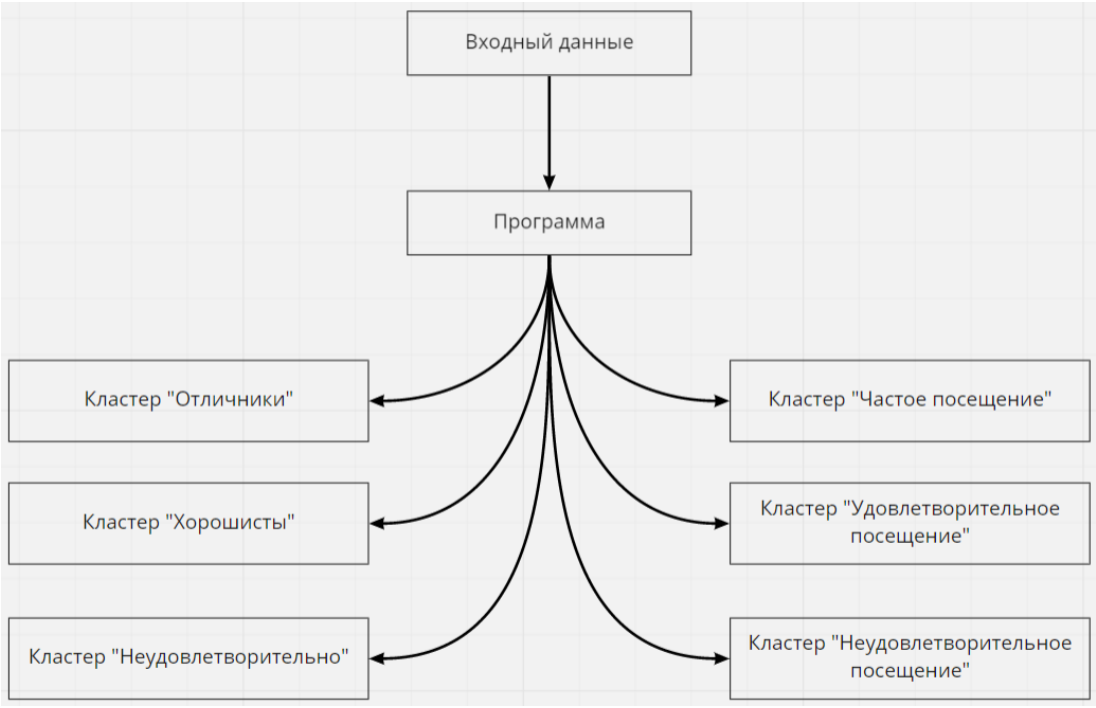



*Рис. 1 Схема модели*

На схеме модели мы можем видеть входные данные (на практике Excel файл со структурой из Табл. 1). Эти данные поступают в программу для препроцессинга. Результатом программы являются 6 наборов данных, называемых «кластерами» (на практике Excelфайлы): 1-3 кластеры содержат студентов с отличными, хорошими и неудовлетворительными оценками; 4-6 кластеры содержат студентов с частой, удовлетворительной и неудовлетворительной посещаемостями. Также, каждый из Excelфайлов содержит так называемый «центроид» кластера для понимания картины успеваемости/посещаемости

В рамках исследуемого предмета важно понимать, что кластеризация – не тоже самое, что классификация [4]. В классическом понимании, отличники – студенты получающие 90-100 баллов. В нашей задаче нет «правил» определения уровни успеваемости. Есть лишь схожие черты и свойства студентов по которым формируются кластеры. Все зависит от общего количества студентов и индивидуальных оценок. Таким образом, в результате может оказаться, что отличники – это студенты, получающие 50-60 баллов, что есть результатом факта того, что остальные студенты получили 0-50 баллов и их количество превышает количество отличников в разы.

**Выводы**

Рассматриваемый метод кластеризации полностью удовлетворяет требованиям современного мира. Он быстро и четко выполняет поставленную задачу, что доказывает корректно работающая модель препроцессинга данных

ЛІТЕРАТУРА

1. Bill Schmarzo Big Data: Understanding How Data Powers Big Business; Wiley - M., 2013. - 240 с.
2. Dr. Arvind Sathi Big Data Analytics: Disruptive Technologies for Changing the Game; Огни - Москва, 2012. - 323 с.
3. Frank J. Ohlhorst Big Data Analytics: Turning Big Data into Big Money (Wiley and SAS Business Series); Гостехиздат - Москва, 2012. - 176 с.
4. Vignesh Prajapati Big Data Analytics with R and Hadoop; КнигапоТребованию - M., 2013. - 238 с.

**АНЖУРОВ Валентин Евгеньевич –** студент группы КИ-41 факультета компьютерных наук; Харьковский национальный университет им. В. Н. Каразина, площадь Свободы, 6, Харьков, Украина, 61077; e-mail: triathlete433@gmail.com; ORCID: 0000-0002-5486- 781X.

Научные интересы:

– *Программирование, структуры данных.*

**ТОЛСТОЛУЖСКАЯ Елена Геннадиевна –** д.т.н., с.н.с., профессор кафедры теоретической и прикладной системотехники факультета компьютерных наук; Харьковский национальный университет им. В. Н. Каразина, площадь Свободы, 6, Харьков, Украина, 61077; e-mail: elenatolstoluzka@gmail.com; ORCID: 0000-0001-2741- 180.

Научные интересы:

– *Технологии автоматического проектирования параллельных программ.*

**УДК 004**

АФАНАСЬЄВА Х.О., ТОЛСТОЛУЗЬКА О.Г.

# МОДЕЛЬ УРАХУВАННЯ ДОСЛІДНИЦЬКОЇ ДІЯЛЬНОСТІ ПРАЦІВНИКІВ СИСТЕМИ ОСВІТИ НА БАЗІ ОПЕРАЦІЙНОЇ СИСТЕМИ IOS

**Вступ**

Розробка мобільних додатків відіграє все більш важливу роль для організацій, яким необхідно спілкуватися зі співробітниками або клієнтами за допомогою вбудованих додатків. На сьогоднішній день існує великий вибір мов програмування для розробки мобільних додатків. Це пов'язано з тим, що для різних мобільних пристроїв доводиться використовувати різні мови програмування, що обумовлене тим, що мобільні пристрої мають різні операційні системи (ОС).

Цільова платформа (iOS, Android, Windows Phone) буде мати значний вплив на мову розробки, яка буде використовуватися. Наприклад, можна розробляти рідні додатки для кожної платформи або використовувати сторонній інструмент для оптимізації своїх додатків на різних платформах. Другий підхід може заощадити час і зусилля, хоча це може вплинути на зручність використання. Сучасні мобільні пристрої пропонують широкий спектр варіантів розробки.

***Метою роботи*** є створення додатку для оптимізації інформації про наукову діяльність працівників системи освіти в рамках факультету комп'ютерних наук обчислювальними засобами мови програмування Swift на базі існуючої комп'ютерної моделі розробленої для системи Windows.

**Swift**

Програмування – основа основ комп'ютерної техніки. Корпорація Apple давно відома своїм умінням задавати тон розвитку індустрії на роки вперед, але з огляду на поступове сходження купертіновцев з ринку професійних рішень, надкушене яблуко асоціюється в першу чергу з споживчими товарами. Однак чергове дослідження громадської думки показує, що розробники ПЗ більш ніж задоволені свіжою пропозицією компанії – мовою програмування Swift. Згідно з інформацією ресурсу StackOverflow, майже 80 відсотків професіоналів із задоволенням працювали або планують працювати з випущеним не так давно інструментом розробки від Apple [2]. Дослідникам вдалося опитати 26 тисяч відвідувачів з більш ніж 150 країн світу. Близько третини постійно зайнятих в сегменті написання мобільного ПЗ респондентів працюють в основному на платформі iOS, а менше половини також займаються створенням додатків для конкуруючої ОС Android.

Swift – багатопарадигмова компільована мова програмування, розроблена компанією Apple для того, щоб співіснувати з Objective C і бути стійкішою до помилкового коду. Swift була представлена на конференції розробників WWDC 2014. Мова побудована з LLVM компілятором, включеного у Xcode 6 beta. Безкоштовний посібник мови програмування Swift доступний для завантаження у магазині iBooks.

Компілятор Swift побудований з використанням технологій вільного проекту LLVM. Swift успадковує найкращі елементи мов C і Objective-C, тому синтаксис звичний для знайомих з ними розробників, але водночас відрізняється використанням засобів автоматичного розподілу пам'яті і контролю переповнення змінних і масивів, що значно збільшує надійність і безпеку коду.

При цьому Swift-програми компілюються у машинний код, що дозволяє забезпечити високу швидкодію. За заявою Apple, код Swift виконується в 1.3 рази швидше коду на Objective-C. Замість збирача сміття Objective-C в Swift використовуються засоби підрахунку посилань на об'єкти, а також надані у LLVM оптимізації, такі як автовекторизація. Мова також пропонує безліч сучасних методів програмування, таких як замикання, узагальнене програмування, лямбда-вирази, кортежі і словникові типи, швидкі операції над колекціями, елементи функційного програмування. 26 Основним застосуванням Swift є розробка користувацьких застосунків для Mac OS X та AppleiOS з використанням фреймворкаCocoa і CocoaTouch. При цьому Swift надає об'єктну модель, сумісну з Objective-C. Вихідний код мовою Swift може

змішуватися з кодом на C і Objective-C в одному проекті [3]. Swift щільно інтегрований у власницьке середовище розробки Xcode і не може бути використаний відособлено на платформах, відмінних від OS X. Окремо варто відзначити, що Swift від компанії Apple не варто плутати з досить давно розробленою скриптовою мовою Swift, націленої на багатониткове програмування і поставленого під вільною ліцензією Apache [4-8].

**Висновки**

Тема розробки мобільних застосувань під платформу iOS є досить цікавою і представляє широке поле для подальших досліджень в галузі розробки мобільного ПЗ. Специфіка даного сегмента полягає в тому, що розробка iOS-додатків повинна проводитися з урахуванням особливостей мобільних пристроїв: відмінностями інтерфейсу, іншим розміром екрану, сенсорним управлінням. Актуальність теми підкреслюється широким спектром можливостей для втілення ідей у вигляді мобільного додатку.

Протягом останніх років показник, що характеризує рівень попиту на мобільні пристрої, постійно зростає. Така статистика дозволяє зробити висновок про те, що розробка мобільних додатків актуальна і доцільна. Отже, тільки корисна розробка отримає гідне визнання з боку користувачів.

ЛІТЕРАТУРА

1. ANON The Swift Programming Language (Swift 2.1) / ANON – Cupertino: Apple Inc., 2014. 528 p.
2. Vandad Nahavandipoor iOS 8 Swift Programming Cookbook / Vandad Nahavandipoor – Boston: O'Reilly Media., 2014. – 902 p.
3. Офіційна документація мови програмування Swift. – Режим доступу: https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/. – Дата доступу 10.03.2020.
4. Innovations Science and Technology: the XVI All-Ukrainian R&D Students Conference Proceeding, (Kyiv, April 18, 2016) / National Technical University of Ukraine "Kyiv Polytechnic Institute". – Part II. – Kyiv, 2016. – 116 p.
5. Гамма, Р.Хелм, Р. Джонсон, Дж. Влиссидес. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб: Питер, 2001. – 368 с.
6. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Problem 2-2, pg.38.

**АФАНАСЬЄВА Христина Олександрівна** – студентка кафедри теоретичної та прикладної системотехніки; Харківський національний університет імені В. Н. Каразіна, м. Харків, майдан Свободи, 4, 61022; e-mail: krisaaf98@gmail.com; ORCID: 0000000262624818.

Наукові інтереси:

– *Програмування, структури даних.*

**ТОЛСТОЛУЗЬКА Олена Геннадіївна** – професор кафедри теоретичної та прикладної системотехніки, д. т. н., с. н. с.; Харківський національний університет імені В. Н. Каразіна, м. Харків, майдан Свободи, 4, 61022; e-mail: elenatolstoluzka@gmail.com; ORCID: 0000000312417906.

Наукові інтереси:

– *Технології автоматичного проектування паралельних програм.*

**УДК 622. 276.53**

БАКУМЕНКО Н.С., МЕНЯЙЛОВ Є.С., УГРЮМОВ М.Л., ЧЕРНИШ С.В.

# ІДЕНТИФІКАЦІЯ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ НА ОСНОВІ МЕТОДІВ ГЛИБОКОГО НАВЧАННЯ

**Вступ**

Нейромережеві технології знайшли застосування в економіці, медицині, промисловості, багатьох інших областях науки і техніки, здатні вирішувати практично будь-які завдання, пов'язані з моделюванням, прогнозуванням, оптимізацією [1]. Проблема ідентифікації моделей для складних техничніх систем полягає в тому, що, як правило, виробничі процеси характеризуються більшою розмаїтістю динамічно взаємодіючих параметрів і зазвичай занадто складні для створення адекватних аналітичних моделей [2]. Слід також зазначити, що в деяких випадках вдалі з точки зору адекватності описуваного процесу аналітичні математичні моделі виявляються неспроможними через високі вимоги до обчислювальної потужності [3]. На даний момент опубліковано безліч робіт, присвячених опису теорії, методів навчання, практику застосування штучних нейронних мереж (ШНС), що навчаються, в різних областях науки і техніки [4-7].

Слід зазначити, в більшості робіт, присвячених вирішенню завдань формування формальних математичних моделей на базі штучних нейронних мереж, відсутній аналіз стійкості рішень до збурень вхідних даних і похибок обчислень. Також до недоліків можна віднести використання апарату ШНМ для вирішення конкретних завдань, що зменшує універсальність і широту застосування [8-10].

Дана робота присвячена дослідженню некоректних задач відновлення процесів-ня багатовимірних статистичних залежностей на основі емпіричних даних, побудови нейромережевих моделей (на прикладах односпрямованої багатошарової ШНМ) за допомогою интеллектуальних інформаційних технологій. В роботі використані моделі крігінга, це пов'язано з тим що неможливо отримати точні значення використовуючи стати-стичні дані, тому вводиться певна ймовірність досягнення необхідної точності. Опис і використання таких моделей наведені в роботах [11-12]. Представлені приклади практичного використання побудований-них таким чином моделей для вирішення завдань аналізу параметрів технічних систем.

**Метод навчання односпрямованої багатошарової штучної нейронної мережі**

Вхідними даними для апроксимації даних є вхідні параметри та керуючі змінні дослідних зразків (аналогів) $\left\{Y_{ph}^{(0)}\right\}$; вихідні параметри $\{d_{pi}\}$.

Для нормалізації початкових даних було використане перетворення:

$$f^0 = \frac{2l_f(f-<f>)}{f_{max}-f_{min}}, \text{ де } <f>=(f_{max}+f_{min})/2, f^0 \in [-1,1],$$

обернене перетворення:

$$f = \left[\frac{(f_{max}-f_{min})f^0}{l_w} + (f_{max}+f_{min})\right]/2, \text{ де } l_w = th(\beta).$$

Алгоритм навчання нейронної мережі зі зворотним поширенням помилки, полягає в знаходженні значення помилки між фактичним і бажаним вихідними даними мережі. Зменшення значення помилки можна досягти, модифікуючи параметри мережі. Процес повторюється до досягнення мережею здатності виконувати бажаний тип перетворення «вхід вихід». В результаті навчання знаходяться ваги зв'язків шарів мережі, а для вихідних параметрів та управляючих змінних – вихідні параметри об'єкта, що вивчається.

Аналітичне подання шуканих функцій для односпрямованої багатошарової мережі має наступну структуру:

$$Y_i^{(2)} = f\left(s_i^{(2)}\right), s_i^{(2)} = w_{i0}^{(2)} + \sum_{j=1}^{H_1} w_{ij}^{(2)} Y_j^{(1)}, i = 1..H_2$$

$$Y_j^{(1)} = f\left(s_j^{(1)}\right), s_j^{(1)} = w_{j0}^{(1)} + \sum_{h=1}^{H_0} w_{jh}^{(1)} Y_h^{(0)},$$

де $f(s) = th(\beta_s)$ – обрана функція активації.

Для забезпечення стабільності (робастності) і інформативності параметрів статистичних моделей систем (процесів) на основі яких навчають ШНМ при апріорній невизначеності вхідних даних, а також достатньою з практичної точки зору точності апроксимації даних доцільно в якості методів навчання ШНМ використовувати стабільні (робастні) статистичні методи оцінки їх параметрів

В якості скалярної згортки цільових функцій використовувалася функція виду:

$$E = \frac{1}{2PI}\sum_{p=1}^{P}\gamma^{p-p}\sum_{i=1}^{I}\left\{ f_{fit}\left[4\left(\frac{\Delta_{f_i,p}}{f_i^*}\right)^2\left(1+\sigma^0_{f_i,p}\right)^{-2}\right] + \beta_{t+1}\cdot f_{fit}\left[((\sigma^0_{f_i,p})^2-1)\right]\right\} \quad (1)$$

де $I = H_{K+1}$, $\gamma$ – рівень значущості ($\gamma = [0.95; 0.99]$),

$$f_{fit}(d_i) = 1 - \exp\left[\left(-\frac{L_{fit}}{4}\right)d_i\right],\ L_{fit} \geq 4\ (d_i > 0),\ \left(\sigma^{\circ}_{f_i}\right)^2 = \frac{\left(\sigma_{f_i}\right)^2}{\left(\sigma^*_{f_i}\right)^2},$$

$$\sigma^2_{f_i} = \beta^4[1-f^2(s_i^{(2)})]^2\cdot\sum_{j=1}^{H_1}\{(w_{ij}^{(2)})^2[1-f^2(s_j^{(1)})]^2\cdot\sum_{h=1}^{H_0}[(w_{jh}^{(1)})^2(\sigma^*_{Y_h^{(0)}})^2]\}$$

$$\left(\sigma^*_{f_i}\right)^2 = \left(\frac{2\cdot l_f}{f_{i.\max}-f_{i,\min}}\right)^2\cdot\left[\frac{\Delta^{\circ}_{f\,i}}{300}\cdot f_{i,\max}\right]^2 n_\alpha,\ (n_\alpha = 1),\ \Delta^{\circ}_{f_i} = \frac{\Delta_{f\,i}}{f_{i,\max}}\cdot 100\,\%,$$ тогда вместо (3.1)

получим:

$$\hat{M}_{t+1} = \arg\inf_{M\in D_M} E(M/D_P). \quad (2)$$

Апроксимуючі функції виду $Y_i^{(K+1)}\left(\vec{Y}_0\right)$в якості рішення будемо шукати методом стохастичної апроксимації на основі яружного методу сполучених градієнтів. Корекцію ваг зв'язків будемо здійснювати за наступною формулою (представлений рекурентний алгоритм навчання, відповідний методу стохастичної апроксимації, що забезпечує збіжність $w_{ij}^{(k)}(t) \xrightarrow[t\to\infty]{} \hat{w}_{ij}^{(k)}$ з ймовірністю $P=1$):

$$w_{ij}^{(k)}(t+1) = w_{ij}^{(k)}(t) + \mu(t)\{\eta_{ij}^{(k)}(t)r_{ij}^{(k)}(t) + \upsilon(t)\alpha_{ij}^{(k)}(t)[w_{ij}^{(k)}(t) - w_{ij}^{(k)}(t-1)]\} + \tilde{w}_{ij}^{(k)}(t+1) \quad (3)$$

**Результати експериментів**

Як приклад реалізації представленої методології ефективного стабільного (робастного) оцінювання параметрів моделей систем і процесів в формі учнів ШНМ була взята вибірка значень відповідних параметрів аеродинамічних характеристик осьового багатоступеневого компресору авіаційного двигуна. В якості вхідних даних для ШНМ були задані значення витрати повітря ($G_{air}$) і кількість обертів на хвилину ротора (n), як вихідних даних - ступінь стиснення ($\pi_c^*$) і коефіцієнт корисної дії ($\eta_c^*$) $\Delta^{\circ}_{\pi}$= 0,7%, $\Delta^{\circ}_{\eta}$= 0,1%. Середня відносна похибка апроксимації склала $\delta^{\circ}_{\pi}$= 0,625%.

**Висновки**

У доповіді проаналізовано методи побудови робастних нейромережевих моделей на основі використання багатошарової односпрямованої штучної нейронної мережі Застосування запропонованих розробок дозволяє отримувати стабільні (надійні) оцінки параметрів моделей нейронної мережі за попередньо невизначеними даними, що забезпечує синтез надійних мета-моделей.

ЛІТЕРАТУРА

1. Тархов Д.А. Нейронные сети. Модели и алгоритмы. Кн.18. Справочное издание. (Серия «Нейрокомпьютеры и их применение»): – М.: Радиотехника, 2005. – 256с.

2. Liao T. W., Chen L. J. Manufacturing Process Modeling and Optimization Based on Multi-Layer Perceptron Network // Journal of Manufacturing Science and Engineering. – 1998. – Vol. 120. – P. 109-119.
3. Залога В. А., Криворучко Д. В., Мишенин А. А. Выбор оптимальной структуры нейронной сети для решения задач теории резания// Резание и инструмент в технологических системах: Межд. научн. техн. сборник. –Х.: НТУ «ХПИ», 2002. –Вып. 63. –С. 65-71.
4. Хайкин С. Нейронные сети: полный курс, 2-е издание. : Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1104 с.: ил. – Парал. тит. англ.
5. Бэстен Д.-Э., ванн ден Берг В.-М., Вуд Д. Нейронные сети и финансовые рынки: принятие решений в торговых операциях. – Москва: ТВП, 1997.-236с.
6. Осовский С. Нейронные сети для обработки информации. – М.: Финансы и статистика, 2002. – 344 с.
7. Назаров А.В., Лоскутов А. И. Нейросетевые алгоритмы прогнозирования и оптимизации систем. – СПб: Наука и техника, 2003. – 384 с.
8. Семейкин В. Д., Скупченко А. В. Моделирование искусственных нейронных сетей в среде Matlab // Вестник Астраханского Государственного технического университета. Серия: управление, вычислительная техника и информатика. — 2009. — № 1. — С. 159–164.
9. Андреев В. О., Савиных Н. В. Интеллектуальные технологии, мягкие вычисления и программные средства их компьютерной реализации // Вестник компьютерных и информационных технологий. — 2006. — № 8. — С. 2–6.
10. Бурiченко М. Ю., Иванцев О. Б., Букреева О. В. Использование программного пакета Matlab для построения искусственных сетей нейронных сетей // Електроніка та системи управління. — 2011. — № 3 (29). — С. 120–123.
11. Koch, P. N., Wujek, B. A., Golovidov, O., and Simpson, T. W., 2002, “Facilitating Probabilistic Multidisciplinary Design Optimization Using Kriging Approximation Models,” Proceedings of the Ninth AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA, Sept. 4–6, AIAA 2002-5415.
12. Gano, S. E., Renaud, J. E., Martin, J. D., and Simpson, T. W., 2006, “Update Strategies for Kriging Models for Using in Variable Fidelity Optimization,” Struct. Multidiscip. Optim., 32(4), pp. 287–298.

**БАКУМЕНКО Ніна Станіславівна** – к. т. н., доцент; доцент кафедри теоретичної та прикладної системотехніки; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: n.bakumenko@karazin.ua; ORCID: 0000-0003-3496-7167.

Наукові інтереси:

*– інтелектуальний аналіз даних.*

**МЕНЯЙЛОВ Євген Сергійович** – старший викладач кафедри математичного моделювання та штучного інтелекту Національного аерокосмічного університету ім. М. Є. Жуковського «Харківський авіаційний інститут», вулиця Чкалова, 17, Харків - 00, Україна, 61000; e-mail: evgeniimenyailov@gmail.com; ORCID: 0000-0002-9440-8378.

Наукові інтереси:

*– математичне моделювання та штучний інтелект.*

**УГРЮМОВ Михайло Леонідович** – д. т. н., професор; професор кафедри теоретичної та прикладної системотехніки; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: ugryumov.mykhaylo52@gmail.com; ORCID: 0000-0003-0902-2735.

Наукові інтереси:

*– методи машинного навчання.*

**ЧЕРНИШ Сергій Вікторович** – аспірант кафедри математичного моделювання та штучного інтелекту Національного аерокосмічного університету ім. М. Є. Жуковського «Харківський авіаційний інститут», вулиця Чкалова, 17, Харків - 00, Україна, 61000; e-mail: mr.serhii.chernysh@gmail.com; ORCID: 0000-0002-1750-5158.

Наукові інтереси:

*– методи штучного інтелекту.*

**УДК 004.7**

БЕЛЫЙ Д. В., МОРОЗ О. Ю.

# МОДЕЛЬ КАМПУСНОЙ ЛОКАЛЬНОЙ КОМПЬЮТЕРНОЙ СЕТИ В АВТОМАТИЗИРОВАННОЙ СИСТЕМЕ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМ ПРОЦЕССОМ ПРОМЫШЛЕННОГО ПРЕДПРИЯТИЯ

**Введение**

Моделирование является основным методом исследования во всех областях знаний, а также является научно подтвержденным методом оценки характеристик сложных систем, которые, в свою очередь используются для разработки и реализации управленческих решений. В работе рассматриваются различные абстрактные уровни представления кампусной сети. Приведена объектно-ориентированная модель топологии кампусной сети для использования ее в автоматизированном проектировании ЛВС. В процессе анализа предметной среды кампусной сети рассматривались основные понятия, связанные с коммуникационным оборудованием [1].

**CAN** (кампусная сеть) — это группа локальных сетей, развернутых на компактной территории (кампусе) какого-либо учреждения и обслуживающие одно это учреждение - университет, промышленное предприятие, порт, оптовый склад и т.д. При этом сетевое оборудование (коммутаторы, маршрутизаторы) и среда передачи (оптическое волокно, медный завод, Cat5 кабели и др.) данных принадлежит арендатору или владельцу кампуса, предприятия, университета, правительства и так далее. Кампусные виды сетей получили широкое распространение в Соединенных Штатах Америки, отсюда и их название. Кампусные сети преимущественно развиты в колледжах и университетах. Зачастую они объединяют разнообразные здания, в том числе административные здания, учебные корпуса, библиотеки, общежития, гимназии и другие сооружения, такие как конференц-центры, технологические центры и прочие учебные заведения. Диапазон CAN составляет от 1 км до 5 км. Если два здания имеют один и тот же домен, и они связаны между собой сетью, то это будет рассматриваться только как CAN. Хоть и CAN в основном используется для корпоративных кампусов, канал передачи данных будет иметь высокую скорость.

**Постановка задачи**

Промышленному предприятию требуется модель кампусной локальной компьютерной сети для автоматизации системы управления технологическим процессом.

**Кампусная сеть** - это просто большая многосегментная локальная сеть на территории до нескольких километров в поперечнике, объединяющая локальные сети близко расположенных зданий.Конфигурацию устройств, на основе которых организуется данная кампусная сеть, а также пакет программного обеспечения для работы сети необходимо выбирать из расчета тех задач, которые и будут решаться информационным вычислительным комплексом [1].

**Система управления производственным предприятием**

Промышленное предприятие является тяжелой системой управления. Производственный процесс характеризуется производственно-техническим, организационным и экономическим единством. Кампусная локальная компьютерная сеть будет представлять из себя реализацию всех устройств, подключенных к сети. Например, компьютеры, аппаратные роутеры или какие-либо еще устройства, например сетевые принтеры или сканеры. Эти устройства ничем не отличаются от компьютеров, т.к. физически подключаются к сегменту управления по общим правилам и имеют внутри себя устройство, которое по всем параметрам аналогично сетевой плате.

**Суть разработанной модели кампусной локальной компьютерной сети**

Совокупность взаимосвязанных процессов в результате которых создается кампусная локальная компьютерная сеть, принято называть производственным процессом. Основой этой совокупности является технологический процесс (ТП). Технология процесса – это порядок выполнения различных видов деятельности по преобразованию входов в выходы.

Характеристики и особенности основных ТП предприятий определяют не только организацию соответствующих производственных процессов, а так же специфику предприятия в целом.

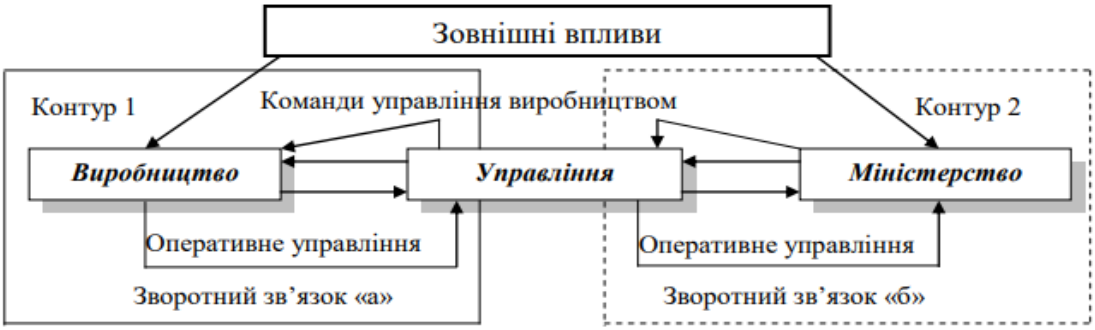



*Рис. 1 Схема модели управления предприятием*

Важной характеристикой технологического процесса является как размер процесса, что требует экономических ограничений, например, учёта затрат или источники формирования их по разным направлениям деятельности предприятия. Одновременно с этим нужно обеспечить всесторонний мониторинг самого процессного управления. В соответствии с назначением в Важной характеристикой является размер процесса, который требует экономических ограничений, среди которых, например, число центров учёта затрат или источники их формирования по разным направлениям деятельности предприятия. Количество таких центров и состав необходимых показателей должны быть небольшим, чтобы не усложнять процедуру учёта и определения результатов. Одновременно нужно обеспечить всесторонний мониторинг процессного управления. В соответствии с назначением в технологиях выделяют две группы показателей: для оценки эффективности процесса (показатели расхода ресурсов на единицу продукта или времени); для удовлетворённости результатами процесса.

Таким образом, предлагаемая модель оценки технологий позволяет проводить более комфортное пользование компьютерной сети

ЛІТЕРАТУРА

1. Мешков В.Е., Решение задачи синтеза топологии ЛВС на основе эволюционных методов проектирования // Мешков В.Е., Береза А.Н. Труды Международных научно-технических конференций «Интеллектуальные системы» (AIS”05) и «Интеллектуальные САПР» (CAD2005). Научное издание в 4-х томах.- М. ФИЗМАТЛИТ. 2005. Т4.- 256с.- ISBN 5-9221-0621-х.- с.120-126.
2. Л.В.Борисова. Автоматизовані системи управління технологічного процесу. – – Х.: НУЦЗУ, 2015.
3. Обзор современных Web - технологий – [Электронный ресурс] - http://www.sciteclibrary.ru/rus/catalog/pages/6643.html.

**БЕЛЫЙ Дмитрий Викторович –** студент группы КУ- 41 факультета компьютерных наук; Харьковский национальный университет имени В. Н. Каразина, площадь Свободы 6, Харьков, Украина, 61022; e-mail: dima.belyj2012@gmail.com; ORCID: 0000-0003-3888-8108.

Научные интересы:

*– Программирование. Управление проектами. Алгоритмы и структуры данных.*

**МОРОЗ Ольга Юрьевна –** старший преподаватель кафедры теоретической и прикладной системотехники факультета компьютерных наук; Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 6, Харьков, Украина, 61022; e-mail: o.moroz@karazin.ua; ORCID: 0000-0002-4920-4093.

Научные интересы:

*– Технологии автоматического проектирования параллельных программ.*

**УДК 004.4**

БІЛЬСЬКИЙ Г.М., ЛАБЕНКО Д.П.

# ІНТЕРАКТИВНА СИСТЕМА РОЗКЛАДУ УЧБОВИХ ЗАНЯТЬ ДЛЯ ЗАКЛАДУ ВИЩОЇ ОСВІТИ

**Вступ**

Розвиток методів навчання у закладах освіти, такі як університети та інститути, полягає, в основному, у полегшенні доступу до знань та розширення їх області. Прикладами таких інтеграцій є інтерактивні дошки, проектори, комп`ютерні класи. В основному вони використовуються рідко (але зараз можна бачити збільшення попиту та полегшення пристосування до роботи з системами) або для відповідних предметів, наприклад програмування або моделювання. Для гуманітарних наук в основному немає потреб для використання комп`ютерного класу.

Але фундаментальним також залишається полегшення самого навчального процесу як для студентів, так і викладачів та інших робітників цих закладів. Ціллю моєї розробки є не тільки привнести використання нових програмних й апаратних технологій, а також полегшити організацію розкладу занять та стандартизувати способи інформування студентів про події, що відбуваються навколо закладу та/або окремих предметів.

**Постановка задачі**

Поставлені вимоги до системи мають собою наступне:

- організація інформації про навчальний процес у доступному вигляді для перегляду та/або редагування, що включає в себе: назви предметів, ПІБ викладачів, їх професійний статус, розпорядок навчальних пар у семестрі, а також спеціальних пар, таких як підсумкові семестрові екзамени та захист курсових та дипломних робіт, номери аудиторій та їх відношення до корпусів закладу (якщо таких більше одного), тип пари (лекція або практика), найменування студентських груп та їх відношення до факультетів та інше;
- доступність описаної інформації у будь-який час за допомогою мережі Інтернет та спеціалізованих мобільних додатків з розширеним функціоналом без необхідності фізичної присутності у самому закладі освіти;
- захист від несанкціонованого доступу до будь-якої частини системи за допомогою розподілення ролей та системи аккаунтів;
- можливість комунікації зі студентами окремої групи, факультету, потоку, або окремим студентом через push-повідомлення у мобільному додатку;
- можливість розширення функціоналу системи у майбутньому;
- забезпечити роботу основних можливостей у режимі оф лайн (перегляд розкладу, тощо)

**Огляд системи**

Система складається з декількох програмних компонентів, що представляють собою клієнт-серверне ПЗ для використання користувачами-студентами, а також адміністраторами, та апаратного, що є сервером.

Основні частини:

- веб-сайт: основний інтерфейс користувача, де відбувається взаємодія з самим розкладом або з індивідуальними налаштуваннями, такими як належність до студентської групи та/або факультету та предметного курсу;
- мобільні додатки для iOS та Android: схожі до веб-сайту за функціоналом, але адаптовані для менших розмірів екранів, а також мають додаткові функції, серед яких – push-повідомлення;
- серверне ПЗ: представляє собою бібліотеки та виконані файли основної програми, задача якої – обробка даних, запитів від клієнтів, та робота з базою даних, що зберігає дану інформацію;

- база даних: сховище для даних про аккаунти користувачів, інформації, що відноситься до навчального процесу та службових даних, необхідних для роботи серверу, веб-сайту та додатків.

Для роботи з системою користувачам потрібно пройти процес створення аккаунту та/або авторизації у системі за допомогою вказаних на етапі реєстрації логіну та паролю. Для забезпечення розділення доступу до захищених ресурсів та надання розширеного функціоналу для певних категорій осіб (викладач, адміністратор і т.п.) використовується концепт ролей аккаунту. Таблиця розподілення ролей зазначена нижче.

*Таблиця.1 Розподілення ролей аккаунтів у залежності від статусу користувача*

| Статус користувача | Роль аккаунта |
|---|---|
| Студент | Студент (student) |
| Викладач/професор/доцент кафедри і т.п. | Викладач (teacher) |
| Завідуючий кафедри/ декан факультету/ деканат/ проректор/ ректор університету | Адміністратор (admin) |
| IT департамент університету/автор цього проекту | Спец-адміністратор (superadmin) |

**Функції системи**

Можливості роботи з системою частково відрізняються по ролям аккаунтів, але обов`язковим для кожної є наявність активного облікового запису, що може також бути прив`язаний до інших онлайн-систем закладу для спрощення процесу користування.

Основні функції для ролі “студент”:

- Початковий вибір та зміна факультету або групи (зміна відбувається через адміністратора після підтвердження з боку деканату або інших осіб управління закладу),
- Перегляд розкладу, інформації про навчальні предмети та викладачів, пошук по різним параметрам, як то назва предмету, день місяця, прізвище викладача тощо,
- Підписка на/отримання повідомлень (через мобільний додаток), наприклад про зміну розкладу пар, аудиторій, нагадування про майбутні модульні, залікові або екзаменаційні роботи,
- Зміна налаштувань аккаунта (email, пароль, відписка від повідомлень).

Функції для ролі “викладач”:

- Перегляд розкладу для своєї групи (якщо куратор), по всім, або тільки по своїм предметам, по тижням у груп, по предметам на кафедрі, що веде викладач тощо,
- Відправлення повідомлень студентам (зі своїм текстом) з можливістю вибору груп, або окремих студентів,
- Зміна налаштувань аккаунта

Функції ролі “адміністратор”:

- Додавання інформації про викладачів, предмети, факультети, аудиторії, групи,
- Редагування розкладу для викладачів по кафедрам, групам, тижням, можливість додавати спеціальні пари для модульних, залікових та екзаменаційних робіт,
- Встановлення відповідності викладачів до їх предметів та кафедр у системі.

Функції ролі “спец-адміністратор”:

- Всі функції ролі “адміністратор”,
- Управління аккаунтами студентів (в т.ч. видалення із системи)
- Редагування інформації про типи пар,
- Перегляд статистики роботи системи та обладнання, на якому система працює (діагностичних даних), наприклад для усунення неполадок,
- Прямий фізичний або віддалений доступ до обладнання та, або операційної системи, де працює ПО системи (серверна частина).

**Технічна складова системи**

Серверна частина:

- C# + .NET Core: крос-платформне рішення для запуску серверного ПО на будь-якій підтримуваній ОС (Windows, Linux, macOS). Для спрощення, основною платформою серверу вибрана Debian на базі Linux,
- PostgreSQL: надійна й популярна реляційна СУБД. У зв`язку з доступними API для .NET Core та простотою встановлення на цільову ОС є оптимальним вибором.

Клієнтська частина:

- ReactJS: клієнтська Javascript-бібліотека для роботи у веб-браузері ті підтримки створення динамчних додатків, дозволяє працювати з інтерфейсом і даними та набагато легше, ніж зі звичайним JS. Має багато доповнень у виді бібліотек, що розширюють функціонал, наприклад Material UI (додає новий дизайн та відповідні елементи інтерфейсу),
- C# + Xamarin: на цей раз універсальна мова програмування використовується разом з SDK для створення крос-платформних клієнтських додатків, що можуть працювати як на Windows, так і на iOS та Android. На початковому етапі заплановано створення Android-додатку. Ще один плюс для C# - спільна кодова база з відмінностями, що специфічні для конкретних задач сервера та клієнта,
- SQLite: простий та легкий варіант БД для використання на клієнті для зберігання кешованих даних та/або налаштувань. Цей вибір використовується тільки для Android, бо для браузерів існує стандарт WebSQL, що підтримується як API Javascript.

**Висновки**

Як показує практика, існуюча альтернатива, що на даний момент є в університеті імені В. Н. Каразіна, не є достатньо поширеною і використовується або для небагатьох предметів, або у деяких випадках не застосовується взагалі.

Створення представленої системи дозволить заощадити час та багато паперу, створити умови для пристосування до нових технологій не тільки студентів, а й працівників закладів вищої освіти. Завдяки можливостям по розширенню, у майбутньому можлива інтеграція інших сервісів, таких як консолідація домашніх та лабораторних завдань, лекцій та контролю за балами успішності.

ЛІТЕРАТУРА

1. .NET | Free, Cross-Platform, Open Source
   URL: https://dotnet.microsoft.com/
2. PostgreSQL: The World's Most Advanced Open Source Relational Database
   URL: https://www.postgresql.org/
3. React - A JavaScript library for building user interfaces
   URL: https://reactjs.org/

**БІЛЬСЬКИЙ Георгій Миколайович** – студент факультету комп`ютерних наук, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків, Україна, 61022; email – mastermangg@gmail.com; ORCID - 0000-0002-3475-5650.

Наукові інтереси:

- *Програмування та архітектура прикладного ПО та систем,*
- *Управління проектами,*
- *Відео- та 3D-дизайн.*

**ЛАБЕНКО Дмитро Петрович** – к.т.н., доцент кафедри ТПС, доцент; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, м. Харків, Україна, 61022; e-mail: d.labenko@karazin.ua.

Наукові інтереси:

- *Автоматизовані системи управління*

**УДК 539.3**

БОКОВ І.П., БОНДАРЕНКО Н.С., СТРЕЛЬНІКОВА О.О.

# ДОСЛІДЖЕННЯ ЛОКАЛЬНОГО НАПРУЖЕНО-ДЕФОРМОВАНОГО СТАНУ АНІЗОТРОПНИХ ПЛАСТИН НА БАЗІ УТОЧНЕНОЇ ТЕОРІЇ

**Вступ**

Широке застосування у сучасній техніці композитних матеріалів зі значною анізотропією пружних властивостей, викликає потребу у необхідності враховувати фізико-механічні характеристики цих матеріалів при проведенні розрахунків на міцність. При розрахунках тонкостінних елементів конструкцій з таких матеріалів доцільним є використання уточнених теорій пластин і оболонок. Теорія {m,n}-апроксимації, що використовує метод І.Н. Векуа розвинення невідомих функцій за поліномами Лежандра від нормальної координати [1], дозволяє врахувати зсувну піддатливість, характерну для більшості композиційних матеріалів.

Силові дії відносяться до одного з основних видів навантажень, яким піддаються об'єкти і вироби сучасної промисловості. Ці дії можуть носити як зосереджений, так і локальний характер. Тому дослідження напружено-деформованого стану анізотропних пластин за дії локальних силових навантажень є актуальним і важливим науково-технічним завданням.

**Аналіз досліджень і публікацій**

Про актуальність даної проблеми свідчать її аналітичні та чисельні дослідження, проведені в публікаціях [2-5]. В роботі [2] узагальнено дослідження із впливу локальних навантажень і контактних взаємодій елементів конструкцій, що призводять до концентрації напружень, появи зон пластичних деформацій і збільшення ризику руйнування при локалізованих впливах. У статті [3] досліджено оболонку, яка перебуває під дією локальних навантажень, що розподілені на малих площадках і змінюються на заданому короткому відтинку часу за лінійним законом. За допомогою теорії {m,n}-апроксимації, що ґрунтується на розвиненні шуканих функцій у ряди Фур'є за поліномами Лежандра від нормальної координати, досліджено напружено-деформований стан трансверсально-ізотропних пластин за дії зосереджених та локальних силових впливів у роботах [4] та [5] відповідно.

**Формулювання задачі**

Розглянуто анізотропну пластину товщини $2h$, що зазнає локального силового впливу в області, розмір якої набагато менший характерного розміру пластини. Краї пластини перебувають на значній відстані від місця прикладення силового навантаження. Це навантаження розподілено рівномірно і діє в нормальному до серединної площини пластини напрямку.

Одним з ефективних методів визначення локального напружено-деформованого стану тонкостінних конструкцій є метод фундаментальних розв'язків [6]. Він ґрунтується на використанні формули згортки, яка стосовно задач локального навантаження записується так:

$$\mathrm{P}(\vec{r}) = \iint_{\Omega} \mathrm{E}(\vec{r} - \vec{t})W(\vec{t})d\Omega ,$$

де P – внутрішні силові фактори; E – силові компоненти фундаментального розв'язку для анізотропної пластини, які отримані із використанням узагальненої теорії {m,n}-апроксимації; $\Omega$ – область локального навантаження; $\vec{r}$ і $\vec{t}$ – вектори поточної точки й точки інтегрування відповідно.

Досліджено вплив пружних сталих анізотропного матеріалу і геометрії області локального навантаження на внутрішні силові фактори. Виконані чисельні дослідження свідчать про те, що при проведенні розрахунків локального напружено-деформованого стану тонкостінних пластин істотно важливим є врахування пружних параметрів анізотропних матеріалів.

**Висновки**

В перспективі описана методика може бути використана при дослідженні локальних силових навантажень, що діють за довільним законом усередині довільної області. Результати, одержані в роботі, можуть застосовуватися для вивчення напружено-деформованого стану пластин на базі узагальненої теорії {m,n}-апроксимації у випадку утримання більшої кількості членів рядів розвинень шуканих функцій у напрямках, нормальних до площини ізотропії.

ЛІТЕРАТУРА

1. Пелех Б.Л., Лазько В.А. Слоистые анизотропные пластины и оболочки с концентраторами напряжений : Киев : Наук. думка, 1982. 296 с.
2. Гудрамович В.С., Даниев Ю.Ф., Пошивалов В.П. Нормирование прочности и надежности технических систем. *Техн. механика*. 2018. № 3. С. 112–120.
3. Луговой П.З., Сиренко В.Н., Скосаренко Ю.В., Батутина Т.Я. Динамика дискретно подкрепленной цилиндрической оболочки при действии локального импульсного нагружения. *Приклад. механика*. 2017. Вип. 53, № 2. С. 71–80.
4. Bokov I., Bondarenko N., Strelnikova E. Analysis of fundamental solutions to the equations of statics constructed for transversal-isotropic plates. *Eastern-European Journal of Enterprise Technologies*. 2015. Vol. 2, Issue-6. P. 56–62.
5. Боков І.П., Бондаренко Н.С., Стрельнікова О.О. Дослідження поведінки внутрішніх силових факторів в трансверсально-ізотропних пластинах за дії локальних навантажень. *Вісник Харківського національного університету ім. В.Н. Каразіна серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*. 2017. Вип. 35. С. 13–20.
6. Шевченко В.П. Методы фундаментальных решений в теории ортотропных оболочек. *Механика композитов* / под ред. А.Н. Гузя, А.С. Космодамианского. Киев, Т. 7, 1998. С. 159–196.

**БОКОВ Ігор Петрович** – к. ф-м. н.; Інститут проблем машинобудування ім. А. М. Підгорного НАН України, вул. Пожарського, 2/10, м. Харків, Україна, 61046; e-mail: igp.bokov@gmail.com; ORCID: 0000-0002-9138-4120.

Наукові інтереси:

*– математичні моделі в механіці деформованого твердого тіла.*

**БОНДАРЕНКО Наталія Сергіївна** – к. ф-м. н.; Інститут проблем машинобудування ім. А. М. Підгорного НАН України, вул. Пожарського, 2/10, м. Харків, Україна, 61046; e-mail: bondarenko.natalya.sergeevna@gmail.com; ORCID: 0000-0001-5254-5545.

Наукові інтереси:

*– математичні моделі в механіці деформованого твердого тіла.*

**СТРЕЛЬНІКОВА Олена Олександрівна** – д. т. н., професор; провідний науковий співробітник відділу гідроаеромеханіки енергетичних машин; Інститут проблем машинобудування ім. А. М. Підгорного НАН України, вул. Пожарського, 2/10, м. Харків, Україна, 61046; e-mail: estrel@ipmach.kharkov.ua; ORCID: 0000-0003-0707-7214.

Наукові інтереси:

*– чисельні методи в задачах механіки суцільного середовища, математичні моделі в механіці деформованого твердого тіла.*

**УДК 517.9**

БОМБА А. Я., МАЛАШ К. М.

# ОСОБЛИВОСТІ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ВПЛИВУ ВИБУХУ НА ДЕФОРМІВНЕ СЕРЕДОВИЩЕ З ЖОРСТКИМИ ВКЛЮЧЕННЯМИ МЕТОДАМИ КВАЗІКОНФОРМНИХ ВІДОБРАЖЕНЬ

**Вступ**

Зі стрімким розвитком сучасних технологій вибухові процеси знаходять усе більшого застосування у науці та техніці. Їх використання має місце у гірництві, будівництві підземних та напівпідземних споруд, нафто-добувній промисловості; також енергія вибуху застосовується для надання матеріалам необхідних інженерних властивостей [1].

Розширення царини використання вибухових процесів призвело до потреби у розробці нових підходів до їх дослідження; створення нових та модифікації існуючих математичних моделей впливу вибуху на середовище. Авторами написано низку робіт, у яких рідинна модель процесу вибуху, розроблена в роботах попередників [2], адаптована для випадку врахування взаємовпливу характеристик процесу вибуху на характеристики середовища. Метою досліджень авторів є визначення меж вирви, впресованої та незбуреної ділянок грунту, утворюваних у ізотропному чи анізотропному середовищі, внаслідок синхронного чи асинхронного вибуху одного чи двох зарядів. У даній роботі розглядаються способи моделювання утворюваних зон вирви, впресованої та незбуреної ділянок грунту за умови наявності у зоні враження жорстко закріпленого непроникного включення.

**Постановка задачі та математична модель**

У середовищі, де має відбутися вибух, виокремлюється для розгляду деяка трьохзв'язна область $G_z\ (z=x+iy)$, обмежена контуром заряду $L_* = \{z: f_*(x,y)=0\} = \{x+iy:\ \ x=x_*(t),\ y=y_*(t),\ \ \alpha_* < t < \beta_*\}$, контуром включення $L_0 = \{z: f_0(x,y)=0\} = \{x+iy:\ \ x=x_0(t),\ \ y=y_0(t),\ \alpha_0 < t < \beta_0\}$ та деяким зовнішнім контуром $L^* = \{z: f^*(x,y,\eta)=0\} = \{x+iy:\ \ x=x^*(t,\eta),\ y=y^*(t,\eta),\ \alpha^* < t < \beta^*\}$. Тут $x_*(t)$, $y_*(t)$, $x_0(t)$, $y_0(t)$, $x^*(t,\eta)$ та $y^*(t,\eta)$ - наперед задані неперервно-диференційовні функції, $\eta$ - параметр, що характеризує положення зовнішнього контура розглядуваної області, який буде уточнюватися в процесі розв'язування задачі за умови стабілізації, як описано у [3].

Процес руху частинок середовища, спричиненого вибуховою хвилею, моделюємо за допомогою закону руху $\vec{\upsilon} = k\ grad\,\varphi$ та рівняння нерозривності $div\,\vec{\upsilon} = 0$. Тут $\vec{\upsilon} = \left(\upsilon_x(x,y), \upsilon_y(x,y)\right)$ - швидкість частинок у точці $(x,y)$, $\varphi = \varphi(x,y) = -P/\rho$ - потенціал поля, утворюваного вибухом, де $\rho$ – густина середовища, $P$ – імпульс тиску, що чинить заряд на частинки середовища під час вибуху, $k = k(|grad\,\varphi|)$ - так званий, фіктивний коефіцієнт проникності середовища, котрий характеризує здатність частинок відриватися [4]. З огляду на наявність зворотнього впливу характкристик вибухового процесу на характеристики середовища, він ідентифікуватиметься у процесі розв'язування задачі за формулою $k = k_0 + \frac{1}{2}\beta\ \left(I - I^*\right)\left(\left(I - I^0\right) + \left|I - I^0\right|\right)$, де $k_0 = k^0$ - початковий розподіл фіктивного коефіцієнта проникності середовища, $I = \sqrt{\varphi_x^2 + \varphi_y^2} = |grad\,\varphi|$ - величина градієнта квазіпотенціалу, $I^0$, $I^*$ - її критичні значення, що використовуються для встановлення меж розділу утворюваних ділянок вирви, впресованої та незбуреної зон грунту, параметр $\beta$ залежить від типу грунту та визначається експериментально.

Аналогічно до [5] вводимо функцію $\psi = \psi(x, y)$, комплексно спряжену до $\varphi = \varphi(x, y)$, яка описуватиме лінії течії. Опишемо два можливих шляхи розв'язування задачі з використанням числових методів квазіконформних відображень. Оскільки область $G_z^*$ непроникна, одна з ліній течії (позначимо її $\psi(x, y) = \psi^*$) її «омиватиме», роздвоюючись у деякій критичній точці $K$ ($K \in L_0$) (значення потенціалу у ній позначимо $\underline{\varphi} = \varphi(K)$) ($\varphi_* < \underline{\varphi} < \varphi^*$) та знову з'єднуючись у іншій критичній точці $M$ ($M \in L_0$) (значення потенціалу у ній аналогічно позначимо через $\overline{\varphi} = \varphi(M)$ ($\varphi_* < \underline{\varphi} < \overline{\varphi} < \varphi^*$)).

**Спосіб 1.** Утворюємо умовний розрізу $\Gamma$ області $G_z$, що проходить через певну фіксовану точку $A \in L_*$ вздовж однієї з ліній течії (яка шукатиметься в процесі розв'язування задачі), та приходимо до задачі на квазіконформне відображення $\omega = \omega(z) = \varphi(x, y) + \mathrm{i}\psi(x, y)$ області $\tilde{G}_z = G_z \setminus \Gamma$ на відповідну їй область квазікомплексного потенціалу $G_\omega = \{\omega = \varphi + i\psi : \varphi_* < \varphi < \varphi^*, \ 0 < \psi < Q\} \setminus \{\omega : \psi = \psi^*, \underline{\varphi} < \varphi < \overline{\varphi}\}$ (Рис. 1), яка зводиться до розв'язання системи рівнянь типу Коші-Рімана

$$\begin{aligned} \kappa(|\,grad\varphi\,|)\frac{\partial\varphi}{\partial x} &= \frac{\partial\psi}{\partial y}, \\ \kappa(|\,grad\varphi\,|)\frac{\partial\varphi}{\partial y} &= -\frac{\partial\psi}{\partial x}, \end{aligned} \quad (x, y) \in \tilde{G}_z, \tag{1}$$

при крайових умовах $\varphi|_{L_*} = \varphi_*$, $\varphi|_{L^*} = \varphi^*$, $\psi|_{AD} = 0$, $\psi|_{BC} = Q = \int_{L_*} -\upsilon_y dx + \upsilon_x dy$,

$\psi|_{\underline{HKME}} = \psi|_{\overline{HKME}} = \psi_*$. Значення $\underline{\varphi}$, $\overline{\varphi}$, $\psi^*$, $Q$ та $\eta$ ідентифікується ітераційно у процесі розв'язування задачі.

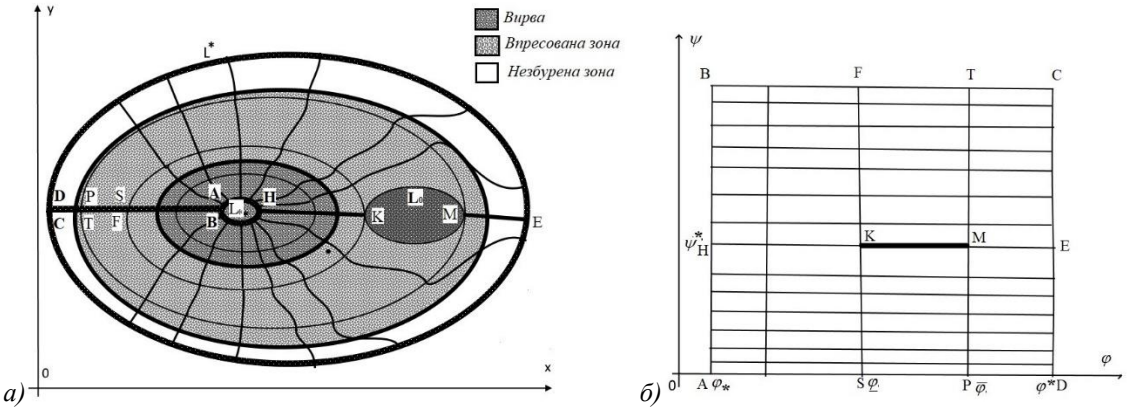



*Рис. 1 Схематичне зображення а)фізичної області та б)відповідної їй області квазікомплексного потенціалу за умови утворення умовного розрізу вздовж довільної лінії течії*

**Спосіб 2.** Умовний розріз $\Gamma$ області $G_z$, проводимо через лінію роздвоєння течії $\psi(x, y) = \psi^*$, яка шукатиметься у процесі розв'язування задачі (Рис. 2) та отримуємо задачу на квазіконформне $\omega = \omega(z) = \varphi(x, y) + \mathrm{i}\psi(x, y)$ відображення області $\tilde{G}_z = G_z \setminus \Gamma$ на область квазікомплексного потенціалу $G_\omega = \{\omega = \varphi + i\psi : \varphi_* < \varphi < \varphi^* \ 0 < \psi < Q\}$. Задача зводиться до розв'язування системи рівнянь (1) при крайових умовах $\varphi|_{L_*} = \varphi_*$, $\varphi|_{L^*} = \varphi^*$, $\psi|_{AD} = 0$, $\psi_* = \psi|_{BC} = Q = \int_{L_*} -\upsilon_y dx + \upsilon_x dy$. Значення $\underline{\varphi}$, $\overline{\varphi}$, $\psi^* = Q$ та $\eta$, а також точне положення точки $A = B$ ідентифікується ітераційно у процесі розв'язування задачі.

У обох описаних способах переходимо до розв'язування оберненої задачі та розв'язуємо її як описано, наприклад, у [4,5].

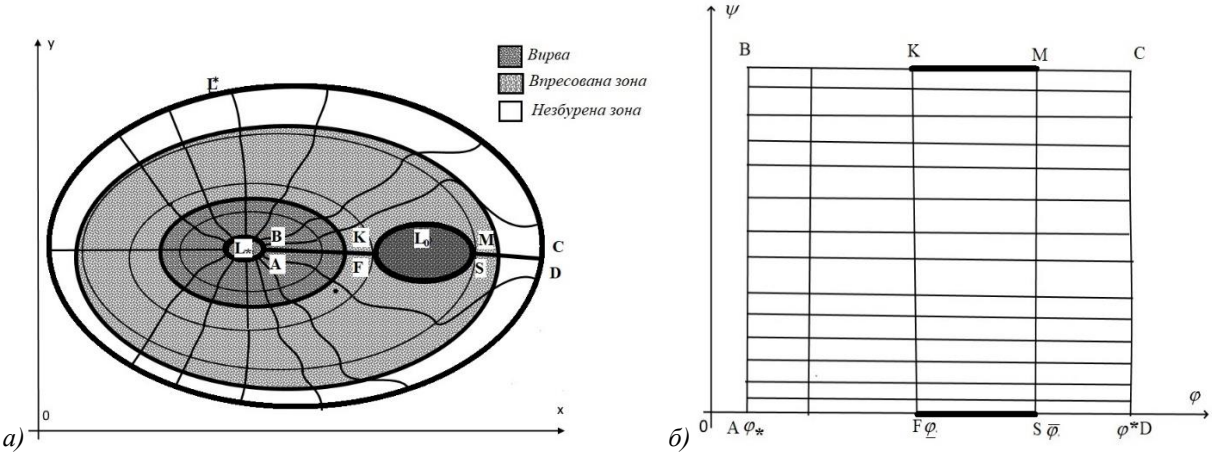



*Рис. 2 Схематичне зображення а)фізичної області та б)відповідної їй області квазікомплексного потенціалу за умови утворення умовного розрізу вздовж лінії розділу течії*

**Висновки**

У результаті розв'язування описаної задаі отримуємо розподіл зон вирви, впресованої та незбуреної ділянок грунту, що утворюються у результаті вибуху. Обидва описані способи мають певні складності у реалізації, що зумовлено необхідністю додаткової ідентифікації лінії роздвоєння течії $\psi(x,y)=\psi^*$ у першому випадку та положення точки $A=B$ - у другому.

ЛІТЕРАТУРА

1. Замышляев Б. В., Евтерев Л. С. Модели динамического деформирования и разрушения грунтовых сред. Москва: Наука, 1990. 215 с.
2. Булавацкий В. М., Лучко И. А. Некоторые обратные задачи импульсно-гидродинамической теории взрыва на выброс. *Исследования по краевым задачам гидродинамики и теплофизики*. Киев, 1979. С. 53-64.
3. Бомба А. Я., Малаш К. М. Моделювання вибухових процесів в анізотропному середовищі з ідентифікацією межі зони впливу. *Математичне та комп'ютерне моделювання. Серія «Технічні науки».* Кам'янець-Подільський: КДПУБ, 2018. Вип. 18. С. 5 – 18.
4. Бомба А. Я., Каштан С. С., Пригорницький Д. О., Ярощак С. В. Методи комплексного аналізу: монографія. Рівне : НУВГП, 2013. 415 с.
5. Bomba A., Malash K. Modeling the formation of craters caused by the two charges explosion using quasiconformal mappings numerical methods. *Advanced Computer Information Technologies: 9th International Conference Proceedings*, Ceske Budejovice, Czech Republic, 5-7 June, 2019. Ceske Budejovice, Czech Republic, 2019. P.: 113 – 116.

**БОМБА Андрій Ярославович** – д. т. н., професор; професор кафедри комп'ютерних наук та прикладної математики; Національний університет водного господарства та природокористування, м. Рівне, вул. Соборна, 11, 33000; e-mail: abomba@ukr.net; ORCID: 0000-0001-5528-4192.

Наукові інтереси:

*– математичне моделювання, квазіконформні відображення, обернені задачі.*

**МАЛАШ Катерина Миколаївна** – аспірант кафедри інформатики та прикладної математики; Рівненський державний гуманітарний університет, м. Рівне, вул. Пластова, 31, Україна, 33000; e-mail: katemalash@gmail.com; ORCID: 0000-0003-4771-9349.

Наукові інтереси:

*– математичне моделювання, квазіконформні відображення, вибухові процеси.*

**УДК 681.3.06**

БОНДАРЕНКО В.А.

# ЭКСПЕРТНАЯ СИСТЕМА КАК СРЕДСТВО ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

**Назначение экспертных систем**

В начале восьмидесятых годов в исследованиях по искусственному интеллекту сформировалось самостоятельное направление, получившее название "экспертные системы" (ЭС). Цель исследований по ЭС состоит в разработке программ, которые при решении задач, трудных для эксперта-человека, получают результаты, не уступающие по качеству и эффективности решениям, получаемым экспертом. Исследователи в области ЭС для названия своей дисциплины часто используют также термин "инженерия знаний", введенный Е. Фейгенбаумом как "привнесение принципов и инструментария исследований из области искусственного интеллекта в решение трудных прикладных проблем, требующих знаний экспертов".

Программные средства, базирующиеся на технологии экспертных систем, или инженерии знаний (обычно используют как синонимы), получили значительное распространение в мире. Важность экспертных систем состоит в следующем:

– технология экспертных систем существенно расширяет круг практически значимых задач, решаемых на компьютерах, решение которых приносит значительный экономический эффект;

– технология ЭС является важнейшим средством в решении глобальных проблем традиционного программирования: длительность и, следовательно, высокая стоимость разработки сложных приложений;

– высокая стоимость сопровождения сложных систем, которая часто в несколько раз превосходит стоимость их разработки; низкий уровень повторной используемости программ и т.п.;

– объединение технологии ЭС с технологией традиционного программирования добавляет новые качества к программным продуктам за счет: обеспечения динамичной модификации приложений пользователем, а не программистом; большей "прозрачности" приложения; лучшей графики; интерфейса и взаимодействия [1].

В работах по искусственному интеллекту под экспертной системой понимается система, объединяющая возможности компьютера со знаниями и опытом эксперта в такой форме, что система может предложить разумный совет или осуществить разумное решение поставленной задачи. Типовая структура экспертной системы приведена на рис. 1.

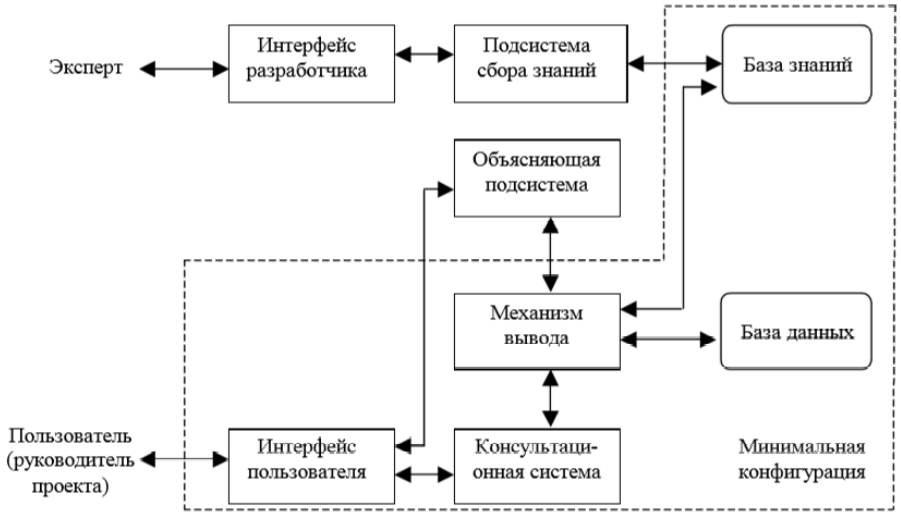



*Рис. 1. Структура экспертной системы*

ЭС предназначены для так называемых неформализованных задач, т.е. ЭС не отвергают и не заменяют традиционного подхода к разработке программ, ориентированного на решение формализованных задач.

Неформализованные задачи обычно обладают следующими особенностями:

– ошибочностью, неоднозначностью, неполнотой и противоречивостью исходных данных;

– ошибочностью, неоднозначностью, неполнотой и противоречивостью знаний о проблемной области и решаемой задаче;

– большой размерностью пространства решения, т.е. перебор при поиске решения весьма велик;

– динамически изменяющимися данными и знаниями [2].

**Средства создания экспертных систем**

Раньше создания и проектирования экспертной системы требовало больших затрат в плане времени (до нескольких лет) и людей (несколько десятков). В настоящее время имеется ряд средств, ускоряющих процесс создания. Эти средства называются инструментальными или инструментарием.

По своему назначению и функциональным возможностям инструментальные программы, которые применяются при проектировании экспертных систем, можно разделить на четыре достаточно большие категории:

1) оболочки экспертных систем;
2) языки программирования высокого уровня;
3) среда программирования, поддерживающая несколько парадигм;
4) дополнительные модули.

Если рассматривать инструментальные средства создания экспертных систем, то наиболее популярными языками программирования являются LISP и PROLOG, а среди оболочек экспертных систем KEE, CENTAUR, G2 и GDA, AT_ТЕХНОЛОГИЯ, которые предоставляют в распоряжение разработчика широкий набор для комбинирования систем представления знаний, языков программирования, объектов и процедур [3].

**Использование ЭС для формировки базы знаний и правил**

Опыт разработки и использования экспертных систем, в том числе для диагностики неисправностей сложных технологических объектов, позволяет утверждать, что при разработке экспертной системы следует придерживаться принципа открытости, что подразумевает возможность внесения изменений в систему в процессе ее эксплуатации. Это предполагает возможность корректировки правил базы знаний. Наиболее открытой для внесения изменений является система, в которой знания представлены в виде совокупности правил и фактов и в которой знания отделены от программного кода, реализующего механизм вывода.

Правила – это утверждения вида:

$$A \leftarrow B_1, B_2 \ldots B_n,$$

где $n \geq 0$, A – заголовок правила, последовательность $B_i$ – тело правила, причем элементы $B_i$ могут представлять собой как факты, так и правила.

На естественном языке правила в общем виде могут быть представлены так:

*ПравилоN: Если*
*Объект1 = Значение1, КД1=к1 Объект2 = Значение2, КД=к2*
. . .
*ОбъектJ = ЗначениеJ, КД=кJ*
*То*
*Объект3 = Значение3, КД=к3*

где Правило, Если, То и КД – ключевые слова, используемые при записи правил; Объект и Значение – соответственно объект из предметной области и его значение, КД – коэффициент достоверности (степень уверенности), дробное число из диапазона [0,1] соответствующее степени уверенности, что состояние объекта характеризуется указанным значением. Значения

коэффициента достоверности используется для вычисления по формуле Байеса коэффициента достоверности заключения.

При формировании базы знаний необходимо учитывать, что во время консультации, в процессе согласования фактов с правилами, механизм вывода выбирает правила из базы знаний в том порядке, в котором они находятся в базе знаний, начиная с первого. Поэтому база знаний должна начинаться с правил, описывающих наиболее вероятные результаты консультации.

Пусть $P_i$– вероятность получения i-го заключения, соответствующего правилу с номером i. Тогда правила в базе знаний должны быть расположены так, чтобы выполнялось условие [4]:

$$P_1 \geq P_2 \geq \ldots \geq P_i \geq \ldots \geq P_k.$$

**Выводы**

Повысить качество принимаемых решений можно за счет интеграции в автоматизированное рабочее место руководителя проекта интеллектуального компонента – экспертной системы. Для представлений знаний (алгоритмов принятия решений) в экспертной системе следует использовать правила, как наиболее открытый для внесения изменений способ представления знаний. Учесть неопределенность фактов и вероятностный характер заключений можно при помощи коэффициента достоверности. Время консультации зависит от порядка следования правил в базе знаний экспертной системы.

ЛИТЕРАТУРА

1. Татжибаева О.А. Разработка экспертных систем: методические указания к расчетно-графическим работам по дисциплине "Системы искусственного интеллекта". – Оренбург: ГОУ ОГУ, 2005. – 23с.
2. Нейлор К. Как построить свою экспертную систему. — М.: Энергоатомиздат, 1991. — 286 с.
3. Иструментальные средства разработки и проектирования экспертных систем [Электронный ресурс]: Портал искусственного интеллекта 2009-2017. URL: http://www.aiportal.ru/articles/expert-systems/design-tools.html (Дата обращения 14.04.2020).
4. Культин Н.Б. Архитектура системы управления проектами. // *Инновции в науке, образовании и производстве. Сборник научных трудов*. – СПб.: из-во Политехн. ун-та, 2007. – 263с.

**БОНДАРЕНКО Владислав Алексеевич –** студент, Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 4, Харьков-22, Украина, 61022; e-mail: vladbond16@gmail.com; ORCID: 0000-0003-2158-7608.

Научные интересы:

- *моделирование информационных процессов в сложных и распределенных системах.*

**УДК 519.688;515.127;519.246.(25+27)**

БРАТЧЕНКО М.І., ДЮЛЬДЯ С.В.

# АЛГОРИТМИ ФУР'Є-СИНТЕЗУ МОДЕЛЕЙ МАТЕРІАЛІВ З ОБМЕЖЕНИМ СПЕКТРОМ ФРАКТАЛЬНОЇ ПОРИСТОСТІ

Моделі структурно складних середовищ дуже різних масштабів (від полімерів, гелів, дисперсних і пористих матеріалів до геологічних утворень земної кори і далі аж до структур розподілу матерії у Всесвіті) за відсутності даних про будь-яке їх впорядкування спираються на їх представлення визначеним у просторі $\mathbf{r}\in\Re^E$ евклідової розмірності $E$ і часі $t$ випадковим скалярним полем $\rho(\mathbf{r},t)$ густини $\rho$ із заданими 1-точковими та кореляційними властивостями. Так, 1-точковий опис статистично однорідного стаціонарного поля $\rho$ надають визначені на напівосі $\rho\ge 0$ (густина є невід'ємною) не залежні від $\mathbf{r},t$ кумулятивна функція розподілу (c.d.f.) $F_1(\rho)=\Pr(\rho'<\rho)=\int_0^\rho f_1(\rho')\mathrm{d}\rho'$ та густина ймовірності (p.d.f.) $f_1(\rho)=\mathrm{d}F_1(\rho)/\mathrm{d}\rho$ зі скінченними моментами $\langle\rho^{1,2,\dots}\rangle<+\infty$, зокрема, середньою густиною $\overline{\rho}=\langle\rho\rangle$ і центрованою дисперсією $\sigma^2=\langle(\rho-\overline{\rho})^2\rangle=\langle\rho^2\rangle-\overline{\rho}^2$, де $\langle\circ\rangle=\int_0^\infty(\circ)\mathrm{d}F_1=\int_0^\infty(\circ)f_1(\rho)\mathrm{d}\rho$ позначає ансамблеве осереднення. Коли спектр $\rho$ неперервний (атмосфери Землі і планет, хмари або гелі), 1-точкові p.d.f. загалом тяжіють до гаусівськості ($f_1(\rho)\to\mathcal{N}(\rho|\,\overline{\rho},\sigma^2)$) з вичерпним описом двома першими моментами. Моделі з дискретними p.d.f. описують пористі матеріали [1]; найпоширенішою з них є бінарна. У ній поле $\rho(\mathbf{r})=\rho_\mathrm{M}\cdot\Theta[\rho(\mathbf{r})]$ визначає фазова функція Хевісайда $\Theta$, яка приймає значення 0 усередині пор і 1 у матриці з парціальною густиною $\rho_\mathrm{M}$. Відносна об'ємна пористість $\omega\subset[0,1)$ надає перші моменти $\overline{\rho}=\rho_\mathrm{M}\cdot(1-\omega)$ і $\sigma^2=\rho_\mathrm{M}^2\cdot\omega\cdot(1-\omega)$ та $k$-ті вищі моменти $\langle(\rho-\overline{\rho})^{k>2}\rangle\ne 0$.

Структурне різноманіття невпорядкованих середовищ й за близьких 1-точкових p.d.f. визначається відмінністю їх $n$-точкових кореляційних співвідношень, що їх описують функції автоковаріації $C(\mathbf{r}_1,\mathbf{r}_2,\dots,\mathbf{r}_n)$, $n\to\infty$. Головною з них є парна автоковаріаційна функція $C(\mathbf{r}_1,\mathbf{r}_2)$, яка у припущенні статистичної однорідності поля залежить від масштабної змінної $\mathbf{l}=\mathbf{r}_2-\mathbf{r}_1$:

$$C(\mathbf{l})\equiv\sigma^2\cdot R(\mathbf{l})=\big\langle\big(\rho(\mathbf{r})-\overline{\rho}\big)\cdot\big(\rho(\mathbf{r}+\mathbf{l})-\overline{\rho}\big)\big\rangle=\big\langle\rho(\mathbf{r})\cdot\rho(\mathbf{r}+\mathbf{l})\big\rangle-\overline{\rho}^2,\qquad(1)$$

де $R(\mathbf{l})=C(\mathbf{l})/\sigma^2$ — нормована на дисперсію автокореляційна функція. Еквівалентний опис парних кореляцій надає вимірювана в експериментах структурна функція (напів-варіограма) $S$:

$$S(\mathbf{l})=\Big\langle\big|\rho(\mathbf{r})-\rho(\mathbf{r}+\mathbf{l})\big|^2\Big\rangle=2\big\langle\rho^2(\mathbf{r})\big\rangle-2\big\langle\rho(\mathbf{r})\cdot\rho(\mathbf{r}+\mathbf{l})\big\rangle=2\sigma^2\cdot\big[1-\sigma^{-2}C(\mathbf{l})\big]=2\sigma^2\cdot\big[1-R(\mathbf{l})\big].\quad(2)$$

Загалом анізотропні ($\mathbf{l}$ є $E$-вектором) $C$, $R$ і $S$ за ізотропії поля залежать лише від відстані $l=\|\mathbf{l}\|$.

Іншою міркою кореляцій є спектр потужності $P(\mathbf{q})=\big|\widetilde{\rho}(\mathbf{q})\big|^2=\Big|\int_{-\infty}^{\infty}\mathrm{d}^E\mathbf{r}\,\big(\rho(\mathbf{r})-\overline{\rho}\big)e^{-i2\pi\mathbf{qr}}\Big|^2$, де $\widetilde{\rho}(\mathbf{q})$ — фур'є-образ поля густини $\rho(\mathbf{r})$, $\mathbf{q}$ — хвильовий вектор. По теоремі Вінера-Хінчина $C(\mathbf{l})=\int_{-\infty}^{\infty}\mathrm{d}^E\mathbf{q}\,P(\mathbf{q})e^{i2\pi\mathbf{ql}}$. У разі ізотропії інтегруючи по $E$-вимірній одиничній гіперсфері $\Omega_E$ отримуємо $C(l)=2\pi l^{1-E/2}\int_0^\infty\mathrm{d}q\cdot q^{E/2}J_{E/2-1}(2\pi ql)P(q)$, де $J_\nu(x)$ — функція Бесселя $\nu$-го порядку.

Засоби моделювання природних середовищ добре розвинуті для гаусівської $f_1$ та у експоненційних $C(l)\propto\exp(-l/l_\mathrm{c})$ або гаусівських $C(l)\propto\exp[-(l/l_\mathrm{c})^2]$ моделях кореляторів, що слушно відтворюють придушення кореляцій на відстанях $l>>l_\mathrm{c}$ — кореляційної довжини поля. Та бачимо, що для моделювання випадкового поля $\rho(\mathbf{r})$ можна виходити не лише з явного вигляду модельних кореляторів, а і з наперед заданого модельного спектру потужності сигналу.

Надалі ми застосуємо його до спектрального синтезу фрактальних [2] полів $\rho(\mathbf{r})$ зі структурною функцією $S(l)\propto l^{2H}$ і спектром потужності $P(q)\propto q^{-\beta}$. Тут $H\subset[0,1]$ — пов'язаний з фрактальною розмірністю $D=E-H$ поля індекс Херста, а $\beta=2H+E=3E-2D\ge 0$. Зазначимо, що однорідні по масштабу $l$ ступеневі функції $S(l)$ і $P(q)$ містять лише безрозмірні показники

$H(D,E)$ і $\beta(D,E)$, тож фрактальна структура не має ніяких характерних розмірів! Це дозволяє, по-перше, побудувати ефективну малопараметричну модель структурно складного середовища та, по-друге, розрізняти матеріали за приналежністю до класу універсальності [2].

Однак практична реалізація цього підходу у конструктивних алгоритмах стикається з математичними і технічними складнощами. Коротко окреслимо їх та опишемо відомі та знайдені й імплементовані нами у алгоритмах шляхи їх подолання з кінцевою метою генерації випадкового поля густини $\rho(\mathbf{r})$ із заданими $\overline{\rho}$, $\sigma^2$ (чи $\omega$ для бінарно-пористого матеріалу) та індексу Херста $H = H_{SF}$ структурної функції $S(l)$.

**1.** Невід'ємність густини і негаусівськість 1-точкового розподілу. Нормальна p.d.f. $N(\rho|\overline{\rho},\sigma^2)$ визначена на осі й надає нефізичну ненульову ймовірність від'ємної $\rho$. Тож маємо звертатися до визначених лише на $\rho \geq 0$ негаусівських розподілів. Найпростішим є широко вживаний у стохастичній демографічній та фінансовій математиці логнормальний розподіл $LN(\rho|\ldots) = N(z = \ln(\rho/\overline{\rho})|\ldots)$. Для нього $\rho = \overline{\rho}\cdot e^{z} \geq 0$. Обираючи $z$ з $LN(z|0,1)$, за заданих $\overline{\rho}$ і $\sigma^2$ отримуємо випадкову $\rho = \overline{\rho}\cdot(1+\varepsilon^2)^{-½}\cdot\exp[\ln z \times \ln^{½}|1+\varepsilon^2|]$, де $\varepsilon = \sigma/\overline{\rho}$. Гамма-розподіл $\Gamma(z;s,\theta)$ з параметрами $s > 0$, $\theta > 0$ має p.d.f. $f_\Gamma(z) = \theta^{-s}\Gamma(s)\cdot z^{s-1}e^{-z/\theta}$ і c.d.f. $F_\Gamma(z) = \Gamma^{-1}(s)\cdot\gamma(s,z/\theta)$, де $\Gamma(x)$ і $\gamma(s,x) = \int_0^x t^{s-1}e^{-t}\mathrm{d}t$ — повна і нижня неповна гамма-функції. Він теж гарантує невід'ємність $z$, і розігравши $z$ з розподілу $\Gamma(z;\varepsilon^{-2},1)$, просто масштабуємо його до $\rho = \overline{\rho}\cdot\varepsilon^2\times z$.

**2.** Проблема генерації негаусівського поля із заданим 2-точковим корелятором $C(l)$ набагато складніша за таку у добре вивченому випадку нормальних 1-точкових p.d.f. [3], бо вимагає нелінійного відображання пробних гаусівських полів, яке викривляє їх корелятор. Для компенсації доводиться застосовувати обчислювально витратні ітеративні алгоритми [4]. Ми ж скористалися альтернативою з роботи [5], де було запропоновано застосовувати з цією метою стандартний для техніки методів Монте-Карло набагато ефективніший метод зворотних c.d.f. За ним генерація негаусівського поля $Z(\mathbf{r})$ з 1-точковою c.d.f. $F_{NG}(z(\mathbf{r}))$ і кореляторами (1–2) складається з розігрування [3,4] пробного поля $X(\mathbf{r})$ з $N(x|0,1)$ та потрібною $C(l)$ з наступним застосуванням до його реалізацій $x$ перетворення [5] $Z(\mathbf{r}) = \mathrm{inv}\{F_{NG}(F_G(x(\mathbf{r})))\}$, де $\mathrm{inv}\{\ldots\}$ — оператор обернення функції, $F_G(x) = \frac{1}{2}\cdot\left(1+\mathrm{erf}\,\frac{x}{\sqrt{2}}\right)$ — c.d.f. $N(x|0,1)$, а $\mathrm{erf}\,x = \frac{2}{\sqrt{\pi}}\int_0^\infty e^{-t^2}\mathrm{d}t$ — функція похибок. Зворотну до c.d.f. $F_{NG}(z(\mathbf{r}))$ шуканого негаусівського поля функцію неважко обчислити алгоритмами знаходження коренів трансцендентних рівнянь. Числові експерименти (як роботи [5], так і наші власні) показали, що і для логнормального, і для гамма-розподілу цей алгоритм практично не змінює кореляційні функції породжуючого гаусівського поля.

**3.** Обмеженість спектру потужності фрактального випадкового поля. У класичній теорії фракталів [2] аксіоматично вважається, що однорідність функцій $S$ і $P$ розповсюджується на усі масштаби $l$ і $q$. Однак такий математичний фрактал володіє нескінченною енергією на $q = 0$ (інфрачервона катастрофа) і тому ніколи не втілюється фізично. Природні середовища повинні мати обмежену з обох боків область фрактальності, лише у якій $S(l)$ і $P(q)$ змінюються за ступеневими законами. Тож для генерації матеріалів з фрактальністю у обмеженому діапазоні масштабів, маємо вести спектральний синтез з обмеженим Θ-функціями спектром потужності

$$P(q) = \sigma^2\cdot\frac{H_{PS}\cdot\Gamma\left(\frac{E}{2}\right)}{\pi^{\frac{E}{2}}\cdot\left(q_{\min}^{-2H_{PS}} - q_{\max}^{-2H_{PS}}\right)}\cdot\frac{1}{q^{\beta}}\cdot\Theta(q-q_{\min})\cdot\Theta(q_{\max}-q), \qquad (3)$$

де $\beta = 2H_{PS}+E \geq 0$, $H_{PS}$ — індекс Херста спектру потужності, $q_{\min,\max} = 1/(2\pi\cdot l_{\max,\min})$ — граничні хвильові числа. Вихідний параметр $l_{\max} \sim l_c$ є масштабом розчеплення кореляцій, на якому поле поступово втрачає фрактальність і схильне до статистичної 'самовідтворюваності' (це й запобігає розбіжності його повної енергії); найменший масштаб $l_{\min}$ фрактальності фізично відповідає (наприклад, атомній) впорядкованості складових середовища на мікрорівні.

Проблема полягає у тому, що $H_{SF} = H_{PS}$ лише при $q_{\min} \to 0$ і $q_{\max} \to +\infty$. За обмеженості спектру між $H_{SF}$ і $H_{PS}$ виникає систематичний зсув, який слід враховувати для фур'є-синтезу поля з бажаними $H_{SF}$, $l_{\max}$ і $l_{\min}$. Залежність $H_{PS}(H_{SF})$ вивчалася в роботі [6] для 1D випадкових процесів (часових рядів). Ми узагальнили підхід [6] на випадок фрактальних випадкових полів довільної $E$. Для них структурна функція поля зі спектром потужності (3) приймає вигляд:

$$S(l)=2\sigma^2\,\frac{\Delta(2\pi q_{\max}l,H_{\rm PS},E)-\Delta(2\pi q_{\min}l,H_{\rm PS},E)}{(2\pi q_{\min}l)^{-2H_{\rm PS}}-(2\pi q_{\max}l)^{-2H_{\rm PS}}},\ \Delta(z,H,E)=z^{-2H}\cdot\left[{}_1F_2\left(-H;1-H,\frac{E}{2};-\frac{z^2}{4}\right)-1\right],$$

де ${}_1F_2(a_1;b_1,b_2;z)$ — узагальнена гіпергеометрична функція. З асимптотичного аналізу $S(l)$ за різних $q_{\min,\max}\cdot l$ випливає, що на найрепрезентативнішому масштабі $l_0=(l_{\min}\cdot l_{\max})^{1/2}$ фрактальності

$$H_{\rm SF}=\frac{l_0}{2S(l_0)}\cdot\left.\frac{{\rm d}S(l)}{{\rm d}l}\right|_{l=l_0}=\frac{H_{\rm PS}-p^{\frac{1}{2H_{\rm PS}}}(H_{\rm PS},E)\cdot\delta^{1-H_{\rm PS}}}{1-p^{\frac{1}{2H_{\rm PS}}}(H_{\rm PS},E)\cdot\delta^{1-H_{\rm PS}}-p^{\frac{1}{2(1-H_{\rm PS})}}(H_{\rm PS},E)\cdot\delta^{H_{\rm PS}}},\qquad(4)$$

де $\delta=q_{\min}/q_{\max}$ а функція $p(H,E)=\left[2H/(E\cdot(1-H))\right]^H\times\left[\Gamma\left(\frac{E}{2}+H\right)/\left(\Gamma(1-H)\cdot\Gamma\left(\frac{E}{2}\right)\right)\right]$. Якщо $\delta=0$, $H_{\rm SF}\equiv H_{\rm PS}$. Уточнені спектральні границі фрактальності поля визначаються хвильовими числами

$$q_{\min}=\left(\frac{\Gamma\left(\frac{E}{2}+H_{\rm PS}\right)}{\Gamma(1-H_{\rm PS})\cdot\Gamma\left(\frac{E}{2}\right)}\right)^{\frac{1}{2H_{\rm PS}}}\frac{1}{\pi l_{\max}},\ q_{\max}=\left(\frac{E\cdot(1-H_{\rm PS})\cdot\Gamma(1-H_{\rm PS})\cdot\Gamma\left(\frac{E}{2}\right)}{2H_{\rm PS}\cdot\Gamma\left(\frac{E}{2}+H_{\rm PS}\right)}\right)^{\frac{1}{2(1-H_{\rm PS})}}\frac{1}{\pi l_{\min}}.\qquad(5)$$

Шукані $H_{\rm PS}$, $q_{\min}$ і $q_{\max}$ знаходимо, розв'язуючи систему рівнянь (4–5) за заданих $H_{\rm SF}$, $l_{\min}$ і $l_{\max}$.

**4.** Генерація бінарної моделі $\rho_{\rm b}(\mathbf{r})$ фрактальної пористості з параметрами $\bar{\rho}$, $\omega$, $H_{\rm b}$ та $l_{\min,\max}$ здійснюється процедурою бінаризації попередньо синтезованого неперервного поля $\rho(\mathbf{r})$

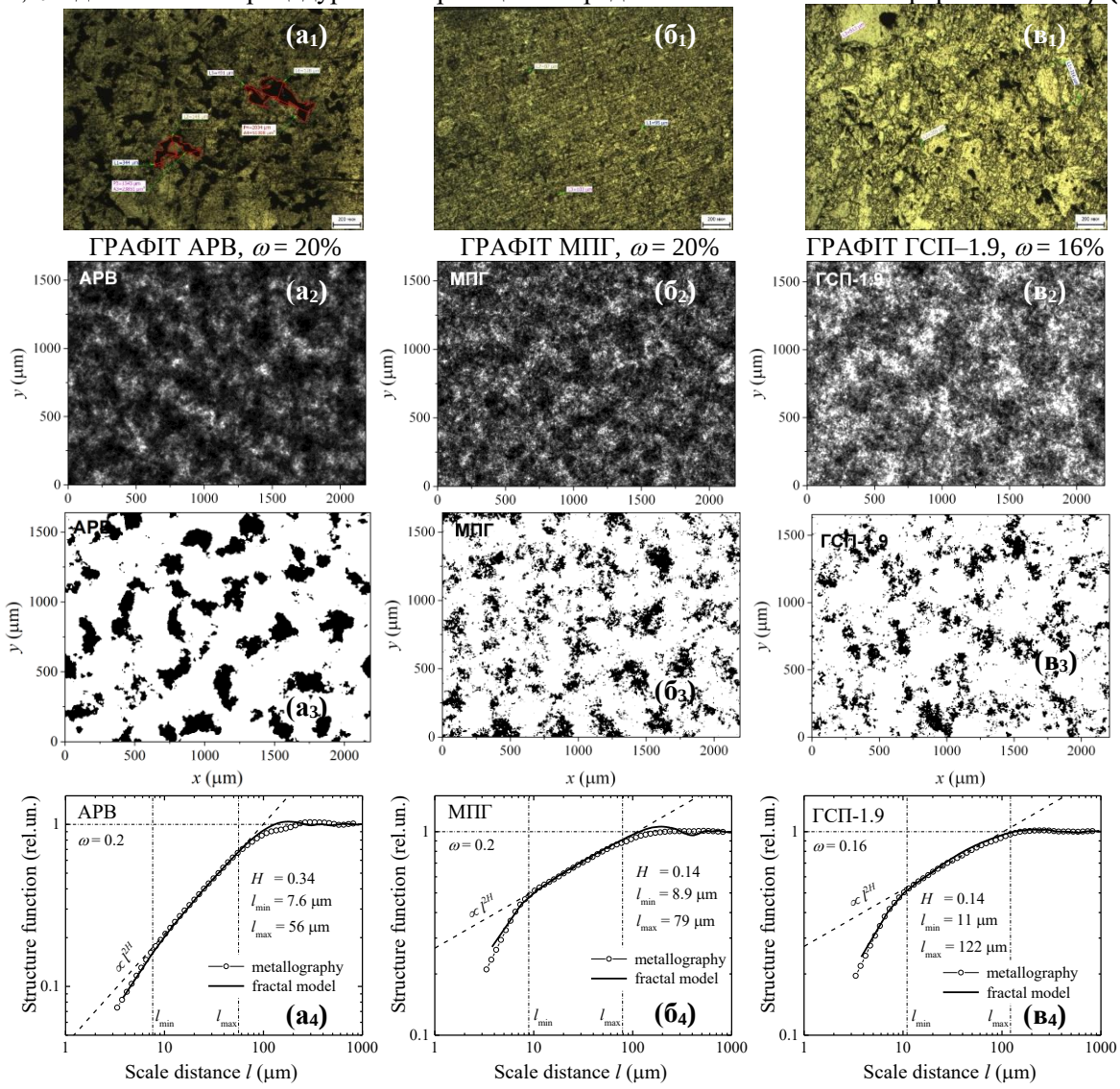



*Рис. 1 Металографічні (×50) знімки шліфів пористих графітів різних марок ($а_1$–$в_1$) та їх реконструкція алгоритмами неперервної ($а_2$–$в_2$) та бінарної ($а_3$–$в_3$) фрактальних моделей з валідацією структурних функцій $S(l)$ ($а_4$–$в_4$) експериментальних (○) та фур'є-синтезованих у бінарній моделі (криві) даних.*

з негаусівською c.d.f. $F_{NG}(\rho)$: $\rho_b(\mathbf{r}) = \bar{\rho}\cdot(1-\omega)^{-1}\cdot\Theta[F_{NG}(\rho(\mathbf{r}))-\omega] \equiv \rho_M\cdot\Theta[F_{NG}(\rho(\mathbf{r}))-\omega]$. Об'ємну пористість $\omega$ ми трактуємо, як ймовірність пустоти, і якщо $\rho(\mathbf{r}) < \rho_{th}(\omega) = \bar{\rho}\cdot\mathrm{inv}\{F_{NG}(\omega)\}$,то **r** належить порі, інакше — матриці з густиною $\rho_M$. Якщо синтез $\rho(\mathbf{r})$ робиться методом зворотної c.d.f. (п. **2**), $F_{NG}(\rho(\mathbf{r})) = F_G(x(\mathbf{r}))$ і ця процедура застосовна безпосередньо до гаусівського поля: $\rho_b(\mathbf{r}) = \rho_M\cdot\Theta[F_G(x(\mathbf{r}))-\omega]$. Це є ефективнішим, коли потрібна лише бінарна модель. Та виникає питання [7,8], як впливає бінаризація на корелятори вихідного поля $\rho(\mathbf{r})$. Вплив є істотним. За винятком граничних $\omega\to(0,1)$ $\rho_b(\mathbf{r})$ зберігає фрактальність. Але його індекс Херста $H_b \approx H_{SF}/2$ удвічі менший, а кореляційна довжина $l_{cb}$ поля загалом збільшена з коефіцієнтом ~1:

$$\left(\frac{l_{cb}}{l_c}\right)^{2H_b} \cong \frac{\sqrt{2}\pi}{4}\left(1-\mathrm{erf}^2\frac{\rho_{th}(\omega)}{\bar{\rho}}\right)\cdot\exp\left(\frac{\rho_{th}^2(\omega)}{2\bar{\rho}^2}\right). \qquad (5)$$

Тож проміжне $\rho(\mathbf{r})$ слід генерувати з подвоєним $H = 2H_b$ та відкорегованою згідно з (5) $l_{max}$.

За наведеними вище методами і формулами спектрального фур'є-синтезу нами були розроблені і реалізовані у бібліотеці класів C++ два алгоритми фрактальних генераторів — Алгоритм 1 синтезу неперервного поля та Алгоритм 2 моделювання бінарної пористості. Обидва оперують на еквідистантній решітці вокселей у 2D/3D ($E$ = 2,3) та поширюються й на довільні $E$. Фур'є-аналіз і синтез здійснюються за допомогою Open Source бібліотеки FFTW++. За стандартним інтерфейсом C++ генератори можуть використовуватись у прикладних кодах на основі, наприклад, GEANT4 і SPPARKS для застосувань у радіаційній фізиці, а також у вирішенні широкого кола астро- й геофізичних та екологічних задач.

Апробацію алгоритмів на прикладі реконструкції у 2D мікроструктурного стану опромінених пористих ядерних графітів ілюструє Рис. 1. Її адекватність підтверджується як візуально, так і кількісним узгодженням експериментальних і змодельованих напів-варіограм.

ЛІТЕРАТУРА

1. Adler P.M. Porous Media: Geometry and Transport. Reed Publ., Boston, MA, 1992. 538 p.
2. Mandelbrot B.B. The Fractal Geometry of Nature. Freeman&Co., NY, 1982. 460 p.
3. Liu Y., Li J., Sun Sh., Yu B. Advances in Gaussian random field generation: A review. *Comp. Geosci.* 2019. Vol. 23, Iss. 5. P. 1011–1047.
4. Vio1 R., Andreani P., Wamsteker W. Numerical simulation of non-Gaussian random fields with prescribed correlation structure. *Publ. Astr. Soc. of the Pacific.* 2001. Vol. 113. P. 1009–1020.
5. Yura H.T., Hanson S.G. Digital simulation of an arbitrary stationary stochastic process by spectral representation. *J. Opt. Soc. of America A.* 2011. Vol. 28, No. 4. P. 675–685.
6. Yordanov O.I, Nickolaev N.I. Self-affinity of time series with finite domain power-law power spectrum. *Phys. Rev. E.* 1994. Vol. 49, No. 4. P. R2517–R2520.
7. Holliger K., Levander A.R., Goff J.A. Stochastic modeling of the reflective lower crust: petrophysical and geological evidence from the Ivera zone (Northern Italy). *J. of Geophys. Res.*. 1993. Vol. 98, No. B7. P. 11967–11980.
8. Prigarin S.M., Martin A., Winkler G. Numerical models of binary random fields on the basis of thresholds of Gaussian functions. *Siberian J. of Numer. Math.* 2004. Vol. 7, No. 2. P. 165–175.

**БРАТЧЕНКО Михайло Іванович** – к. ф.-м. н., старший науковий співробітник Національного наукового центру «Харківський фізико-технічний інститут» Національної академії наук України, вул. Академічна, 1, Харків–108, Україна, 61108; e-mail: mbrat@kipt.kharkov.ua; ORCID: 0000–0003–3565–2036.

Наукові інтереси:

- *багатомасштабне моделювання фізичних систем, критичні явища*.

**ДЮЛЬДЯ Сергій Володимирович** – к. ф.-м. н., заступник директора Науково-виробничого комплексу «Відновлювані джерела енергії та ресурсозберігаючі технології» Національного наукового центру «Харківський фізико-технічний інститут» Національної академії наук України з наукової роботи, вул. Академічна, 1, Харків–108, Україна, 61108; e-mail: sdul@kipt.kharkov.ua; ORCID: 0000–0001–8584–3637.

Наукові інтереси:

- *теорія й математичне моделювання у радіаційній фізиці твердого тіла.*

**УДК 004.942;621.039.532.2;62-405.8;620.195;519.245**

БРАТЧЕНКО М.І., ДЮЛЬДЯ С.В.

# МОДЕЛЮВАННЯ ОКИСЛЕННЯ ФРАКТАЛЬНО-ПОРИСТИХ ЯДЕРНИХ ГРАФІТІВ МЕТОДОМ КІНЕТИЧНОГО МОНТЕ-КАРЛО

Радіаційна та корозійна стійкість графіту до високотемпературного окислення є одним з ключових чинників безпеки та ефективності його застосувань у атомній енергетиці і, зокрема, у новітніх реакторах IV покоління. Пріоритетні дані корозійних іспитів графітів [1] під *in situ* опроміненням на прискорювачі електронів ELIAS ННЦ ХФТІ у потоці кисню вимагають теоретичного пояснення, для чого у нагоді стає комп'ютерне *multiscale*-моделювання. У даній роботі ми пропонуємо модель і алгоритм та наводимо перші результати мезоскопічного моделювання дифузійно-реакційної кінетики окислення графіту методом кінетичного Монте-Карло (kMC) із застосуванням нової фрактальної моделі пористої структури ядерних графітів.

Її засадами послугували дані аналізу вихідних та опромінених [1] зразків графітів методами скануючої електронної мікроскопії й металографії разом із даними реконструкції [2] поля їх густини $\rho(\mathbf{r})$ і об'ємної пористості $\omega$ статистичним алгоритмом SNESIM (*Single Normal Equation Simulation*) пакету SGeMS (*Stanford Geostatistical Modeling Software*).

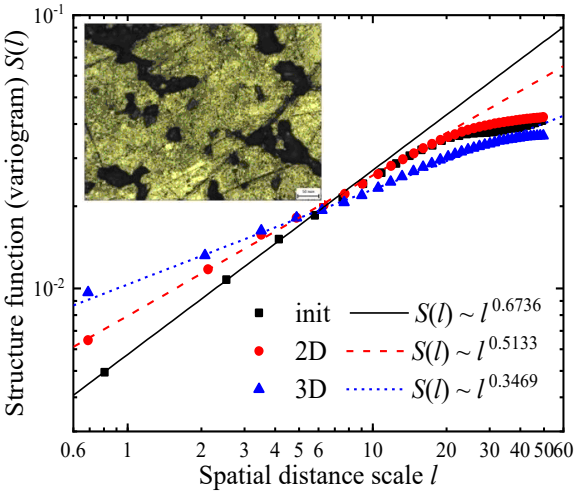



*Рис. 1 Структурні функції опроміненого* (■) *та реконструйованого* [2] *у* 2D (●) *і* 3D (▲) *зразку графіту марки АРВ, як функції масштабу l, мкм.*

Перебудувавши отримані у роботі [2] структурні функції $S(l) = \langle[\rho(l) - \rho(|\mathbf{r}+\mathbf{l}|)]^2\rangle$ (варіограми) металографічних (×200) шліфів зразків у подвійно-логарифмічному масштабі (див. Рис. 1), ми побачили, що на відстанях $l < 10$ мкм між точками $S(l) \propto l^{\alpha}$ є ступеневою функцією $l$. Знайдена поведінка є свідченням масштабної інваріантності — *self-affinity* — структури матеріалу, що загалом визначають, як її фрактальність [3]. Вона є обмеженою, бо за більших $l > 25$ мкм $S(l) \to 2\sigma^2$ насичується на подвоєній дисперсії вимірів $\sigma^2$. Це свідчить про розчеплення 2-точкових кореляцій на масштабах $l > l_{\max} \sim l_c$ — кореляційної довжини поля.

З показнику $\alpha = 2{\cdot}H$ нахилу подвійно-логарифмічної $S(l)$ ми обчислили індекс Херста $H \subset [0,1]$, що відіграє визначну роль у теорії [3]. Він визначає фрактальну розмірність $D = E - H$ $E$-вимірного поля та розрізняє два класи масштабно-інваріантних багатовидів — персистентні ($H > ½$), у яких сприятливіша ймовірність збереження тенденцій стану на фоні флуктуацій, та антиперсистентні ($H < ½$), для яких переважає тенденція до зміни стану. У зразка (■) Рис. 1 $H = \alpha/2 \approx 0.3369 \approx ⅓ < ½$, тож його пористість антиперсистентна. Це викликане існуванням на металографічному шліфі поруч із крупними порами безлічі погано розділених малих пор. Фрактальність виявляє цю неочевидну структурну закономірність. Фрактальна розмірність 2D-шліфу $D_{2D} = 2 - H \approx 1.6632$. Він репрезентує перетин (*zeroset* [3]) 3D-зразку з $D = D_{2D} + 1 \approx 2.6632$.

Реконструкція з Рис. 1 [2] цілком відтворює як дисперсію $\sigma^2$ вихідного 2D-перетину, так і його фрактальність й антиперсистентність на малих $l$. Але індекс Херста процедурно викривлений: відносно вихідного (■) $H_{■} \approx ⅓$ він зменшений до $H_{●} \approx ¼$ і $H_{▲} \approx 1/6$ за синтезу поля у 2D (●) і 3D (▲). Цієї вади використаного у роботі [2] алгоритму SNESIM позбавлена модель спектрального фур'є-синтезу фрактально-корельованої структури пористого матеріалу, яку ми застосовуємо у даній роботі (її алгоритмічні подробиці ми доповідаємо й публікуємо окремо).

Моделювання окислення й корозії (втрати маси) графітів є вельми складною задачею. Складність походить з гетерогенного характеру поверхневих реакцій і тісно пов'язує їх хімію з мезо- і мікроструктурою пористого матеріалу та наявністю у порах на відкритих окислювачеві

інтерфейсах т.з. активних центрів (АЦ) хімічних реакцій. Атомістичне *ab initio* моделювання окислення масивних пористих зразків на даний час не має перспективи. Залишається будувати феноменологічні моделі мезоскопічного рівня. Підтверджена гіпотеза про фрактальність пористості графітів надає для них слушний математичний апарат малопараметричного опису.

Наша модель дифузії і реакцій у фрактально-пористому графіті відображає кінетику окислення на еволюційну об'єктну модель поверхневої взаємодії мезоскопічних комірок. У кожній з них процеси трактуємо феноменологічно (і аналітично) у наближенні середнього поля.

Структурна залежність швидкості окислення графіту обумовлена зміненням у часі $t$ і просторі $\mathbf{r}$ повної експонованої до окислювача площі поверхні пор та відповідною зміною кількості АЦ. На атомному масштабі $a$ існує багато типів АЦ, реакційні здібності яких істотно різні. Детально урахувати їх усіх у моделі більш високого – мезоскопічного – рівня достеменно неможливо. Тому введемо скорочений укрупнений опис на мезоскопічному масштабі $l$, який:

- є досить малим ($l << l_c$), щоб розділяти фрактальну пористість, тож набагато меншим за розмір ($\sim l_c$) найбільших пор. Цим адекватно враховується відкрита кисневі площа, і можна нехтувати просторовим розкидом його та продуктів реакцій. Швидкість $k$ реакції на масштабі $l$ тоді пропорційна до середньої концентрації $n_O$ $O_2$ біля поверхні.
- є досить великим ($l \sim l_{min} >> a$), щоб містити достатньо багато АЦ. Це дозволяє нехтувати флуктуаціями їх поверхневої щільності та ввести ефективну швидкість окислення на одиницю площі, як осереднену на масштабі $l$ за різними типами АЦ.

Розіб'ємо зразок графіту на воксели об'єму $l^3$. Стан воксела задаватимемо об'ємними вмістами $n_X$ компонентів X ∈ {M,O,P}, де M — матриця (C), O — окислювач ($O_2$), а P — продукт(и) газифікації (CO, $CO_2$). Еволюцію $n_X(t)$ з часом $t$ експозиції у кожнім $i$-тім вокселі будемо описувати зв'язаною системою рівнянь локально гомогенної хімічної кінетики з урахуванням у $\tau$-наближенні дифузійного вирівнювання вмістів газоподібних компонентів:

$$\dot{n}_M^{(i)} \equiv \mathrm{d}n_M^{(i)}/\mathrm{d}t = -k \cdot n_M^{(i)} \sum_{j\in \mathrm{NN}(i|n_M=0)} n_O^{(j)}, \tag{1.1}$$

$$\dot{n}_O^{(i)} \equiv \mathrm{d}n_O^{(i)}/\mathrm{d}t = -k \cdot \sum_{j\in \mathrm{NN}(i|n_M>0)} n_M^{(j)} \cdot n_O^{(i)} + \sum_{j\in \mathrm{NN}(i|n_M=0)} D_{ij} \cdot l^{-2} \cdot \left(n_O^{(j)} - n_O^{(i)}\right), \tag{1.2}$$

$$\dot{n}_P^{(i)} \equiv \mathrm{d}n_P^{(i)}/\mathrm{d}t = \quad k \cdot \sum_{j\in \mathrm{NN}(i|n_M>0)} n_M^{(j)} \cdot n_O^{(i)} + \sum_{j\in \mathrm{NN}(i|n_M=0)} D_{ij} \cdot l^{-2} \cdot \left(n_P^{(j)} - n_P^{(i)}\right), \tag{1.3}$$

де $k$ — константа реакції окислення матеріалу матриці, $D_{ij}$ — коефіцієнт взаємної дифузії окислювача й продукту окислення поміж вокселами $i$ та $j$, функція NN($i|Q$) повертає множину $\{j\}$ індексів найближчих сусідів воксела $i$, для яких виконана булева умова $Q \in$ {true,false}. Зокрема, система (1) виходить з модельного припущення, що зміна вмісту $n_M$ вуглецю у вокселі пропорційна до вмісту $n_O$ кисню у його сусідах, які належать до пор (тобто у яких $n_M = 0$).

Матриця $D_{ij}$ враховує різну дифузію в порах істотно різного масштабу. Ми розрізняємо мезопористість $\omega$ з більших ($\geq l$) за розмір воксела пор, у яких коефіцієнт $D_{OP}$ обчислюється за наближеннями теорії Чепмена-Энскога, та нерозділену нанопористість $\omega_M$, яка властива лише вокселам, що містять матеріал матриці. Останню ми описуємо у середньополевому наближенні ефективним коефіцієнтом дифузії $D_M << D_{OP}$. Тож $D_{ij} = D_{OP}$, якщо $n_M^{(i)} = 0 \wedge j \in \mathrm{NN}(i \mid n_M = 0)$, $D_{ij} = D_M$, якщо $n_M^{(i)} > 0 \vee j \in \mathrm{NN}(i \mid n_M > 0)$, та $D_{ij} = 0$, якщо $j \notin \mathrm{NN}(i \mid \mathrm{true})$, тобто не є сусідом.

Оцінка $D_M$ вимагає даних про пористість на наномасштабі. Об'ємну нанопористість $\omega_M$ можна оцінити у дихотомічній моделі "матриця ⊕ пори", у якій середня густина $\omega$-пористого графіту $\rho = \rho_M \cdot (1 - \omega)$, $\rho_M$ — густина матриці на масштабі $l$ вокселю. По аналогії вважаємо, що $\rho_M = \rho_C \cdot (1 - \omega_M)$, де $\rho_C = 2.267$ г/см$^3$ — теоретична густина монокристалу г.щ.у.-вуглецю, що не містить пор. Тоді для певної середньої $\rho$ нанопористість $\omega_M = 1 - (\rho/\rho_C) \cdot (1 - \omega)^{-1}$ майже лінійно спадає зі зростом мезопористості $\omega$. Застосовність цієї формули обмежена малими $\omega \leq 0.3$, що для ядерних графітів цілком слушне. Інтерпретуючи $\omega_M \subset [0,1]$, як ймовірність стрибка за випадкового блукання газових молекул у нанопорах матриці, отримуємо, що $D_M = \omega_M \cdot D_{OP}$.

Для kMC-розв'язку системи (1) представимо кінетику послідовністю дискретних у $t$ переходів $n_X$ до їх локальних стаціонарних значень $\bar{n}_X$. Розглядаємо 3 варіанти виходу $n_X$ у вокселі на стаціонар: **(A)** вичерпання матеріалу матриці, **(B)** вичерпання оксиданту та **(C)**

вирівнювання вмістів окислювача і продуктів окислення за рахунок взаємної дифузії. Частоти $\Gamma \sim \tau^{-1}$ зміни стану вокселів залежать від їх локального оточення і для $\forall(i,j)$ пари сусідніх вокселів вони та значення $\vec{n}_{\rm X}^{(i)}$ у позначеному оператором $\mapsto$ кінцевому стані мають вигляд:

**(A)** $\Gamma_{\mathbf{A}}^{(ij)} = -k \cdot n_{\rm O}^{(j)}$, $\quad \vec{n}_{\rm M}^{(i)} \mapsto 0$, $\quad \vec{n}_{\rm O}^{(j)} \mapsto \vec{n}_{\rm O}^{(j)} - \vec{n}_{\rm M}^{(i)}$, $\quad \vec{n}_{\rm P}^{(j)} \mapsto \vec{n}_{\rm P}^{(j)} + \vec{n}_{\rm M}^{(i)}$, (2)

**(B)** $\Gamma_{\mathbf{B}}^{(ij)} = -k \cdot n_{\rm M}^{(i)}$, $\quad \vec{n}_{\rm M}^{(i)} \mapsto \vec{n}_{\rm M}^{(i)} - \vec{n}_{\rm O}^{(j)}$, $\quad \vec{n}_{\rm O}^{(j)} \mapsto 0$, $\quad \vec{n}_{\rm P}^{(j)} \mapsto \vec{n}_{\rm P}^{(j)} + \vec{n}_{\rm M}^{(i)}$, (3)

**(C)** $\Gamma_{\mathbf{C}}^{(ij)} = \dfrac{D_{ij}}{l^2}$, $\quad \vec{n}_{\rm O,P}^{(i)} \mapsto \dfrac{\left(n_{\rm C} - n_{\rm M}^{(i)}\right)\cdot\left(n_{\rm O,P}^{(i)} + n_{\rm O,P}^{(j)}\right)}{2n_{\rm C} - n_{\rm M}^{(i)} - n_{\rm M}^{(j)}}$, $\vec{n}_{\rm O,P}^{(j)} \mapsto \dfrac{\left(n_{\rm C} - n_{\rm M}^{(j)}\right)\cdot\left(n_{\rm O,P}^{(i)} + n_{\rm O,P}^{(j)}\right)}{2n_{\rm C} - n_{\rm M}^{(i)} - n_{\rm M}^{(j)}}$, (4)

якщо $|\vec{n}_{\rm X} - n_{\rm X}| > \Delta n_{\rm X}$, деякої допустимої похибки розрахунку, інакше $\Gamma_{\mathbf{A,B,C}}^{(ij)} = 0$. У формулах (4) $n_{\rm C} = \rho_{\rm C}/M_{\rm C} = 1.138\times10^{23}$ см$^{-3}$ — об'ємна концентрація атомів вуглецю в бездефектному графіті.

У цій мезоскопічній кінетичний моделі часова еволюція матеріалу, що окислюється, ізоморфна такій для скінченого (за числом станів 3 = ||M,O,P||) клітинного ($E$-вимірного) стохастичного (за частотами змін стану) автомату. Алгоритм його kMC-моделювання такий:

**i.** Для $t = 0$ задаємо вихідні $n_{{\rm X}\in\{\rm M,O,P\}}$ у вокселах, зокрема, за допомогою спектрального генератору бінарно-пористих матеріалів із заданими параметрами фрактальності.

**ii.** Задаємо граничні умови підведення оксиданту — поверхневі та/або об'ємні.

**iii.** Для усіх $\forall(i,j)$-пар NN-сусідніх вокселів обчислюємо за формулами (2–4) частоти $\Gamma_{\mathbf{e}}^{(ij)}$ подій $\mathbf{e} \in \{\mathbf{A},\mathbf{B},\mathbf{C}\}$ і пропоновані $\vec{n}_{\rm X}^{(i,j)}$ та розігруємо пропоновані часи відбуття подій $t_{\mathbf{e}}^{(ij)} = t - \left(\Gamma_{\mathbf{e}}^{(ij)}\right)^{-1} \ln\xi$, де $\xi$ — рівномірно розподілене на (0,1] випадкове число.

**iv.** Обираємо пару вокселів $(i_*,j_*)$ і подію $\mathbf{e}_*$ з найменшим значенням $t_{\mathbf{e}_*}^{(i_*j_*)} = \min t_{\mathbf{A,B,C}}^{(ij)}$.

**v.** Поновлюємо $n_{\rm X}^{(i_*,j_*)} \mapsto \vec{n}_{\rm X}^{(i_*,j_*)}$ для тої $\mathbf{e}_*$-тої події, котра виграла на кроці **(iv)**.

**vi.** Просуваємо поточне значення часу $t \mapsto t_{\mathbf{e}_*}^{(i_*j_*)}$.

**vii.** Перераховуємо для вокселів $(i_*,j_*)$ та їх NN-сусідів $\Gamma_{\mathbf{e}}^{(ij)}$, $\vec{n}_{\rm X}^{(i,j)}$ та $t_{\mathbf{e}}^{(ij)} = t - \left(\Gamma_{\mathbf{e}}^{(ij)}\right)^{-1} \ln\xi$.

**viii.** Повторюємо, починаючи з кроку **(iv)**, до досягнення заданого $t = t_{\max}$ моделювання.

Алгоритм був реалізований нами у kMC-пакеті SPPARKS [4]. Апробація моделі і коду була виконана для досліджених у експериментах [1] ННЦ ХФТІ графітів (див. у Табл. 1).

*Табл. 1. Структурні параметри ядерних графітів, які застосовані у валідації моделі й алгоритму.*

| Марка ядерного графіту | Середня густина $\rho$, г/см$^3$ | Мезо-пористість $\omega$ | Нано-пористість $\omega_{\rm M}$ | Індекс Херста $H$ | Фрактальна розмірність $D = 3 - H$ | $l_{\min}$, $(\sim l)$ мкм | $l_{\max}$, $(\approx l_{\rm c})$ мкм |
|---|---|---|---|---|---|---|---|
| АРВ | 1.65 | 0.20 | 0.09 | 0.34 | 2.66 | 7.6 | 56 |
| МПГ | 1.74 | 0.20 | 0.041 | 0.14 | 2.86 | 8.9 | 79 |
| ГСП-1.9 | 1.90 | 0.16 | 0.0022 | 0.14 | 2.86 | 11.0 | 122 |

Моделювалася кінетика їх окислення киснем за тиску $P = 1$ атм і температури $T = 800$°C, близької до верхньої границі дифузійно-лімітованого режиму корозійного окислення графітів. Термодинамічні параметри $O_2$, CO і $CO_2$ бралися з банку даних U.S. NIST, за 800°C вони дуже близькі до тих, що їх надає рівняння стану $PV = RT$ ідеального газу ($R = 8.321$ Дж·моль$^{-1}$·К$^{-1}$). Коефіцієнт дифузії $D_{\rm OP}$ по Чепмену-Енскогу склав 1.4 см$^2$/с, а $D_{\rm M} = \omega_{\rm M}\cdot D_{\rm OP}$ обраховували для нанопористостей $\omega_{\rm M}$ марок графітів з Табл. 1. Швидкість $k$ реакції окислення обчислювали по Ареніусу: $k = k_0\cdot\exp(-E_{\rm a}/RT)$. Феноменологічну енергію активації $E_{\rm a}$ за даними огляду [5] оцінили у $E_{\rm a} = 200$ кДж/моль з префактором константи реакції $k_0 = 3\cdot10^{-13}$ см$^3$/с. Моделювання велося у 2D на полі розміром 500×500×1 мкм для 100×100 вокселів розміром у 5×5×1 мкм.

Просування дифузійного фронту окислення візуалізували у $(x,y,t)$ засобами дружнього до SPPARKS пакету OVITO. На кожному кроці по часу $t$ обчислювали й осереднювали по Монте-Карло питому втрату маси $\delta m/S$ зразку, пористість $\omega(t)$ та, фур'є-аналізом структури, її фрактальні показники $H(t)$, $D(t) = 3 - H(t)$, а також $l_{\max} \approx l_{\rm c}(t)$. Саме останні, побудовані вперше, викликають найбільший інтерес і є головним результатом моделювання. Їх ілюструє Рис. 2(а,б).

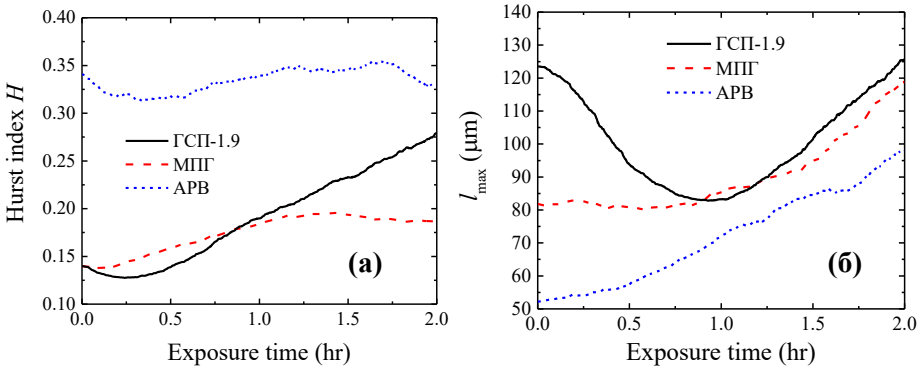



*Рис. 2 Часова еволюція параметрів H (а) і $l_{max} \approx l_c$ (б) фрактальної структури графітів різних марок.*

Окислені графіти не втрачають фрактальності, але її покажчики істотно змінюються з часом експозиції. У графітів різних марок залежності Рис. 2 різні. Саме це є визначальним з прикладної точки зору. Нам вдалося якісно пояснити їх у класичних уявленнях про різниці гранулометричного складу і спектру пористості зразків. Цей розгляд виходить за межі даної роботи, та імовірно сприятиме оптимізації технологій реакторних графітів.

Змодельована корозійна кінетика (див. Рис. 3) спочатку лінійна та втім прямує до параболічної $\propto t^{1/p}$. Результати моделювання узгоджуються з визначеним у іспитах [1] рейтингом АРВ → МПГ→ГСП графітів за зростом корозійної стійкості, але, як видно з порівняння з експериментом (○) на Рис. 3, істотно занижують $|\delta m/S|$. Вочевидь константи [5] не враховують прискорення [1] реакцій іонізуючим опроміненням. Ефект не є тривіальним і вимагає подальшого дослідження.

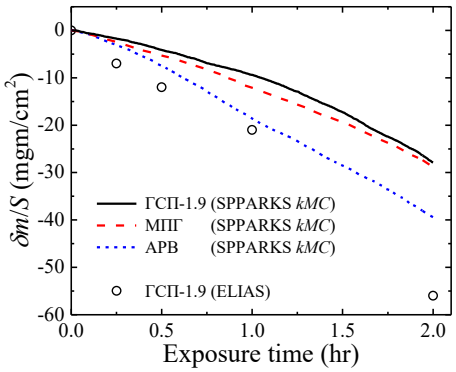



*Рис. 3 Кінетичні криві корозійної втрати маси зразків графітів різних марок в порівнянні з вимірами* (○) [1] *для графіту ГСП-1.9 під e⁻-опроміненням.*

ЛІТЕРАТУРА

1. Зеленский В.Ф., Одейчук Н.П., Рыжов В.П., Борисенко В.Н., Гамов В.О., Ляшенко А.Н. *и др.* Исследование коррозионной стойкости графитов под облучением электронами в потоке кислорода при температурах 600–800°С. *ВАНТ.* 2013. Вып. 5(87). С. 125–130.
2. Комир А.И. Реконструкция структуры ядерного графита методами многоточечной статистики. *Вестник НТУ «ХПИ».* 2016. № 17(1189). С. 18–24.
3. Федер Е. Фракталы. М.: «Мир», 1991. 254 с.
4. Plimpton S. , Battaile C., Chandross M., Holm L. *et al.* Crossing the mesoscale No-Man's Land via Parallel Kinetic Monte Carlo. Sandia report SAND2009–6226, October 2009. 85 p. URL: https://spparks.sandia.gov/pdf/sand09.pdf (Last accessed: 06.03.2019).
5. El-Genk M.S., Tournier J.-M.P. Comparison of oxidation model predictions with gasification data of IG-110, IG-430 and NBG-25 nuclear graphite. *JNM.* 2012. Vol. 420, Iss. 1-3. P. 141-158.

**БРАТЧЕНКО Михайло Іванович** – к. ф.-м. н., старший науковий співробітник Національного наукового центру «Харківський фізико-технічний інститут» Національної академії наук України, вул. Академічна, 1, Харків–108, Україна, 61108; e-mail: mbrat@kipt.kharkov.ua; ORCID: 0000–0003–3565–2036.

Наукові інтереси:

- *багатомасштабне моделювання фізичних систем, критичні явища.*

**ДЮЛЬДЯ Сергій Володимирович** – к. ф.-м. н., заступник директора Науково-виробничого комплексу «Відновлювані джерела енергії та ресурсозберігаючі технології» Національного наукового центру «Харківський фізико-технічний інститут» Національної академії наук України з наукової роботи, вул. Академічна, 1, Харків–108, Україна, 61108; e-mail: sdul@kipt.kharkov.ua; ORCID: 0000–0001–8584–3637.

Наукові інтереси:

- *теорія й математичне моделювання у радіаційній фізиці твердого тіла.*

**УДК 004.052**

БУБЕР Д.И., ПАВЛОВ А.Н.

# МОДЕЛЬ РАСЧЕТА ПОКАЗАТЕЛЕЙ НАДЕЖНОСТИ АСУ ТП

**Определение надежности**

Надежность автоматизированных систем управления — свойство объекта выполнять заданные функции, сохраняя во времени значения установленных эксплуатационных показателей в заданных пределах, соответствующих определенным режимам и условиям пользования, технического обслуживания, ремонтов, хранения и транспортировании.

Надежность *Н* является комплексным свойством объекта, включающим следующие четыре составляющие:

- безотказность автоматизированных систем управления *Б* — свойство объекта непрерывно сохранять работоспособность в течение некоторого времени или некоторой наработки;
- ремонтопригодность автоматизированных систем управления *Р* — свойство объекта, заключающееся в приспособленности к предупреждению и обнаружению причин возникновения его отказов, повреждений и устранению их последствий путем проведения ремонтов и технического обслуживания;
- сохраняемость автоматизированных систем управления С — свойство объекта непрерывно сохранять исправное и работоспособное состояние в течение и после хранения и (или) транспортировки;
- долговечность автоматизированных систем управления *Д* — свойство объекта сохранять работоспособность до наступления предельного состояния при установленной системе технического обслуживания и ремонтов.[1]

**Рисунок АСУ ТП. Описание АСУ ТП**

Автоматизированная система управления технологическим процессом - человеко-машинная система управления, обеспечивающая автоматизированный сбор и обработку информации, необходимой для оптимизации управления технологическим объектом в соответствии с принятым критерием.[2]

На рисунке (рис.1) изображен пример АСУ ТП. Данная АСУ ТП состоит из комплекса аппаратных и программных средств. Аппаратные средства оснащены датчиками, которые выполняют измерение технологических параметров. В случае отказа одного из элементов, в АСУ предусмотрены резервные элементы (или системы), необходимые для безотказной работы АСУ ТП.

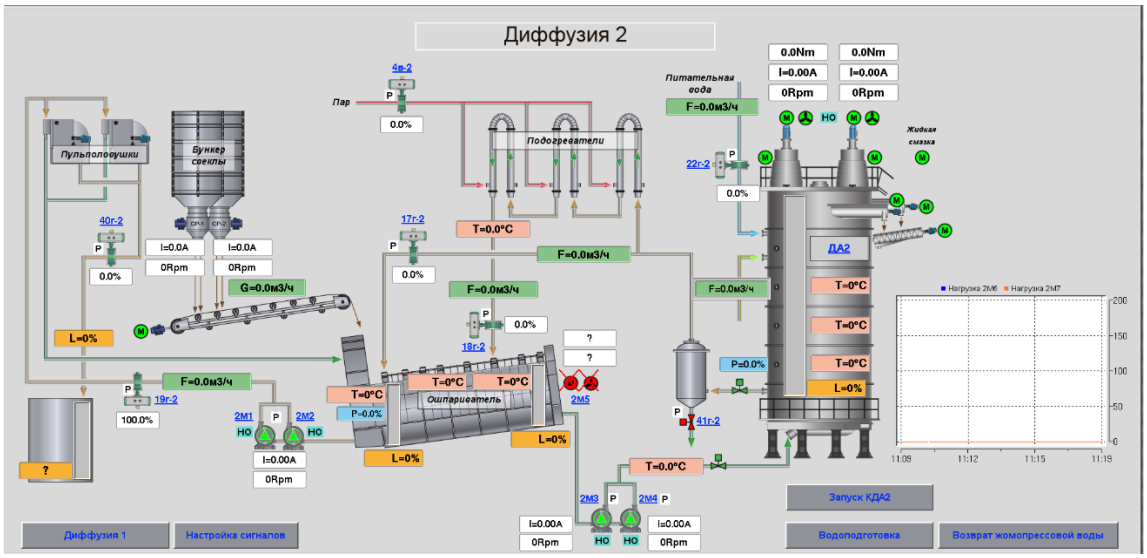



*Рис. 1 АСУ ТП диффузионных аппаратов на сахарном заводе*

**Как представить АСУ ТП в виде структурно-логических схем**

Структурная схема для расчета надежности любой из подсистем, входящих в сложную радиоэлектронную систему, рассматривается как сочетание последовательно и параллельно связанных элементов. Последовательное соединение относится к подсистемам или элементам сложных систем, работающим без резерва. Параллельное соединение относится к резервированным подсистемам или элементам.[3] Блок-схема параллельного соединения приведена на Рис.2.

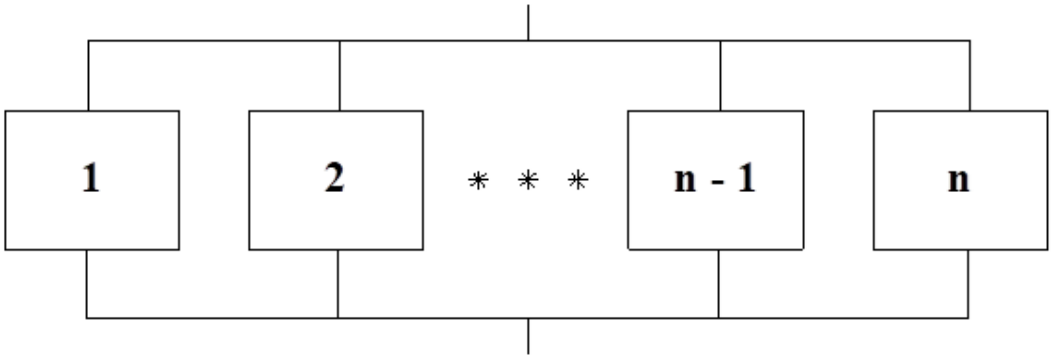



*Рис. 2 Блок схема параллельного соединения.*

На рисунке (Рис.3) приведен пример структурно логической схемы диффузионных аппаратов сахарного завода. В АСУ имеется параллельное соединение(резервирование) таких элементов: пульповушки, бункер свеклы, 2 пары электромоторов.

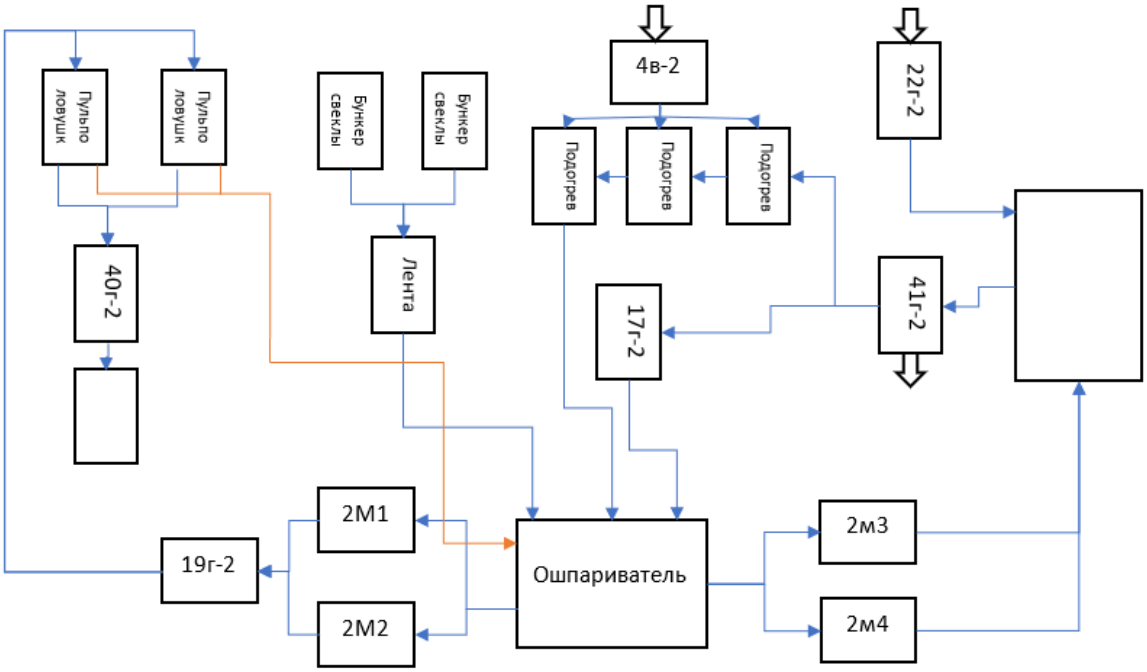



*Рис.3 Структурно логическая схема АСУ ТП диффузионных аппаратов*

**Алгоритм модели расчета параметров в СЛС**

Для построения модели надежной АСУ ТП необходимо воспользоваться алгоритмом расчета параметров. Данный алгоритм позволяет построить надежную структурно-логическую схему по заданным параметрам, внести коррективы в подсистемы. Блок-схема данного алгоритма изображена на Рис. 4.

Алгоритм требует от пользователя структуру и некоторые заданные параметры, которые требуется достичь. Следующим шагом АСУ ТП необходимо разделить на подсистемы,

отвечающие за конкретное действие (система энергоснабжения, система подачи воды, и т. д.). В свою очередь СЛС может иметь как последовательное, так и параллельное соединение. Для каждого типа соединения существуют формулы (1), (2) расчета вероятности безотказной работы. Следующий шагом следует проверить, удовлетворяет ли текущий коэффициент готовности (7) изначально заданному значению, в случае удовлетворения условия получаем готовую СЛС. В противном случае необходимо выделить наиболее слабый элемент СЛС и принять решение, зарезервировать элемент, либо в случае избытка резервных элементов – заменить на более усовершенствованный. Далее производится повторный пересчет параметров с проверкой на удовлетворение заданного условия.

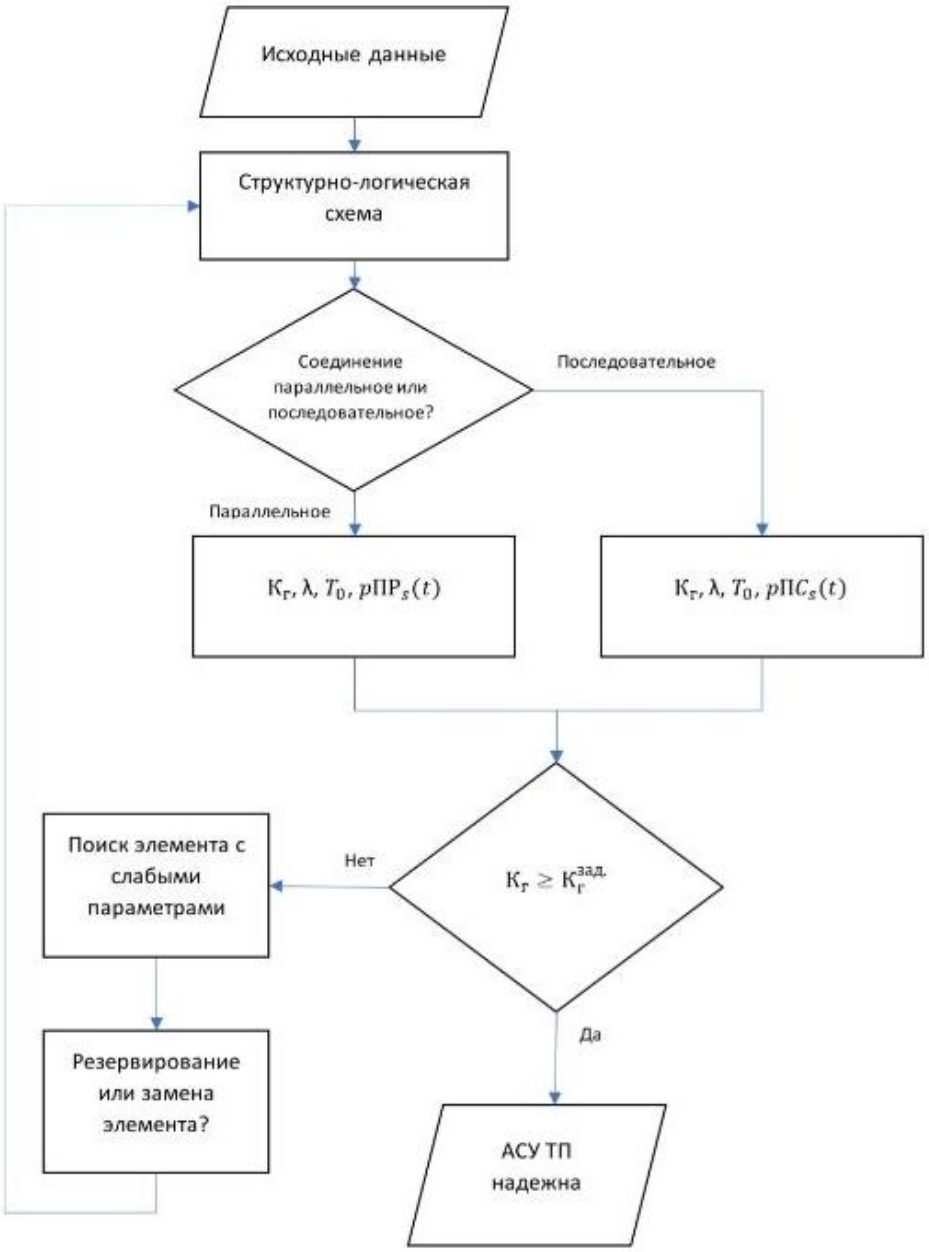



*Рис. 4 Блок-схема модели расчета параметров САС*

**Формульные соотношения**

Надежность системы ***$P_s(t)$,*** состоящей из ***n*** последовательных элементов, характеризуемая вероятностью безотказной работы, согласно теории вероятностей равна

$$pПС_s(t) = \prod_{i=1}^{n} P_i(t) \quad (1)$$

где $P_i(t)$ – вероятность безотказной работы i-го элемента.

Вероятность безотказной работы $P_s(t)$ для системы с параллельными элементами равна

$$p\Pi P_s(t) = 1 - \prod_{i=1}^{n}[1 - P_i(t)] \quad (2)$$

Среднее время наработки на отказ резервированной аппаратуры с восстановлением определяется выражением

$$T_{\text{op}} = (\sum_{i=0}^{n} Q_i / \Lambda_n Q_n) \quad (3)$$

где $n$ – число резервных элементов

$$Q_0 = 1;\ Q_1 = \frac{\Lambda_0}{M_1};\ Q_2 = \frac{\Lambda_1 \Lambda_0}{M_2 M_1}; \ldots; Q_i = \left(\frac{\Lambda_{i-1}}{M_i}\right) Q_{i-1}; \quad (4)$$

$$\Lambda_{i-1} = k\,\lambda + (n-i)\,\lambda; \quad M_i = i\,\mu \quad (5)$$

где $\lambda$ – интенсивность отказов;

$k$ – число рабочих элементов;

$\mu = 1/T_в$ – интенсивность восстановления;

Параметр потока отказов такой системы (аппаратуры) равен

$$\lambda = 1/T_{op}\ , \quad (6)$$

а коэффициент готовности

$$K_{\text{г}}^{\text{ср}} = \sum_{i=0}^{n} Q_i / \sum_{i=0}^{N} Q_i \quad (7)$$

где $N = n + k$ – общее число элементов системы

ЛИТЕРАТУРА

1. Основные понятия о надежности автоматизированных систем управления технологическими процессами и производствами. URL: ritm.pro/osnovnye-ponjatija-o-nadezhnosti-avtomatizirovannyh-sistem-upravlenija-tehnologicheskimi-processami-i-proizvodstvami
2. В.А. ВТЮРИН, АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ ОСНОВЫ АСУТП: Учебное пособие для студентов специальности 220301 «Автоматизация технологических процессов и производств» (по отраслям): СПб, 2006. 153 с.
3. Андреев Ф.М., Бердников А.Г. Методические рекомендации по проектной оценке надежности: Харьков: ХНУ имени В. Н. Каразина, 2016. 18 с.

**БУБЕР Дмитрий Игоревич** – студент 4 курсу факультета компьютерных наук специальности «Автоматизация та компьютерно-интегрированные технологии», Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 4, Харьков-22, Украина, 61022; e-mail: buber.dima@gmail.com; ORCID: 0000-0001-8199-5916

Научные интересы:

*– математическое моделирование сложных систем.*

**ПАВЛОВ Анатолий Николаевич** – старший преподаватель кафедры теоретической и прикладной системотехники; Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 4, Харьков-22, Украина, 61022; e-mail: tps@karazin.ua.

Научные интересы:

*– математическое моделирование сложных систем.*

**УДК - 004.052:681.3**

БУЄВИЧ-СИСОЄВ В.М., ШМАТКОВ С.І.

# МОДЕЛЬ КЛАСИФІКАЦІЇ СТАНІВ КОМП'ЮТЕРНИХ СИСТЕМ

**Вступ**

З розвитком сучасних технологій передачі даних з'явилося поняття мультисервісної мережі (МСС), що дозволяє оператору зв'язку об'єднати надання всіх типів послуг: передачу даних, голосу, відео через єдину телекомунікаційну середу передачі. Доставка по єдиній мережевій інфраструктурі такого різнорідного трафіку є досить складним завданням, і є найбільш перспективним напрямком розвитку мереж. Нормальне функціонування МСС неможливо без аналізу трафіку та класифікації стану комп'ютерних систем, які є життєво важливим компонентом для розуміння вимог і можливостей мережі. Тому дослідження структури трафіку мереж, а також класифікація станів комп'ютерних систем є актуальним завданням сучасної науки.

**Аналіз основних показників якості роботи комп'ютерних мереж**

При розробці моделі класифікації станів комп'ютерних систем проаналізовані основні показники якості роботи комп'ютерних мереж, які є основними при наданні послуг. До таких показників відносять: середню затримку передачі пакетів інформації (мс), відхилення від середнього значення затримки передачі пакетів інформації (мс), коефіцієнт втрати пакетів інформації, коефіцієнт помилок в пакетах інформації визначеним стандартом CISCORFC 791 [1]. Розглянуті основні типи трафіка, які можуть зустрічатися в мережах. Крім того на сьогоднішній день прийнято ряд національних стандартів у сфері якості послуг зв'язку.

**Послуги мережі і класи сервісу [2]**

До послуг мережі відносяться:

- передача традиційного телефонного трафіку;
- передача трафіку даних Інтернет;
- передача трафіку даних корпоративної мережі;
- передача трафіку мобільних мереж;
- доступ в мережу Інтернет;
- доступ до мереж передачі даних;
- передача голосового трафіку IP- телефонії;
- передача відеотрафіка для організації відеоконференцій;
- організація віртуальної приватної мережі VPN;
- послуги щодо забезпечення гарантійного рівня обслуговування.

Існує кілька підходів до класифікації інформації по різних характеристиках:

- за призначенням інформації;
- по необхідній швидкості передачі;
- за характером навантаження;
- по чутливості до затримок;
- по чутливості до втрат.

За призначенням інформацію можна розділити на наступні групи:

- інформація від користувача (площину U),
- службова інформація для управління з'єднаннями в мережі (Площину C),
- службова інформація для моніторингу та експлуатаційного управління мережами зв'язку (площину M).

За чутливості до затримок розрізняють:

- Інформацію реального часу. До неї відноситься мовна та відеоінформація в режимі діалогу, час затримки для яких не більше 0,1 сек.
- Транзакції, які мають час затримки не більше 1 сек. До транзакцій відносяться, наприклад, запити до розподілених баз даних.
- Дані. Ті, хто має час затримки більше 1 сек.

Проаналізувавши основні показники якості роботи комп'ютерних мереж та згідно з рекомендацією Міжнародного союзу електрозв'язку (МСЕ) Y.1541 [3] сервіси діляться на

класи якості послуг (6 класів), тому в залежності від значень мережі і класів сервісу передачі інформації: затримки доставки, варіації затримки доставки, частки втрачених пакетів, переданих з помилкою та ін.[4]., ми по класу якості послуги, типу трафіку та характеристиці доставки інформації можем виділити 6 станів комп'ютерних систем, що представлені в таблиці 1.

*Таблиця 1. Відповідність комп'ютерної системи різним видам трафіка*

| Клас стану комп'ютерної системи | Характеристика стану комп'ютерної системи |
|---|---|
| 0 | Потоки реального часу, дуже чутливі до варіації затримки |
| 1 | Потоки реального часу, чутливі до варіації затримки |
| 2 | Передача даних, відрізняється високим ступенем інформаційного обміну (сигнальна інформація) |
| 3 | Передається зі змінною швидкістю, не критичні до затримок, передається за встановленим з'єднанню. (комп'ютерні дані, що передаються в Frame Relay) |
| 4 | Потоки, чутливі до втрати інформації в процесі її передачі (масиви даних, потокове відео) |
| 5 | Традиційні додатки IP мереж |

**Висновок.** Метою роботи було розробити модель класифікації станів щоб на її базі в дисертаційній роботі виконати аналіз потокового трафіку. В статті описаний аналіз трафіку, проведено огляд послуг і класів сервісу мережі, розроблена модель станів. В результаті проведеної роботи реалізована модель класифікації станів комп'ютерної системи. Дана модель буде застосовуватись в дисертаційній роботі, а також може бути використана для навчання технічних фахівців управлінню трафіком в умовах перевантаження.

ЛИТЕРАТУРА

1. Marina del Rey, INTERNET PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION, Information Processing Techniques Office 1400 Wilson Boulevard Arlington, Virginia 22209.
2. Уэнделл Одом Книга "Официальное руководство Cisco по подготовке к сертификационным экзаменам CCENT/CCNA ICND1 100-101" (Cisco CCENT/CCNA ICND1 100-101: Official Cert Guide) 978-5-8459-1906-9
3. МСЭ-Т Recommendation Y.1541. Network Performance Objectives for IPBased Services//May 2002.
4. Олифер В., Олифер Н. Компьютерные сети. Принципы, технологии, протоколы : Учебник для вузов. 5-е изд., 2016. 992 с.
5. Лосев Ю. І., Руккас К. М., Шматков С. І. Комп'ютерні мережі: навч. посіб. / за редакцією Ю. І. Лосева. Харків : ХНУ імені В. Н. Каразіна, 2013. 248 с.

**БУЄВИЧ-СИСОЄВ Владислав Миколайович** – ст. викладач; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 6, м. Харків, 61022; e-mail: Vladbuevich1993@gmail.com; ORCID*: https://orcid.org/0000-0002-4195-0526.

**ШМАТКОВ Сергій Ігорович** – Завідувач кафедрою, доктор технічних наук, профессор кафедри теоретичної та прикладної системотехніки, Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 6, м. Харків, 61022; e-mail: sershmat@gmail.com.

Наукові інтереси:

*–методи управління, організація інформаційних процесів і моделювання складних адаптивних розподілених систем, системний аналіз.*

**УДК 004.7**

БУЗОВЕРЯ Д.О., МОРОЗ О.Ю.

# АНАЛИЗ ТЕХНОЛОГИЙ СОЗДАНИЯ WEB-САЙТОВ ДЛЯ ОБЕСПЕЧЕНИЯ РАБОТЫ РЕСТОРАННОГО БИЗНЕСА

**Введение**

Интернет представляет собой сеть взаимодействующих компьютеров друг с другом, которые также называют серверами. В свою очередь каждый сервер имеет свой уникальный IP-Address, необходимый для идентификации данного сервера в общей сети. Количество подключенных устройств в этой общей сети в 2020 достигнет 25 миллиардов, и это без учета ноутбуков, планшетных компьютеров и смартфонов, так как они подключены к Сети изначально. Естественно, что такой компьютер должен иметь свой IP-Address для мгновенного обращения к нему.

При этом все такие сервера можно разделить на 2 вида:

• Клиент-сервер. Клиент-сервера хранят содержание самих сайтов. В зависимости от: мощности сервера, типов сайтов, количества контента и посещаемости – на одном таком клиент-сервере может располагаться от одного и до нескольких тысяч сайтов.

• DNS-сервер.

После отправки клиент-сервером файлов в браузер, он, в соответствии со своими алгоритмами и принципами преобразования, преобразует файлы к тому виду, который был задуман разработчиком web-сайта.

Целью данной работы является изучение списка наилучших технологий для создания Web-сайтов для обеспечения максимальной финансовой прибыли работы ресторанного бизнеса.

**Постановка задачи**

Ресторану требуется Web-система для увеличения объемов продаж, рекламного изучения в потребительской среде, а также организации обратной связи с покупателями.

**Веб-система** виртуальной площадки хостинга состоит из двухуровневой системы веб-серверов. Внешний уровень принимает запросы из сети, пересылает его на внутренний уровень, читает ответ и отдаёт его в сеть. Такая схема позволяет разгрузить внутренний уровень за счёт его независимости от скорости сегментов сети Интернет и гибко управлять внутренним уровнем системы, распределяя запросы извне к веб-серверам внутреннего уровня с различной, иногда несовместимой между собой, функциональностью.

На этот проект выделяется определенные человеческие, временные и финансовые ресурсы, а также предъявляются определенные требования к качеству.

Для создания оптимального Web-сайта необходимо количественно и качественно оценить возможные варианты использованных технологий.

**Классификация технологий для создания Web-сайтов**

Выбор технологии для создания Web-сайта характеризуется следующими аспектами: легкость в использовании; широкий функционал; возможность связи с другими технологиями; популярность. Названные условия связаны между собой и приводят к определенным технологиям которые выделяются по всем этим признакам.

В работе сделана классификация технологий для создания Web-сайтов, которая позволила систематизировать и выделить наилучшие. Данная классификация представлена в виде схемы (рис.1), образованной группами технологий отвечающих за определенную сферу.

Схема представляет собой иерархическую структуру, соответствующую следующим иерархическим моделям: технология отвечающая за программную часть, за разметку сайта, за внешний вид, за базу данных . Она может быть использована при анализе и выборе технологий для написания сайта.

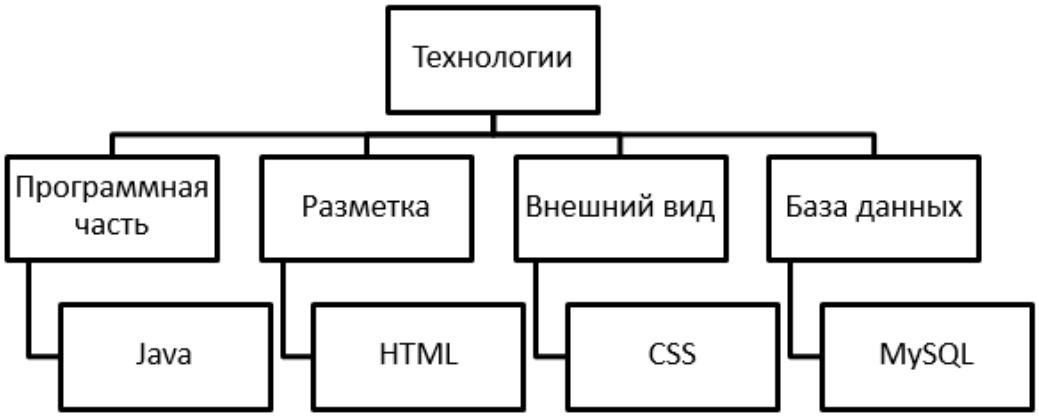



*Рис. 1 Схема технологий*

**Суть разработанной классификации технологий для создания Web-сайтов**

Проведя исследования, была разработана классификация, которая по-нашему мнению представляет собой выборку наилучших технологий для проектирования web-сайтов. Они хорошо взаимодействуют между собой, что позволяет разработчику не тратить время на тестирование возможных комбинаций технологий. В работе дано обоснование проектных решений, которые были приняты в процессе создания сайта ресторана, а также определены основные компоненты, необходимые для разработки составляющих сайта, на основании чего сформулирована и построена его структура.

Таким образом, предлагаемая модель оценки технологий позволяет проводить более комфортное создание веб-сайта.

ЛІТЕРАТУРА

1. Гарнаев А.Ю. Web-программирование на Java и JavaScript. – СПб.: БХВ-Петербург, 2005.
2. Хабибуллин И. Разработка Web-служб средствами Java. – СПб.: БХВ-Петербург, 2003.
3. Обзор современных Web - технологий – [Электронный ресурс] - http://www.sciteclibrary.ru/rus/catalog/pages/6643.html.

**БУЗОВЕРЯ Денис Олегович –** студент группы КУ-41 факультета компьютерных наук; Харьковский национальный университет имени В. Н. Каразина, площадь Свободы 6, Харків, Украина, 61022; e-mail: d_byzoverya@i.ua; ORCID: 0000-0002-5866-7719.

Научные интересы:

*– Программирование. Управление проектами. Алгоритмы и структуры данных.*

**МОРОЗ Ольга Юрьевна –** старший преподаватель кафедры теоретической и прикладной системотехники факультета компьютерных наук; Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 6, Харьков, Украина, 61022; e-mail: o.moroz@karazin.ua; ORCID: 0000-0002-4920-4093.

Научные интересы:

*– Технологии автоматического проектирования параллельных программ.*

**УДК 004.7**

БУТКО Е.А., ПАВЛОВ А.Н.

# МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ СИСТЕМЫ РАСЧЕТА ОСНОВНЫХ ПАРАМЕТРОВ СЕРВЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

**Актуальность вычислительной техники**

На текущем этапе развития электронной вычислительной техники, повышение оперативности вычислительных систем за счет увеличения отдельных электронных вычислительных машин не является актуальной. Наибольшей популярностью и актуальностью в сфере совершенствования вычислительной техники пользуется создание и применение кластеров и многопроцессорных электронно-вычислительных машин, многократно повышающих общую производительность и эффективность вычислительных систем. На данный момент существует большое количество разнообразных вычислительных систем и сетей для достижения своей конкретной цели. Хорошим примером являются серверные вычислительные системы для прогнозирования погодных условий, предупреждении стихийных бедствий, которые в режиме реального времени собирают и обрабатывают огромное количество данных.

**Методология построения вычислительных систем**

В связи с быстрым и непрерывным развитием электронной вычислительной техники, появлением более сложных в реализации компьютерных систем и сетей, постоянным увеличением априорной информации, усилением взаимосвязей явлений и процессов различной природы, появилась потребность в решении сложных, неформализованных задач, оценке существующих систем и повышении их эффективности. Для достижения указанных целей рационально использовать системный подход, в котором моделирование является одним из основных методов исследования. Системный подход предполагает такие этапы решения сложной задачи:

- Изучение предметной области, в которой проводится исследование;
- Выявление и формализация задачи моделирования;
- Математическое моделирование исследуемых объектов и процессов
- Статистическая обработка результатов моделирования
- Формулировка и оценка эффективности альтернативных решений
- Формулирование выводов и предложений по решению проблемы, повышению эффективности использования исследуемой системы, объектов.

В общем случаем, процесс исследования можно представить в виде формальной формы:

$$Y_{(t)} = f[X_{(t)}, Q_{(t)}] \text{ – функция выходов,} \quad (1)$$

$$Q_{(t)} = g[X_{(t)}, Q_{(t-1)}] \text{ – функция переходов,} \quad (2)$$

$$X_{(t)} = u[Y_{(t-1)}] \text{ – функция управления процессом.} \quad (3)$$

Здесь X(t) - множество значений входных факторов в момент времени t, Q (t) - множество значений параметров, характеризующих различные внутренние состояния сложной системы в этот же момент времени, Y(t) и Y(t-1) - множества значений измеряемых показателей изучаемых свойств системы в обозначенные моменты времени. Первые два уравнения моделируют суть изучаемого процесса, а третье уравнение является математическим описанием (моделью) процесса воздействий исследователя на изучаемую систему. Исследователю, как правило, доступно только определенное подмножество Y'(t) наблюдаемых параметров и весьма ограниченное подмножество X'(t) управляемых факторов. Его представление о внутренних состояниях исследуемой системы также ограничено некоторым подмножеством Q '(t).

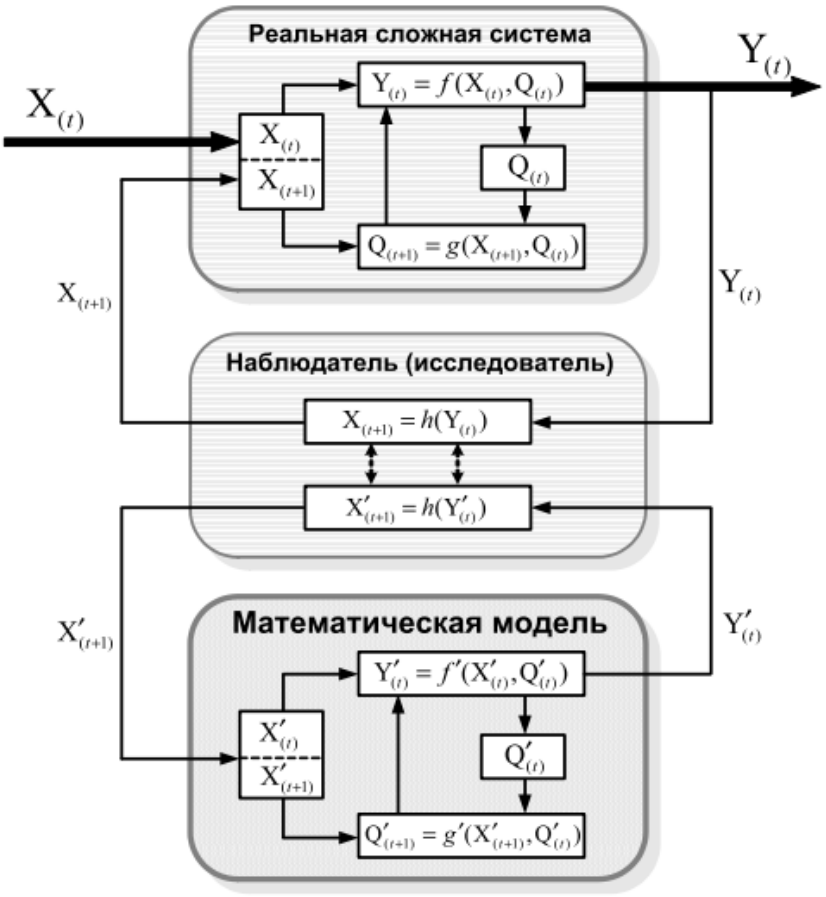



*Рис. 1. Схема обобщенной математической модели процесса сложной системы*

В представлении исследователя математическая модель системы имеет вид:

$$Y'_{(t)} = f'[X'_{(t)}, Q'_{(t)}] \quad (4)$$

$$Q'_{(t+1)} = g'[X'_{(t+1)}, Q'_{(t)}] \quad (5)$$

Схема обобщенной математической модели процесса сложной системы приведена на рисунке 1.

**Система расчета основных параметров серверных вычислительных систем**

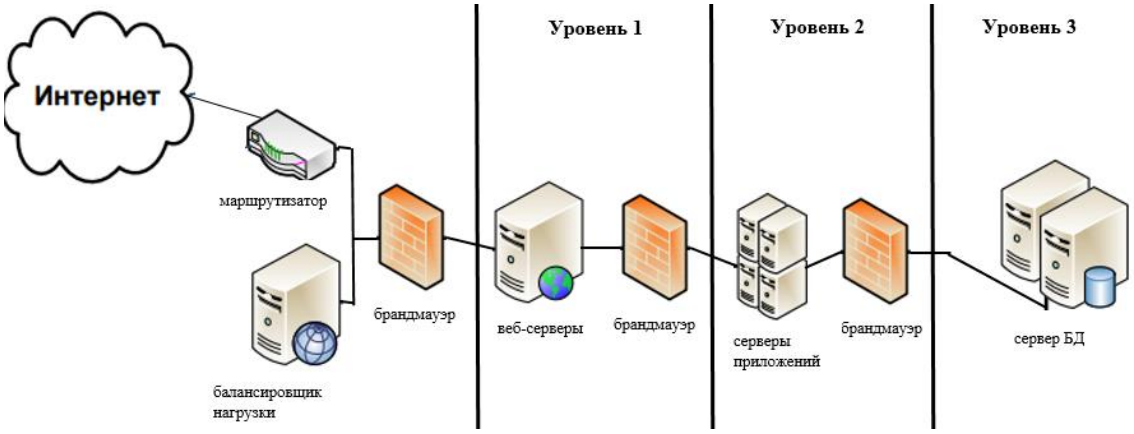



*Рис. 2. Трехуровневая архитектура вычислительной сети*

Системы, основанные на Web-технологиях, как правило, строятся по многоуровневой архитектуре: 1-ый уровень реализует интерфейс пользователя с сетью (уровень представления), 2-ой уровень – содержит логику приложения (прикладной уровень), 3-ий уровень – отделяет 1-ый и 2-ой уровни, повышая надежность системы.

Каждая из задач, выполняемая вычислительной сетью, характеризуется значениями определенных свойств:

- $P_{\text{запр}}$ – общий поток запросов в ед. времени (количество запросов в час);
- V – пропускная способность сети (Мб/с);
- $K_{\text{пр}}^{CPU}$ – коэффициент пропускной способности процессорного блока серверов системы;
- $K_{\text{пр}}^{IO}$ – коэффициент пропускной способности блока ввода-вывода серверов системы;
- $k_{\text{запр}}$ – количество запросов передаваемых системе;
- $t_{\text{запр}}$ – время, требуемое на обработку одного запроса (мс).

Также структурные элементы каждого сервера обладают такими свойствами:

- процессоры – коэффициентом быстродействия $r_{CPU}$;
- устройства ввода-вывода – коэффициент пропускной способности $r_{in/out}$;
- устройства памяти – объемом памяти $C_{RAM}$, $C_{HDD}$.

Формализуем принципы математической модели текущей системы, определив характер задач поступающих на ее вход:

- каждый запрос представлен вектором $x_i = (i, z_1, \dots, z_j)$, i – номер запроса, $z_j$ – значение j-го параметра, для i-го запроса;
- запросы поступают в виде потока $X_{(t)}$ с периодичностью $f(X)$;
- множества запросов сводятся к среднему значению $m(z_i)$;
- на выходе – вектор $y_i = (i, t_i^{\text{обр}}, t_i^{\text{кон}})$, где $t_i^{\text{обр}}$ – время затраченное на обработку i-го запроса, $t_i^{\text{кон}}$ – время конца обработки.
- функция выхода системы $Y_{(t)} = f_1[X_{(t)}, Q_{(t)}]$ – процесс обработки $X_{(t)}$ в $Y_{(t)}$ и оценок отклика системы $W_{(t)}$
- внутреннее состояние $Q_{(t+1)} = f_2[X_{(t+1)}, Q_{(t)}]$, где $Q_{(t)}$ = декартовому произведению множеств текущий состояний компонентов системы $q_i$, $f_2$ – алгоритм, описывающий функцию переходов от $Q_{(t)}$ к $Q_{(t+1)}$.

Общая концептуальная математическая модель обработки запросов для трехуровневой архитектуры представлена в виде схемы:

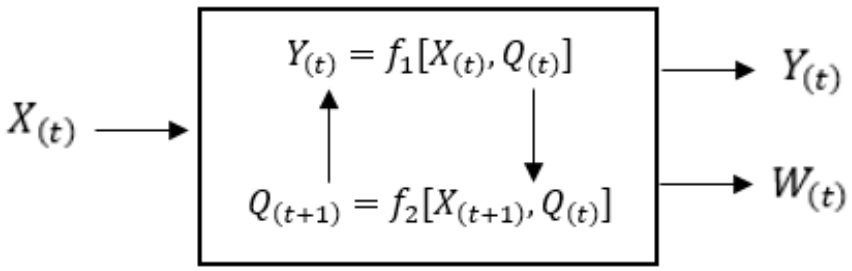



*Рис. 3. Обобщенная схема математической модели сетевой трехуровневой архитектуры*

Расчет основных параметров серверной компьютерной системы:

Нахождение времени обслуживания одного запроса для блоков CPU и IO Веб-сервера соответственно, где $w_{n,i}$ – коэффициент нагрузки CPU/IO блока сервера.

$$tW^{CPU} = w_{1,i} * KW_{\text{пр}}^{CPU} \quad (6)$$

$$tW^{IO} = w_{2,i} * KW_{\text{пр}}^{IO} \quad (7)$$

Нахождение времени обслуживания одного запроса для блоков CPU и IO Сервера приложений соответственно, где $a_{n,i}$ – коэффициент нагрузки CPU/IO блока сервера.

$$tA^{CPU} = a_{1,i} * KA_{\text{пр}}^{CPU} \quad (8)$$

$$tA^{IO} = a_{2,i} * KA_{\text{пр}}^{IO} \quad (9)$$

Нахождение времени обслуживания одного запроса для блоков CPU и IO Сервера приложений соответственно, где $d_{n,i}$ – коэффициент нагрузки CPU/IO блока сервера.

$$tD^{CPU} = d_{1,i} * KD_{\text{пр}}^{CPU} \quad (10)$$

$$tD^{IO} = d_{2,i} * KD_{\text{пр}}^{IO} \quad (11)$$

ЛИТЕРАТУРА

1. Стеценко І.В. Моделювання систем: навч. посіб. Черкаси: ЧДТУ, 2010. 399 с.
2. Устенко А.С. Основы математического моделирования и алгоритмизации, 2000. Электронный ресурс. Режим доступа: http://dit.isuct.ru/IVT/BOOKS/Model/Model2/index.html
3. Лосев Ю. І., Руккас К. М., Шматков С. І. Комп’ютерні мережі: навч. посіб. / за редакцією Ю. І. Лосева. Харків : ХНУ імені В. Н. Каразіна, 2013. 248 с.
4. Міжнародний науковий журнал «Інтернаука» №7, 2017.

**БУТКО Евгений Андреевич** – студент кафедры теоретической и прикладной системотехники; Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 4, Харьков-22, Украина, 61022; e-mail:butkoevgenii@gmail.com; ORCID*: 0000-0003-0496-6921.

Научные интересы:

– математическое моделирование системы расчета основных параметров серверных вычислительных систем и сетей.

**ПАВЛОВ Анатолий Николаевич** – старший преподаватель кафедры теоретической и прикладной системотехники; Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 4, Харьков-22, Украина, 61022; e-mail: tps@karazin.ua.

Научные интересы:

– *математическое моделирование сложных систем.*

**УДК 621.311**

ВАНІН В.А., ЛАЗУРЕНКО О.П., КРУГОЛ М.М.

# МАТЕМАТИЧНІ МОДЕЛІ ТА ОПТИМІЗАЦІЯ РОБОТИ ГРУП МЕХАНІЗМІВ ВЛАСНИХ ПОТРЕБ ТЕС

**Постановка задачі.**

Основними споживачами електричної енергії в системах власних потреб теплових електричних станцій є відцентрові механізми – вентилятори та насоси. Сукупність цих механізмів, забезпечує надійну і економічну роботу основного устаткування ТЕС – парового котла та турбогенератора, утворюючи взаємодіючі складні гідравлічні та електричні системи. Метою роботи є математичне моделювання та оптимізація режимів роботи таких систем.

Деякі особливості математичного моделювання процесу керування складними системами із гідравлічними, газодинамічними та електромеханічними процесами розглядались в роботах [1, 2]. Математичні моделі механізмів власних потреб ТЕС можна знайти в [3, 4].

Існують три способи сумісної роботи відцентрових механізмів в мережі – паралельна, послідовна та змішана робота. Так, якщо стоїть задача збільшення продуктивності механізмів – використовується їх паралельна робота. Для збільшення результуючого напору – відцентрові механізми включають послідовно.

**Паралельна робота відцентрових механізмів.**

Розглянемо паралельну роботу відцентрових механізмів на спільну мережу (рис.1). Умовою їх узгодженої роботи є рівність напорів $H_i$ з якими вони працюють [5]. Продуктивність групи $N$ паралельно працюючих відцентрових механізмів $Q_\Sigma$ є сумою продуктивностей кожного з механізмів $Q_i$.

$$H_\Sigma = H_1 = \ldots = H_i = H_N, \quad Q_\Sigma = \sum_{i=1}^{N} Q_i, \quad i = 1 \ldots N .$$

Як показано в [3], напор, що розвиває відцентровий механізм та його ККД, є функціями від продуктивності, кута відкриття направляючого апарату $\alpha$ та швидкості обертання робочого колеса або частоти живильної напруги $f$, якщо для приводу використовуються асинхронні двигуни з перетворювачами частоти.

$$H = H_i(Q, \alpha_i, f_i) \quad \Rightarrow \quad Q = Q_i(H, \alpha_i, f_i), i = 1..N \tag{1}$$

$$\eta = \eta_i(Q, \alpha_i, f_i) \tag{2}$$

Тоді, група з $N$ паралельно працюючих механізмів на спільну мережу з еквівалентним гідравлічним опором $\xi$ буде характеризуватися параметрами керування – векторами частот живильної напруги та кутів відкриття направляючого механізму агрегату

$$\{\alpha_i, f_i\}_{i=1}^{N}, \vec{\alpha} = (\alpha_1 \ldots \alpha_N), \vec{f} = (f_1 \ldots f_N), \tag{3}$$

еквівалентним напором та еквівалентним ККД

$$H = H_\Sigma\left(Q_\Sigma, \vec{\alpha}, \vec{f}\right), \tag{4}$$

$$\eta_\Sigma = \Omega(\eta_1 \ldots \eta_N) = \Psi\left(\vec{Q}, \vec{\alpha}, \vec{f}\right), \tag{5}$$

причому, ККД системи, можна розраховувати як середньозважений

$$\Psi\left(\vec{Q}, \vec{\alpha}, \vec{f}\right) = \frac{\sum_{i=1}^{N} P_i \eta_i(Q_i, \alpha_i, f_i)}{\sum_{i=1}^{N} P_i}, \tag{6}$$

де $P_i$ – потужність електроприводу механізму,

та законом мережі

$$H = H_c(Q_\Sigma). \tag{7}$$

Для побудови характеристики мережі можна скористатися наступною залежністю

$$H = \xi Q^2 .$$

Побудова еквівалентної характеристики групи паралельно працюючих механізмів можна знайти, наприклад в [5].

Часто виникає необхідність визначення розподілення продуктивностей, між паралельно працюючими на спільну мережу відцентровими механізмам, тому виникає наступна задача

***Задача 1.*** *Нехай, задана група N паралельно працюючих відцентрових механізмів з характеристиками*

$$H = H_i(Q, \alpha_i, f_i), i = 1 \ldots N, \text{ де } \alpha_i = \overline{\alpha}_i,\ f_i = \overline{f}_i,$$

*які працюють на спільну мережу з еквівалентним гідравлічним опором* $\xi$. *Знайти розподілення продуктивностей* $Q_i^*$, *з якими працюють відцентрові механізми групи.*

Для вирішення цієї задачі будується еквівалентну характеристику групи паралельно працюючих механізмів. Виходячи з умови роботи групи відцентрових механізмів на мережу [3], отримаємо рівняння

$$H_\Sigma\left(Q^*, \vec{\alpha}, \vec{f}\right) = H_c\left(Q^*\right).$$

Графічно-чисельним методом знайдемо $Q^*$ - продуктивність групи механізмів в мережі з відомим напором $H_\Sigma^*$. Продуктивності механізмів знайдемо з рівнянь

$$H_i\left(Q_i^*, \alpha_i, f_i\right) = H_\Sigma^*, i = 1 \ldots N.$$

В умовах вирішення задачі підвищення енергоефективності роботи теплових електричних станцій, виникає потреба визначення оптимальних параметрів керування відцентровими механізмами.

***Задача 2.*** *Задана група N паралельно працюючих відцентрових механізмів з характеристиками*

$$H = H_i(Q, \alpha_i, f_i), i = 1 \ldots N.$$

*Задані необхідна сумарна продуктивність групи* $Q_\Sigma^*$, *та коефіцієнт гідравлічного опору мережі -* $\xi$. *Знайти розподілення продуктивностей, з якими працюють відцентрові механізми групи* $Q_i^*$ *та параметри керування -* $\vec{\alpha} = (\alpha_1 \ldots \alpha_N)$, $\vec{f} = (f_1 \ldots f_N)$, *таким чином, що ККД групи механізмів* $\Psi\left(\vec{Q}, \vec{\alpha}, \vec{f}\right)$ *буде мати найбільше значення.*

Введемо множини $A = \{\vec{\alpha}\}, \alpha_i \in \left[\alpha_{i\min}, \alpha_{i\max}\right]$ - область значень вектора $\vec{\alpha}$, що визначається конструктивними обмеженнями значень кутів відкриття направляючих механізмів, та

$f \in F, F = \left\{\vec{f}\right\}, f_i \in \left[f_{i\min}, f_{i\max}\right]$ - область значень вектора $\vec{f}$, що визначається конструктивними обмеженнями по частотам живильної напруги.

Така задача оптимізації запишеться у вигляді

$$\left(\vec{\alpha}, \vec{f}\right) = \arg \max_{\alpha \in A,\ f \in F,\ \omega\left(\vec{\alpha}, \vec{f}\right) = 0} \Psi\left(\vec{Q}, \vec{\alpha}, \vec{f}\right), \tag{8}$$

де $\Psi\left(\vec{Q}, \vec{\alpha}, \vec{f}\right)$ - функція ККД системи паралельно працюючих механізмів, наприклад як середньозважене ККД механізмів групи,

$\omega\left(\vec{\alpha}, \vec{f}\right) = 0$ - рівняння, що зв'язує між собою параметри керування таким чином, щоб виконувалась умова забезпечення еквівалентним механізмом заданої продуктивності в мережі

$$H_\Sigma\left(Q_\Sigma^*, \vec{\alpha}, \vec{f}\right) - H_c\left(Q_\Sigma^*\right) = 0,$$

**Послідовна робота відцентрових механізмів.**

Розглянемо послідовну роботу відцентрових механізмів на спільну мережу (рис. 2). Умовою їх спільної роботи є рівність продуктивностей з якими вони працюють [5]. Еквівалентний напор групи є сумою напорів кожного з механізмів.

$$Q_\Sigma = Q_1 = \ldots = Q_i = Q_N, \quad H_\Sigma = \sum_{i=1}^{N} H_i, \quad i = 1 \ldots N,$$

Для послідовного з'єднання відцентрових механізмі також будуть виконуватися залежності (1-7). Еквівалентну характеристику послідовно працюючих механізмів можна знайти аналітично – сумуючи характеристики кожного з механізмів

$$H_\Sigma = \sum_{i=1}^{N} H_i(Q, \alpha_i, f_i)$$

Для знаходження оптимальних параметрів керування групою послідовно включених відцентрових механізмів вирішимо наступну задачу.

***Задача 3.*** *Задана група N послідовно працюючих відцентрових механізмів з характеристиками*

$$H = H_i(Q^*, \alpha_i, f_i), i = 1 \ldots N .$$

*Задана необхідна сумарна продуктивність групи* $Q^*$, *в мережі з еквівалентним коефіцієнтом гідравлічного опору* $\xi$.

*Необхідно знайти розподілення напорів* $H_i^*$ *та параметри керування* $\vec{\alpha} = (\alpha_1 \ldots \alpha_N)$, $\vec{f} = (f_1 \ldots f_N)$, *таким чином, що ККД групи механізмів* $\Psi(\vec{Q}, \vec{\alpha}, \vec{f})$ *буде мати найбільше значення.*

Для вирішення цієї задачі необхідно вирішити задачу оптимізації (8) з обмеженням $\omega(\vec{\alpha}, \vec{f}) = 0$ у вигляді

$$\sum_{i=1}^{N} H_i(Q^*, \alpha_i, f_i) - H_c(Q_\Sigma^*) = 0 .$$

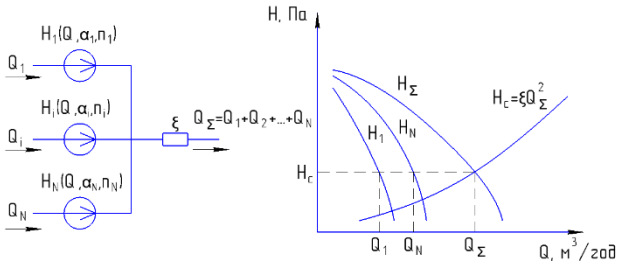



*Рис. 1 – Паралельна робота відцентрових механізмів*

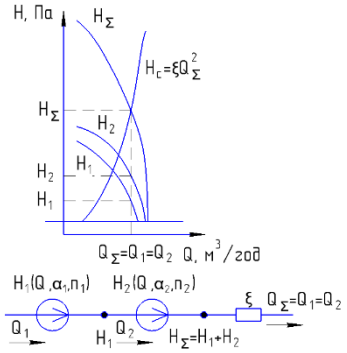



*Рис. 2 – Послідовна робота відцентрових механізмів*

**Приклад.**

Розглянемо повітряний тракт барабанного парового котла MANN-120. Два дуттєвих вентилятора ВДН-17-3, що приводяться в обертання асинхронними електричними двигунами, працюють на спільний повітропровід, в якому відбувається підігрів повітря, та його транспортування до горілок котла. Згідно з експлуатаційною документацію еквівалентний коефіцієнт гідравлічного опору повітропроводу $\xi = 0{,}04$. Тобто, розглянемо випадок *N*=2 паралельно працюючих механізмів на мережу із еквівалентним гідравлічним опором 0,04 мм·с$^2$/м$^6$.

На рис. 3 показане рішення задачі 1 з наступними вхідними даними: частота живильної напруги обох електроприводів – 50Гц. Кути відкриття направляючого механізму $30^0$ та $10^0$. Результатом розрахунку є розподілення продуктивностей у вигляді - $Q_1^* = 38{,}21$ тис.м3/год, $Q_2^* = 46{,}15$ тис.м3/год, $Q_\Sigma^* = 84{,}35$ тис.м3/год, та напори, що розвиваються ними $H_\Sigma^* = H_1^* = H_2^* = H_c^* = 290$ Па.

Також, для даних механізмів була вирішена задача 2, з заданою сумарною продуктивністю $Q_\Sigma^*$=80 тис.м$^3$/год (рис. 4). Результатом розрахунку є оптимальні параметри

керування даної групи вентиляторів. 1ДВ: $Q_1^* = 45{,}11$ тис.м3/год, $\alpha_1 = 31{,}2^0$, $f_1 = 47{,}9$ Гц, 2ДВ: $Q_2^* = 34{,}85$ тис.м3/год, $\alpha_2 = 32{,}7^0$, $f_2 = 44{,}6$ Гц

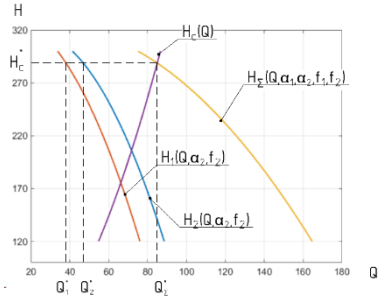



*Рис 3. – Побудова еквівалентної характеристики двох паралельно працюючих відцентрових механізмів*

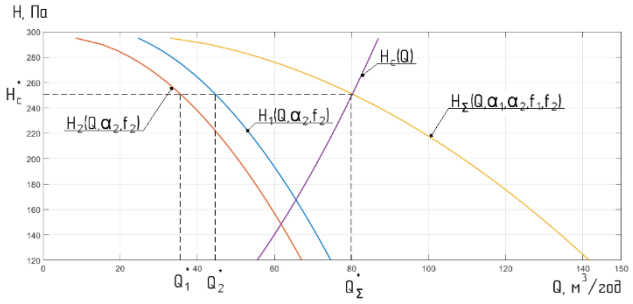



*Рис 4. – Побудова еквівалентної характеристики двох паралельно працюючих відцентрових механізмів, при вирішенні задачі оптимізації*

**Висновок.**

Приведений алгоритм розрахунку оптимальних параметрів керування відцентровими механізмами, які працюють на спільну мережу, дає змогу проводити детальний аналіз роботи гідравлічних мереж у взаємодії з електричними, для теплової електричної станції з метою підвищення її енергоефективності. Даний алгоритм може бути взятий за основу для аналізу більш складних гідравлічних систем.

ЛІТЕРАТУРА

1. Селезнев В.Е., Алешин В.В., Прялов С.Н. Математическое моделирование трубопроводных сетей и систем каналов: методы, модели и алгоритмы / Под ред. В.Е. Селезнева. – М.: МАКС Пресс, 2007. – 695 с.
2. Костышин В.С. Моделирование режимов работы центробежных насосов на основе электрогидравлической аналогии. Ивано-Франковск, 2000. - 163 с
3. Group Regulation Efficiency Analysis for Thermal Power Plant Auxiliaries/ N.Kruhol, O.Lasurenko, V.Vanin, et al. // 2019 IEEE 6th International Conference on Energy Smart Systems (ESS). doi: 10.1109/ESS.2019.8764242.
4. Wang Shuping, Ye Jiantao, Li Wei, Du Xiaofeng, Chen Zinia, Wang Shuping - Energy efficiency evaluation investigation on high voltage inverter retrofit for fans and pumps in power plants, CIGRE, 2012
5. Вахвахов Г.Г. Работа вентиляторов в сети. М.: Стройиздат, 1975, - 101 с

**ВАНІН Віктор Антонович** – д. т. н., професор; професор кафедри вищої математики; Національний технічний університет «Харківський політехнічний інститут», вул. Кирпичова, 2, Харків, Україна, 61000; e-mail: vvaplb5256@gmail.com; ORCID: 0000-0002-3523-7505

Наукові інтереси:

*– математичне моделювання гідро-газодинамічних та електродинамічних процесів в суцільному середовищу*

**ЛАЗУРЕНКО Олександр Павлович** – к. т. н., професор; професор кафедри електричних станцій; Національний технічний університет «Харківський політехнічний інститут», вул. Кирпичова, 2, Харків, Україна, 61000; e-mail: lazurenkoAP@i.ua; ORCID: 0000-0002-4409-629X

Наукові інтереси:

*– підвищення енергоефективності в системі виробництва, передачі та розподілу електричної енергії.*

**КРУГОЛ Микола Михайлович** – аспірант кафедри електричних станцій; Національний технічний університет «Харківський політехнічний інститут», вул. Кирпичова, 2, Харків, Україна, 61000; e-mail: kruhgol@gmail.com; ORCID: 0000-0002-6090-4875

Наукові інтереси:

*– підвищення енергоефективності теплових електричних станцій.*

**УДК 004.35: 004.9**

ВАРЛАМОВА Н., ЛАЗУРИК В., СТЄРВОЄДОВ М.

# АПАРАТНО-ПРОГРАМНИЙ КОМПЛЕКС ДЛЯ ПРОВЕДЕННЯ ПСИХОФІЗІОЛОГІЧНИХ І ПСИХОСОЦІАЛЬНИХ ДОСЛІДЖЕНЬ

Застосування сучасних інформаційних технологій та мікроконтролерної техніки забезпечує якісно нові можливості для створення апаратно-програмних засобів для психологічної діагностики особистості або групи людей на всіх її етапах. Явною перевагою комп'ютерних засобів психологічної та професійної діагностики є швидкість отримання, обробки та інтерпретації отриманих первинних даних, що, в свою чергу, позитивно відбивається на зайнятості експерта - звільняє його від трудомістких рутинних операцій і дозволяють зосередитися на вирішенні інших професійних завдань. Можливість адаптивної і оперативної зміни або модифікації особистісних психофізіологічних тестів, яка з'являється при цьому, дозволяє будувати більш адекватні моделі стану випробуваного, чітко визначати його фізіологічні, психічні та поведінкові характеристики. Це вкрай необхідно для вирішення мети тестування - психодіагностики стану для профвідбору, переатестації, визначення курсу реабілітації і т.п. Використання сучасної апаратно - програмної бази, технології Интернету речей й хмарних сервісів надає широкі можливості для дистанційної роботи психодіагностичних систем. Включення функції віддаленої діагностики до складу приладу або системи збільшує їх професійну і комерційну привабливість.

Метою цієї роботи було створення апаратно-програмного комплексу (АПК), призначенням якого є проведення психофізіологічних і псіхосоціальних досліджень, зокрема для вивчення реакції людини на комплексні світлові, колірні, звукові і тактильні подразники, які пред'являються випробуваному у вигляді плоских 2Д або об'ємних 3Д сигналів на спеціально розроблених стендах. При цьому, АПК є складовою частиною єдиної навчально-дослідницької інформаційно-вимірювальної і керуючої системи прийняття рішень в предметної області вивчення професійної придатності, психофізіологічного стану людини і психологічної сумісності групи людей [1].

Основні функції апаратно-програмного комплексу і, які описано в докладі, наступні:

1. Пред'явлення випробуваному комплексних плоских 2Д або об'ємних 3Д сигналів і фіксація відповідної реакції;
2. Збір, попередня обробка, стискання і передача даних в хмарний сервіс для подальшої обробки інформації;
3. Обробка даних і приведення їх до вигляду, зручного для сприйняття дослідниками та експертами;
4. Представлення доступу до даних дослідникам і експертам за допомогою спеціальних комп'ютерних програм і мобільних додатків;
5. Обмін даними між дослідниками, експертами і випробуваними для уточнення результатів тестів і остаточної експертної оцінки;
6. Автоматичний підбір методик;
7. Формування бази даних і презентація результатів.

Згідно з функціями розроблено структуру моделі апаратно-програмного комплексу, що включає в себе блок управління (робоче місце експерта), блок тестування (робоче місце досліджуваного), блок отримання передачі даних, блок обробки збереження даних. Також описано структуру кожного блоку, що містить в собі перелік компонентів та способи взаємодії між ними.

Комплекс передбачає наявність декількох інформаційно-виконавчих модулів психодіагностичної системи і блоків психологічних методик, побудованих відповідно до мети дослідження (психодіагностика, профвідбір і ін.).

Інформаційно-виконавчі модулі психодіагностичної системи побудовані на базі потужного 32 розрядного мікроконтролера STM32F4.

Підключення, ініціація, отримання показників датчиків та їх візуалізація на сенсорному моніторі реалізується завдяки бібліотекам, що знаходяться у складі STM32 IDE.

Передача даних від виконавчих пристроїв здійснюється завдяки роботі Wi-Fi модуль ESP8266-12 та AT-командам (до складу модулю входить процесор AT-команд). Розглянуто кінцеве підключення зконфігурованого модулю.

Збором даних, їх обробкою та аналізом займається хмарний сервіс ThingSpeak, що має для цього відповідні канали. Передача даних до хмарного сервісу здійснюється завдяки HTTP запитам.

Проведення дослідження та створення експертної оцінки здійснюється завдяки мобільному додатку, що пов'язаний із СУБД та хмарним сервісом ThingSpeak завдяки відповідним інтерфейсам.

Розроблений поетапний план реалізації моделі апаратно-програмного комплексу: від установки, налаштування і підключення елементів комплексу до загального тестування його роботи.

ЛИТЕРАТУРА

1. N.V. Varlamova, N.G. Styervoyedov. Hardware - software complex for psychological and professional diagnostics with the remote control function. Вісник Харківського національного університету імені В.Н. Каразіна, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»: 2018, Том 38 ,с.25 – 32.
2. Кочина М. Л. Многофункциональный прибор для проведения психофизиологических исследований / М. Л.Кочина, А. Г. Фирсов. // Прикладная радиоэлектроника. – 2010. – Т.9.– №2. – С.260–265.
3. Росляков, А.В. Р75 Интернет вещей: учебное пособие [текст] / А.В. Росляков, С.В. Ваняшин, А.Ю. Гребешков. – Самара: ПГУТИ, 2015. – 200 с.
4. Беличенко П.В., Благиня Н.В., Мальцев А.В., Хруслов М.М. Аппаратно-программное обеспечение в психологии. Труды международной науч.-техн. конф.. — Х.: ХНУ им. В.Н.Каразина, 2014.- С. 22-25.
5. В. Д. Балин, В. К. Гайда, В. К. Гербачевский и др.. Практикум по общей, экспериментальной и прикладной психологии / Под общей ред. А. А. Крылова, С. А. Маничева. — 2-е изд., доп. и перераб. — СПб.: Питер. — 560 с.: ил. — (Серия «Практикум по психологии»). 2003.
6. Кальніш В. В. Принципи професійного психофізіологічного відбору /В. В. Кальніш, А. І. Єна // Гігієна праці. – Вип. 32. – К., 2001. – С.131–144.

**ВАРЛАМОВА Наталя Володимирівна** – аспірант кафедри моделювання систем і технологій ; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 6, Харків-22, Україна, 61022; e-mail: natess123@gmail.com; ORCID: 0000-0001-5117-7293.

Наукові інтереси:

– *Інтернет речей, психологія.*

**ЛАЗУРИК Валентин Тимофійович** – д. ф.-м. н., професор; декан факультету комп'ютерних наук; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 6, Харків-22, Україна, 61022; e-mail: mst@karazin.ua; ORCID: 0000-0002-8653-5609.

Наукові інтереси:

– *моделювання пучково-плазмених систем.*

**СТЄРВОЄДОВ Микола Григорович** – к. т. н., професор; завідуючий кафедрою електроніки і управляючих систем; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 6, Харків-22, Україна, 61022; e-mail: n.styervoyedov@karazin.ua, Keus@karazin.ua; ORCID: 0000-0003-0136-6437.

Наукові інтереси:

– *проектування систем автоматичного управління, схемотехніка электронних обчислювальних машин.*

**УДК 532.59**

ВАХНЕНКО В.О., ВЕНГРОВИЧ Д.Б.

# ТЕОРЕТИЧНЕ ОБҐРУНТУВАННЯ МЕТОДУ ДІАГНОСТИКИ ВЛАСТИВОСТЕЙ СЕРЕДОВИЩА ДОВГИМИ НЕЛІНІЙНИМИ ХВИЛЯМИ

Більшість середовищ за умови локальної рівноваги можна вважати безструктурними. Традиційно припускають, що збурення з довжиною хвилі $\lambda$, що значно перевищує характерний розмір $\varepsilon$ структурних неоднорідностей, поширюються в них, як в однорідних. Відомо, що з погляду механіки суцільного середовища можлива ідеалізація реального середовища за допомогою однорідного середовища. У багатьох випадках це дало змогу досягти значного успіху під час опису хвильових процесів.

Сучасний стан експериментальних досліджень потребує удосконалення моделей неоднорідних середовищ з детальним урахуванням їхньої структури. Реальні середовища не є однорідними.

Як правило, для побудови моделей тією чи іншою мірою використовують формалізм механіки суцільного середовища. В таких випадках початковим є принцип локальної дії, що дає можливість перенести закони механіки точкової маси на суцільне середовище. Під час перетворення інтегральних рівнянь збереження у диференційні рівняння припускають існування диференційно малого мікрооб'єму $d\upsilon$. З одного боку, цей об'єм настільки малий, що можна поширити закони механіки точкової маси на весь мікрооб'єм $d\upsilon$, з іншого – мікрооб'єм, хоч і малий порівняно з об'ємом усього середовища, все ж містить так багато структурних елементів середовища, що в цьому сенсі він може бути розглянутий як макроскопічний. Отже, перехід до диференційних рівнянь збереження ґрунтується на такому припущенні: розмір мікроструктурних масштабів $\varepsilon$ малий порівняно з характерним макроскопічним масштабом течії λ, що виправдовує граничний перехід $\varepsilon / \lambda \to 0$. Загалом стягування об'єму $d\upsilon$ в точку є правильним для неперервних функцій. Це означає, що всі точки всередині диференційно малого об'єму еквівалентні. Тому еквівалентність точок у мікрооб'ємі обґрунтовує припущення про використання усереднених характеристик хвильового поля. Отже, рівняння руху можуть бути записані в усереднених характеристиках, таких як густина, масова швидкість, тиск, що властиві кожному окремому компоненту середовища. Зауважимо, що характерні структурні розміри окремих компонентів у цих моделях явно не фігурують.

У разі застосування моделей однорідного середовища до опису динамічних хвильових процесів у структурованому середовищі виникають деякі принципові труднощі. Тут структуру середовища розглянуто на макрорівні. Ми відмовилися від припущення, що диференційно малий об'єм $d\upsilon$ містить усі компоненти середовища, хоч і розглянуто довгохвильові наближення, коли довжина хвилі $\lambda$ набагато більша за характерну довжину структури середовища $\varepsilon$ (рис. 1). Вважаємо, що окремо взятий компонент структурованого середовища моделюється однорідним середовищем (диференційно малий об'єм $d\upsilon$ значно менший за характерний розмір окремого компонента). Згідно з математичним аналізом методом асимптотичного усереднення [1, 2], структура середовища безпосередньо впливає на нелінійні хвильові процеси навіть для збурень з довжиною хвилі, що значно перевищує розміри неоднорідностей. Математичне формулювання цього твердження означає, що система усереднених рівнянь не виражається в усереднених характеристиках (тиск, масова швидкість, питомий об'єм) і містить члени з характерним розміром окремих компонентів.

Елементарними неоднорідними середовищами, для яких можна проаналізувати вплив структури, є середовища з регулярною структурою. Регулярність структури і нелінійність досліджуваних хвильових процесів визначають вибір математичних моделей. Лінійні розміри тіла є значно більшими, ніж розмір неоднорідностей, проте неоднорідності настільки великі, що їхній стан описують класичними рівняннями суцільного середовища (рис. 1). Закономірності поширення довгохвильових збурень досліджуємо на прикладі середовища з регулярною

структурою, вважаючи, що і напруження, і масова швидкість є неперервними функціями на межі сусідніх компонентів (див. рис. 1).

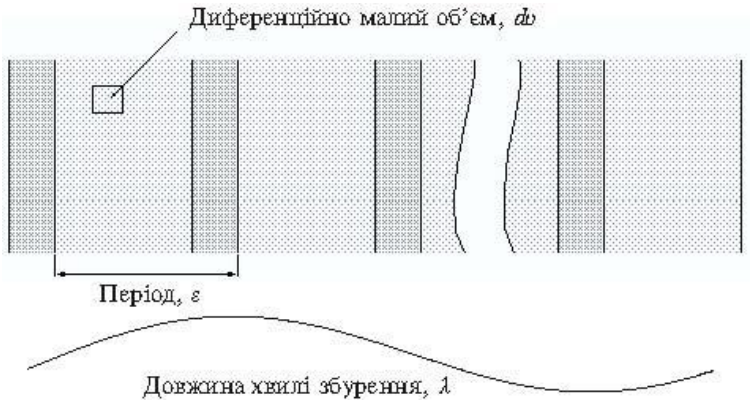



*Рис. 1. Модель шарувато-неоднорідного середовища з двома однорідними компонентами в періоді*

Асимптотична природа усереднених методів стала зрозумілою відносно недавно. Процеси у середовищах з мікроструктурою математично можуть бути представлені із швидкоосцилюючими коефіцієнтами. Метод усереднення найзручніше застосувати до середовищ з періодичною чи квазіперіодичною структурою. Тоді для середовищ з регулярною структурою коефіцієнти є періодичними функціями. Для вивчення динаміки поведінки середовищ регулярної структури застосовують асимптотичний метод усереднення рівнянь із швидкоосцилюючими періодичними коефіцієнтами [1]. Метод був математично обґрунтований для опису механіки композитних матеріалів. Для опису динамічної поведінки багатокомпонентних середовищ на нижчому ієрархічному рівні у феноменологічному підході використовують методи суцільного середовища. При цьому вважають, що кожен мікрооб'єм перебуває в рівновазі (припущення про локальну рівновагу). Це робиться з метою введення термодинамічних величин – густини, тиску, енергії тощо. Динамічні процеси, у тому числі хвильові, характеризують ще такими величинами, як масова швидкість, швидкість поширення хвильових збурень, наприклад, ударної хвилі.

Кожен окремий компонент як неоднорідність у межах локальної рівноваги вдається описувати рівнянням руху суцільного середовища. Зрозуміло, що параметри потоку і характеристики середовища змінюються від компонента до компонента внаслідок індивідуальних властивостей компонентів, тоді як вигляд самих рівнянь руху залишається однаковим.

Обмежимося записом рівнянь руху для пласкої симетрії. Для аналізу хвильових течій всередині кожного компонента використаємо гідродинамічні рівняння, що виражають закон збереження маси та закон збереження імпульсу спільно з рівнянням стану

$$\frac{\partial V}{\partial t}-\frac{\partial u}{\partial m}=0,\quad \frac{\partial u}{\partial t}+\frac{\partial p}{\partial m}=0,\quad dp=c^2 d\rho\,, \tag{1}$$

де $V=\rho^{-1}$ – питомий об'єм; $u$ – масова швидкість; $dm=\rho_0 dx$ – масова лагранжева просторова координата. Відповідно до поставленої задачі на межах компонентів немає розривів масової швидкості і тиску: $[u]=0$, $[p]=0$. Рівняння записано в лагранжевих координатах, оскільки вони пов'язані з елементом маси середовища. Для застосування методу асимптотичного усереднення важливим є те, що в цих змінних структура стисливого середовища є сталою в процесі деформування.

Застосуємо асимптотичний метод усереднення до рівнянь руху. Незалежну змінну $m=s+\varepsilon\xi$ відповідно до методу багатьох масштабів розбиваємо на повільну $s$ і швидку $\xi$ змінні, тут $\varepsilon-$ безрозмірний період структури. Нові змінні $s$ і $\xi$ вважаємо незалежними

змінними. Тоді початкову похідну запишемо у вигляді $\frac{\partial}{\partial m}=\frac{\partial}{\partial m}+\varepsilon\frac{\partial}{\partial \xi}=0$. Повільна змінна $s$ відповідає глобальній зміні хвильових полів, а швидка змінна $\xi-$ їхній локальній зміні. Розв'язки $p,V,u$ шукаємо у вигляді рядів за степенями періоду структури $\varepsilon$ з функціями, періодичними за $\xi$, наприклад: $V(m,t)=V^{(0)}(s,t,\xi)+\varepsilon V^{(1)}(s,t,\xi)+\varepsilon^2 V^{(2)}(s,t,\xi)+\ldots$ Після усереднення за періодом структури $\xi$ отримуємо усереднену систему рівнянь [1]

$$\frac{\partial\left\langle V^{(0)}\right\rangle}{\partial t}-\frac{\partial u^{(0)}}{\partial s}=0,\quad \frac{\partial u^{(0)}}{\partial t}+\frac{\partial p^{(0)}}{\partial s}=0,\quad d\left\langle V^{(0)}\right\rangle=-\left\langle (V^{(0)})^2/c^2\right\rangle dp. \qquad (2)$$

За означенням $\langle\cdot\rangle=\int_0^1(\cdot)d\xi$. Надалі обмежимося тільки нульовим наближенням, а верхній індекс (0) опускаємо. Тиск $p^{(0)}$ і масова швидкість $u^{(0)}$ не залежать від швидкої змінної $\xi$, чого не можна сказати про питомий об'єм $V^{(0)=}V^{(0)}(\xi)$. На великому масштабі $s$ дія збурень проявляється в хвильовому русі середовища, тоді як на мікромасштабі $\xi$ дія є однорідною (безхвильовою) на всьому періоді структури середовища через те, що тиск і масова швидкість на всьому періоді є сталими величинами.

Рівняння (2) були виведені для строго періодичного середовища. Проте можна довести, що вони також будуть справедливі для середовищ з квазіперіодичною структурою. Дійсно, тиск $p$ і масова швидкість $u$ не залежать від швидкої змінної $\xi$. Тому на мікрорівні дія зовнішнього навантаження статично однорідна (безхвильова) на всьому періоді структурованого середовища. Однак на повільному масштабі $s$ ця дія проявляється у хвильовому русі середовища. На мікрорівні поведінка середовища підпорядковується тільки термодинамічним законам. Там спостерігається механічна рівновага. Водночас на макрорівні рух середовища описується законами хвильової динаміки для усереднених змінних. З математичної позиції в нульовому порядку за $\varepsilon$ розмір періоду вважаємо нескінченно малим, тобто маємо наближення $\varepsilon\to 0$. Це означає, що місцезнаходження окремих компонентів у періоді не має ніякого значення, однак масовий вміст кожного компонента повинен зберігатися. В результаті решта усереднених характеристик для середовищ з періодичною та квазіперіодичною структурою збігатиметься. Це означає, що довгохвильові рухи не відрізнятимуться між собою в періодичних, квазіперіодичних і статистично однорідних середовищах.

В [2] доведено, що структура середовища в загальному випадку вносить додаткову нелінійність. Зазначений ефект був використаний для розробки математичних основ нового методу діагностики, в якому властивості багатокомпонентного середовища вдається визначити за особливостями поширення довгих нелінійних хвиль.

Опишемо метод діагностики структури середовища, якщо відомі закономірності поширення хвиль. Ми знайшли [2], що функцію $\zeta=\zeta(V/c^2)$, обернену до шуканої $V/c^2=V/c^2(\zeta)$, можна визначити через обернене фур'є-перетворення

$$\zeta(V/c^2)=F^{-1}\left[\sum_{n=0}^{\infty}\frac{\left\langle V(Vc^{-2})^{n+1}\right\rangle}{(n+1)!\langle V\rangle}i^n q^n\right](V/c^{-2}). \qquad (3)$$

Коефіцієнти $\left\langle V(Vc^{-2})^n\right\rangle$ ($n=3,4,\ldots$) для цієї формули легко обчислити, якщо знати функційну залежність $\langle V\rangle$ від $p$ або $\left\langle V^2c^{-2}\right\rangle$ від $p$. Їх послідовно визначають з рекурентного співвідношення $\frac{d\left\langle V(Vc^{-2})^n\right\rangle}{dp}=-(n+1)\left\langle V(Vc^{-2})^{n+1}\right\rangle$, яке випливає з рівняння стану. Для

прикладу на рис. 2 представлено розрахункові результати з визначення структури шарувато-періодичного середовища, яке належним чином можна наблизити до діагностовного середовища.

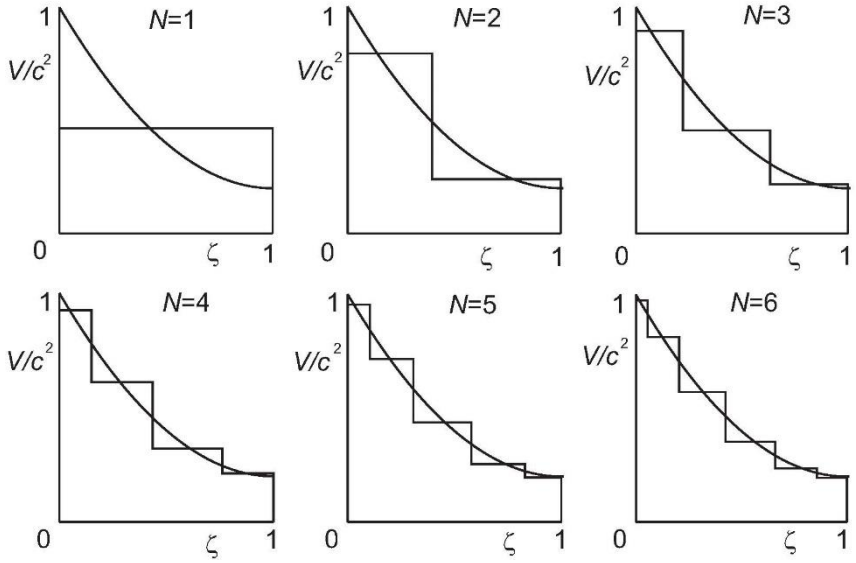



*Рис. 2 Наближення періодично-неоднорідного середовища з розподілом в періоді*

$V/c^2 = 0{,}2 + 0{,}8(1-\zeta)^2$ *шарувато-періодичними середовищами ( $N$ − число прошарків на періоді)*

Таким чином, теоретично обґрунтовано метод діагностики властивостей середовищ довгими нелінійними хвилями в рамках асимптотичної усередненої моделі структурованого середовища. Показано, що за допомогою запропонованого методу можна апроксимувати діагностовне середовище $N$ компонентним середовищем і визначити масовий вміст цих компонентів.

ЛІТЕРАТУРА

1. Vakhnenko V.O., Danylenko V.A., Michtchenko A.V., An asymptotic averaged model of nonlinear long waves propagation in media with a regular structure. *Inter. J. Non-Linear Mech.*. 1999. **34**. С. 643–654.
2. Vakhnenko V.O., Danylenko V.A., Michtchenko A.V., Diagnostics of the medium structure by long wave of finite amplitude. *Inter. J. Non-Linear Mech.*. 2000. **35**. С. 1105–1113.

**ВАХНЕНКО В’ячеслав Олексійович** – д. ф.-м. н., завідувач відділу ДТДТ, Інститут геофізики імені С.І. Субботіна НАН України, вул. академіка Палладіна, 32, Київ-03680, Україна, 03680, e mail: vakhnenko@ukr.net. ORCID: 0000-0002-1250-9563.

Наукові інтереси:

*– нелінійні еволюційні рівняння, петлеподібні солітони, інтегровність.*

**ВЕНГРОВИЧ Дмитро Богданович** – к. ф.-м. н., завідувач Відділення геодинаміки вибуху, Інститут геофізики імені С.І. Субботіна НАН України, вул. академіка Палладіна, 32, Київ-03680, Україна, 03680, e mail: vengrovich@gmail.com. ORCID: 0000-0002-1901-5697.

Наукові інтереси:

*– динаміка структурованих середовищ.*

**УДК 681.5.004.0**

ВЕРБИЦКИЙ Д.Я., ЧУБ О.И

# МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ С ПО ПРИ ПОМОЩИ СРЕДСТВ РАСПОЗНАВАНИЯ РЕЧИ

**Введение**

В современном мире технологический процесс направлен на упрощение жизни конечного пользователя всячески помогая ему автоматизируя многие его рутинные и не только, процессы.

Одним из главных аспектов управления ПО является голос, он не так распространен как другие способы ввода информации и управления ПО, проблема в том что речь это достаточно динамическое значение чтобы использовать его как инструмент взаимодействия, но с помощью NLP (Natural Language Processing - обработка естественного языка) что относиться к направлению искусственного интеллекта и математической лингвистики - более чем достижимо, яркий пример тому голосовые ассистенты призванные упростить жизнь пользователя.

В своей работе я хочу представить модель взаимодействия пользователя с ПО при помощи средств распознавания речи испольуия алгоритм word2vec (инструмент для анализа семантики естественных язык), а также другие методы векторного представления слов.

Для формирования модели также будет использоваться карта запросов исходя из окружения в котором будет применяться модель и мобильный телефон как устройство ввода так как это самое популярное устройство у среднестатистического пользователя.

Целью данной работы показать пример базового голосового помощника в качестве программы для взаимодействия при помощи голоса пользователя используя мобильное или настольное устройство.

Анализ предыдущих исследований показал, что прямых аналогов нет, но Майкл Хан, Дэн Юрафський и Ричард Футрелл занимались оптимизацией векторного соотношение для поиска похожих по значению. Их труд касается как раз алгоритмов типа world embedding, общее мнение труда это упрощение поиска похожих по звучанию но лексически различных между собой слов, они пытались выделит отдельный слой для соотношений слов этого типа лексики. Но этот алгоритм медленнее что к word2vec и не нашел своего места в голосовых ассистентов, мы же будем обращать внимание прежде всего на скорость и оптимизацию работы модели относительно поставленных.

Также при создании модели мы рассмотрим voice recogniotion (распознавание речи) - чтобы поместить слово к обработке алгоритма word2vec, оно должно пройти путь по превращению речевого сигнала в цифровую информацию для дальнейшего использования.

Для обеспечения качества мы должны следовать таким этапам:

- Обработка речи начинается с оценки качества речевого сигнала. На этом этапе определяется уровень помех и искажений.
- Результат оценки поступает в модуль акустической адаптации, который управляет модулем расчета параметров языка, необходимых для распознавания.
- В сигнале выделяются участки, содержащие речь, и происходит оценка параметров речи. Происходит выделение фонетических и просодических вероятностных характеристик для синтаксического, семантического и прагматического анализа. (Оценка информации о частях речи, форме слова и статистические связи между словами.)
- Далее параметры вещания поступают в основной блок системы распознавания - декодер. Это компонент, который сопоставляет входной речевой поток информации, хранящейся в акустических и языковых моделях, и определяет наиболее вероятную последовательность слов, которая и является конечным результатом распознавания.

**Постановка** задачи

Для комфортного формирование и редактирования модели необходима вычислительная мощность сопоставимой с мощностью сервера для стабильного поддержания 5 000 пользователей в сети. Основные требования к модели :

- Ввод команд должен упрощать и снижать количество итерации пользователя для достижения цели поддерживаемой окружением в рамках которой модель используется до минимально возможных.
- Обработка речи начинается с оценки качества речевого сигнала. На этом этапе определяется уровень помех и искажений.
- Неправильные действия пользователя не должны приводить к аварийным ситуациям.
- Модель должна защищать и не распространять личные данные пользователя без его согласия.

**Суть модели**

Модель будет осуществлять работу непосредственно на устройстве не используя при это никаких сервисов, также будет рассмотрен вариант с использованиям сервисов и его недоставки. Благодаря этому будет снижено время ответа от ассистента и будет возможна работоспособность в рамках плохо соединения или вообще его отсутствия

**Вывод**

Данная модель взаимодействия поможет упростить управление ПО, а также добавить интерактивность которая не схожа с привычными интерфейсами управления, которая также поможет людям с ограниченными возможностями.


- https://habr.com/ru/company/Voximplant/blog/446738/ - общее описание NLP для текста
- https://ru.wikipedia.org/wiki/Word2vec описание алгоритма word2vec.
- https://ru.wikipedia.org/wiki/Виртуальный_ассистент описание ассистента виртуального ассистента их примеры.
- https://nlp.stanford.edu/pubs/hahn2020universals.pdf оптимизация алгоритма word2vec
- https://hackernoon.com/stepping-into-nlp-word2vec-with-gensim-e7c54d9a450a пример word2vec
- https://en.wikipedia.org/wiki/Speech_recognition распознавание речи



**ВЕРБИЦЬКИЙ Данило Ярославович** – студент группы КУ-41 факультету компьютерных наук; Харьковский национальный университет имени В.Н. Каразина, майдан Свободы, 4, Харків-22, Украина, 61022; e-mail: danil.verbytski@gmail.com; ORCID: 0000-0003-3005-7545.

Научные интересы:

– *Программирование*
– *Управление проектами*

**ЧУБ Ольга Игоревна** – доцент кафедры теоретической и прикладной системная системотехники; ХНУ имени В.Н. Каразина Свободы, 4, Харків-22, Украина, 61022; e-mail: chubolya@gmail.ua; ORCID: 0000-0002-1216-856X.

Научные интересы:

– *Экономическая кибернетика*
– *Управление проектами*

**УДК 004.05**

ВИШНЯКОВ Є. В.

# АНАЛІЗ СКЛАДНОСТІ ТА ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗА ДОПОМОГОЮ МЕТРИКИ КАФУРИ

**Вступ**

В зв'язку з постійно зростаючою кількістю програмного забезпечення (ПЗ), стає дедалі важливішим вимірювання модульної складності програмного забезпечення для подальшої експлуатації. Неточні оцінки складності зв'язків між елементами програми є частою причиною зриву виконання робіт і додаткових витрат при розробці та супроводі програмного забезпечення. Відповідно до результатів, наданих у звіті CISQ, загальна кількість втрат, пов'язаних з експлуатацією неякісного програмного забезпечення в США в 2018 дорівнювала приблизно $2.84 трильйона доларів [1]. При супроводі, через неякісний дизайн, вартість обслуговування може значно перевищувати заплановану. Провідну роль в підвищенні якості ПЗ грають метрики оцінки різних аспектів якості. Їх систематичне використання дає змогу планувати процес удосконалення та реструктуризації програмних продуктів.

Існують різні метрики для вимірювання якості програмного забезпечення. Наприклад, широко використовувана метрика МакКейба [2] для вимірювання кількості унікальних шляхів виконання програми або метрика Холстеда [3], яка показує складність програми через кількість унікальних операторів і операндів. Ці метрики як і метрика Кафури використовують лексичний аналіз вихідного коду програми. Однак реалізація метрики Генрі-Кафури представляє більш складну задачу ніж вимірювання метрики МакКейба. Складність виникає через необхідність враховувати не тільки потік керування програмою, який також називають цикломатичною складністю, але і потік даних. Вимірювання потоку даних між модулями є складною задачею через те, що його потік є двонаправленим, на відміну від потоку команд. Двонаправленість виникає через те, що модулі не тільки приймають дані, а і повертають модулю, який викликає модулю. Існує і інший тип метрик, наприклад метрика Йен-Вінчестер [4]. Головною характеристикою у цій метриці виступає вимірювання складності між великими елементами системи. Метрика Кафури, також відома як метрика Генрі-Кафури, може бути віднесена до тієї самої групи метрик, що і метрика Йен-Вінчестер, через те, що обидві метрики використовують вимірювання зв'язків між модулями як основу своїх значень. Метрика Кафури, на відміну від вищезазначеної метрики, є більш ефективною, через те, що у ній проводиться аналіз зв'язків між всіма модулями у системі, а не тільки між елементами ієрархії.

Для визначення метрики Кафури вводиться визначення локального та глобального потоків інформації між модулями [5]. Локальний потік у свою чергу поділяється на прямий та непрямий. Локальний потік інформації між модулями А і Б існує коли:

1. Модуль А викликає модуль Б (прямий локальний потік).
2. Модуль А викликає модуль Б та модуль Б повертає значення до викликаючого модуля (непрямий локальний потік).
3. Модуль С викликає модулі А і Б та передає результат виконання модуля А до Б (непрямий локальний потік).

Глобальний потік інформації між модулями А і Б через глобальну структуру С існує, якщо модуль А розміщує інформацію у С а модуль Б дістає її з С.

Використовуючи вищезазначені визначень вводиться величина інформаційної складності процедури "I" наведений у формулі (1).

$$I = length * (fanin * fanout)^2, \quad (1)$$

де fanin – кількість локальних вхідних поток до процедури та кількість глобальних структур даних з яких процедура читає інформацію

fanout – кількість локальних вихідних потоків із процедури та кількість глобальних структур даних, з яких процедура читає дані.

length – довжина модуля вираженої у фізичній кількості строк коду.

Наведена вище формула для обчислення інформаційної складності процедури і є метрикою

Кафури. Приклад використання наведеної метрики вказано у наступній Табл. 1:

*Табл.1 Приклад обчислень функцій у модулі за допомогою метрики Кафури*

| Назва функції | fan_in | fan_out | length | I |
|---|---|---|---|---|
| func_one | 2 | 3 | 2 | 72 |
| func_two | 3 | 1 | 2 | 18 |
| separate_func_three | 1 | 1 | 1 | 1 |
| ClassOne. __init__ | 0 | 2 | 1 | 0 |
| ClassOne. func_three | 2 | 2 | 2 | 32 |
| ClassTwo. __init__ | 0 | 2 | 1 | 0 |
| ClassTwo. func_three | 2 | 2 | 3 | 48 |
| func_four | 1 | 5 | 8 | 200 |

Виходячи з даних наведених у Табл. 1 функція “func_four”, яка має найбільше сполучень, має найбільший показник складності метрики. Функція “ClassOne.__init__” ніде у коді програми не використовується, тому показник вхідних сполучень дорівнює нулю. Через те, що функція ніколи не використовується вона, таким чином, не додає складності жодній іншій функції, що в свою чергу не додає складності до загальної структури програми. Тому метрика показує складність функції як таку, що дорівнює нулю.

**Вибір цільової мови програмування**

Варто зазначити, що метрика Кафури може бути особливо корисна для проведення аналізу модульної складності програмного забезпечення, написаного на мовах програмування, які у свої основі не мають жорстких вимог стосовно структури програм, типізації та інших особливостей мови, що допомогли б спростити інтерпретацію програми вже через самі правила мови програмування. До таких мов програмування відносяться скриптові мови програмування (СМП). Більшість скриптових мов програмування не мають статичної типізації та правил стосовно того, в якій частині програми повинні бути визначені змінні, функції чи імпортовані модулі. Відсутність таких вимог може призводити до зростання складності програм та ускладнення їх супроводу. Наступним чинником вибору СМП як об’єкта аналізу модульної складності за допомогою поданої метрики, була зростаюча популярність такого типу мов програмування. СМП зараз популярні у промисловому програмуванні. Також варто зазначити, що на даний момент вже створена велика кількість програмного забезпечення з використанням СМП, через велику кількість бібліотек для спеціальних задач. Зі збільшенням кількості програмного коду, як такого який знаходиться у стані експлуатації або розробки, зростає необхідність використання спеціальних програм по вимірюванню метрик для аналізу складності та необхідності рефакторингу програм для полегшення їх подальшого супроводу.

Для вибору цільової мови програмування для реалізації оцінки метрики Кафури, було проведено порівняння декількох мов програмування для виокремлення спільних для всіх і специфічних для кожної мови особливостей які впливають на розрахунки метрики. Були визначені такі характеристики, які на думку автора, мають найбільший вплив на обчислення метрики, а саме: наявність та тип модульної структури, глобальна область видимості, процедури з побічним ефектом, необхідність попереднього декларування змінних, можливість включення в текст програми змісту інших файлів.

Було проведене порівняння наступних мов програмування за вказаними вище ознаками: C, C#, Python, Fortran, Haskell. Порівняння відбувалося виходячи з характеристик мов

програмування (МП) наведених у відповідній документації для кожної з мов. МП “Haskell”, хочаж і має модульну структуру, але не була вибрана до аналізу за допомогою наведеної метрики, через відсутність глобальної області видимості та функцій з побічним ефектом. Відсутність двох зазначених властивостей випливає з того, що ця МП є функціональною, що у свою чергу відсилає до приципів лямбда обчислення та вказує на відсутність побічних ефектів у функціях [6]. МП “C” хоча і має механізми додавання змісту одних файлів до інших, але не має модульної структури, хоча існує можливість розглядати окремі файли як модулі. Вимоги, які накладає ця мова на структуру програми є достатніми для підтримання якісної структури. МП “C#” є об’єктно-орієнтованою мовою програмування та має свою специфіку, щодо структури програми. Програма складається з класів об’єднаних у простори імен та при розгляді складності програми треба головним чином звертати увагу на використання просторів у різних файлах та інших просторах імен. Існує глобальна область видимості та побічні ефекти у функціях, через зміну параметрів переданих за посиланням. МП “Fortran” за замовчуванням не має необхідності попереднього декларування змінних. Тип змінної за замовченням визначається за першою літерою імені змінної, що може призводити до проблем із розумінням роботи програми. Ця проблема може бути компенсована використанням спеціального інструмента однозначного декларування змінних. Також у цій МП доступні механізми власноручного визначення модулів та імпортування функцій та змінних з них. Такий механізм імпортування є достатьо прозорим.

Після проведення порівняння, для подальшого опрацювання була обрана МП “Python”. У кожному модулі існує глобальна область видимості, до якої додаються імена з підключених модулів, що потенційно може призвести до ускладнення структури. Найбільшим джерелом складнощів при роботі з механізмом імпорту є можливість доступу до оригінального модуля через інші модулі, що імпортують оригінал. Ще одним аргументом до розгляду цієї мови є наявність механізму, який часто призводить до погіршення якості структури програми, а саме динамічна типізація. У контексті вимірювання потоків даних у МП “Python”, варто зазначити факт того, що всі параметри передаються за посиланням і можуть бути змінені у функції та ці зміни будуть відображені у викликаючій функції. Такий механізм створення побічних ефектів, впливає на потоки даних у програмі у сторону збільшення останніх. На даному етапі виконання роботи, було проведене порівняння МП за визначеними критеріями та встановлена цільова МП.

ЛІТЕРАТУРА

1. Krasner H. The cost of poor quality software in the US: A 2018 report. Consortium for IT Quality Software. 2018, p. 5.
2. T. J. McCabe. A Complexity Measure. IEEE Transactions on Software Engineering, vol. SE-2, no. 4, 1976, pp. 308-320
3. Halstead, Maurice H. Elements of Software Science. Amsterdam: Elsevier North-Holland, Inc. ISBN 0-444-00205-7, 1977, p. 127.
4. B. H. Yin and J. W. Winchester. “The establishment and use of measures to evaluate the quality of software”. Proceedings of the Software Quality and Assurance Workshop, 1978, pp. 45-52.
5. S. Henry and Dennis Kafura “Software structure metrics based on information flow” IEEE transactions on software engineering, Vol. SE-7, NO-5, 1981, p. 20.
6. Greg Michaelson “An introduction to functional programming through lambda calculus” Dover Publications Inc. Mineola, New York, 1989, p. 4.

**ВИШНЯКОВ Є. В.** – студент кафедри моделювання систем і технологій; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: YVysh@protonmail.com; ORCID: 0000-0003-3093-1335.

Наукові інтереси:

– *Тестування та верифікація програмного забеспечення.*

**УДК 004.7(075)**

ВОЕВОДА В.Р., БЕРДНИКОВ А.Г.

# МОДЕЛЬ ИНТЕГРАЛЬНОГО КАНАЛА В АВТОМАТИЗИРОВАННОЙ СИСТЕМЕ УПРАВЛЕНИЯ

**Введение**

Необходимыми условиями для нормального функционирования современных телекоммуникационных систем являются стандартизация, логическая корректность и совместимость процедур обмена.

Выполнение этих условий в полной мере обеспечивают интегральные цифровые сети (Integrated Services Digital Network – ISDN), которые дают возможность передавать в единой форме дискретную информацию (данные) и аналоговые сообщения.

Цифровые сети интегрального обслуживания ISDN широко внедряются как средство передачи любого вида информации стандартизированным способом.

Одним из известных и востребованных способов интеграции является объединение сетей различного функционального назначения, позволяющее создать, например, интегральную технологическую сеть, которая объединяет функции телефонной сети и сети передачи данных. Активное развитие современных сетевых технологий позволяет решить задачу интеграции достаточно эффективно и экономно. Поэтому разработка модели интегрального канала передачи информации представляется актуальной.

Целью данной работы является создание модели интегрального канала передачи в автоматизированных системах управления технологическими процессами (АСУ ТП) на базе апробированных решений технологии Cisco.

**Постановка задачи**

Интеграция в цифровых сетях связи может проводиться разными способами на различных уровнях и позволяет:

- на уровне терминальных устройств – использовать один тип терминального устройства для выполнения различных функций;
- на уровне каналообразующего и коммутационного оборудования – использовать цифровую систему коммутации в качестве оборудования каналообразования;
- на уровне объединения сетей различного назначения – создать интегральную сеть, реализующую функции разнообразных сетей (например, интегральная технологическая сеть, объединяющая в себе функции телефонной сети и сети передачи данных или оперативно-технологической и общетехнологической связи;
- на уровне предоставления услуг – сконцентрировать все функциональные возможности обслуживания и передать их по абонентской линии с использованием одного и того же абонентского номера.

В данной работе стоит задача создания модели интегрального канала передачи информации на базе технологии Cisco. Это позволит снизить стоимость разработки проекта сети при гарантированной высокой надежности оборудования авторитетной компании, расширить функциональные возможности при обслуживании всех видов связи в системе управления и повысить производительность при обработке информации

Для решения поставленной задачи использованы возможности приложения Cisco Packet Tracer по моделированию компьютерных сетей с использованием протокола Frame Relay по предоставлению услуг интеграции каналов в интегральной сети ISDN.

Приложение Cisco Packet Tracer, описанное в [3], является средой для разработки моделей сети, использующей технические решения технологии Cisco. Packet Tracer предоставляет функции моделирования, визуализации, разработки сетей, а также облегчает изучение сложных технологических принципов.

В интегральном канале протокол ISDN принимает информацию (пакеты) отправителей и фиксирует их, затем, протокол Frame Relay, изымает из этих пакетов кадр, в котором находит номер идентификатора DLCI (Data Link Connection Identifier), содержится адрес получателя

пакета. Frame Relay – протокол канального уровня сетевой модели OSI. Технология ISDN поддерживает на прикладном уровне факсимильную связь со скоростью передачи 64 Кбит/с (V = 64 Кбит/с), телексную связь, V = 9600 бит/с, видеотекс, V = 9600 бит/с и ряд других служб связи. Место протокола Frame Relay в общей структуре интегральной сети ISDN показано на рис.1.

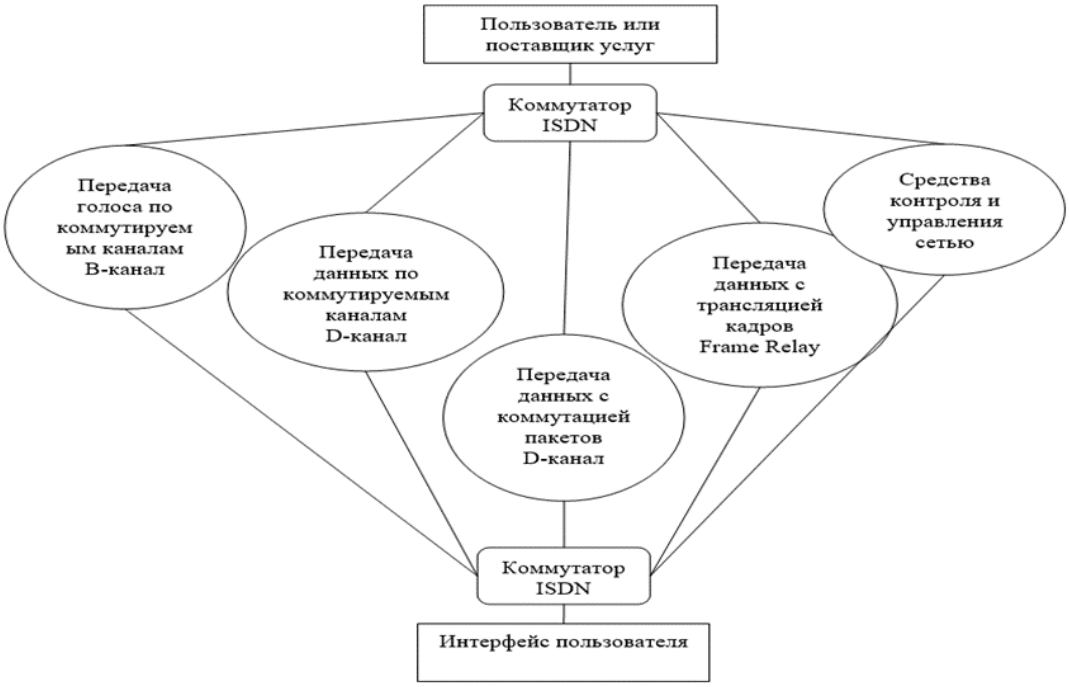



*Рис.1.*

Данные могут поступать по разным потокам и это задача именно коммутатора – определять поток по которому будут поступать данные. Есть два процесса:

Демультиплексирование – разбиение суммарного потока на несколько потоков из составляющих этого же одного потока.

Мультиплексирование(агрегирование) – образование из нескольких отдельных потоков общего агрегированного потока, который передается по одному физическому каналу связи.

Операции мультиплексирования/демультиплексирования имеют такое же важное значение в любой сети, как и операции коммутации, потому что без них пришлось бы для каждого потока предусматривать отдельный канал, что привело бы к большому количеству параллельных связей в сети и свело бы на нет все преимущества неполносвязной сети.

Функционирование интегральной сети ISDN с использованием технологии Frame Relay поясняется схемой на рис 2.

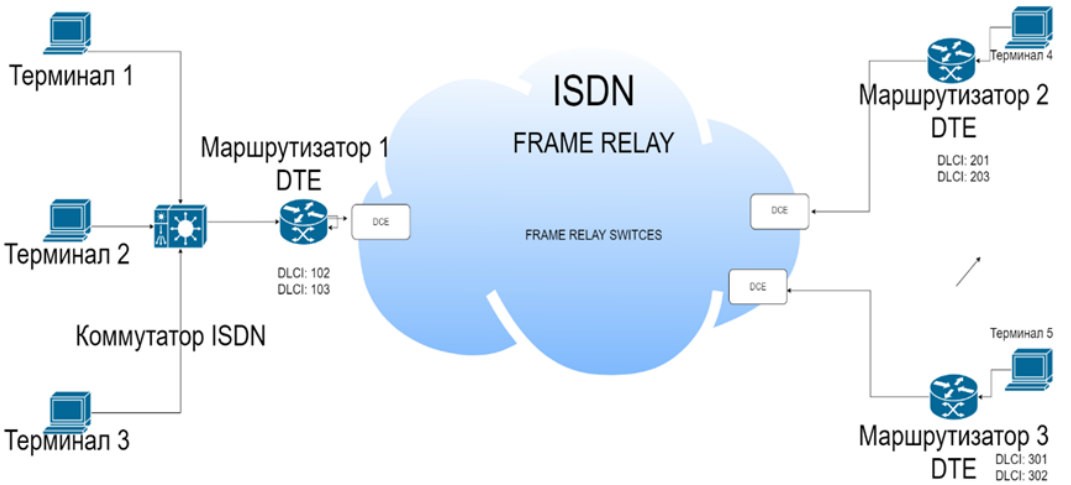



*Рис.2.*

Для передачи информации с Терминала 1-3 в адрес Терминала 4 или Терминала 5 через сеть ISDN необходимо, чтобы все маршрутизаторы обеспечивали инкапсуляцию пакетов с соответствующим значением DLCI в протокол Frame Relay, далее в «облаке» Frame Relay Switches сообщение обрабатывается провайдерским устройством DCE (Data Communication Equipment) и направляется с заданной скоростью на соответствующее устройство пользователя DTE (Data Terminal Equipment).

Технология ретрансляции кадров Frame Relay предназначена передачи данных между узлами в сегменте одной локальной сети.

Исходная структура сети, на базе которой производится разработка модели интегрального канала, описана в официальном курсе изучения Cisco, CCNA 4, Frame Relay. В данном источнике находится настройка Frame Relay, в которой показывается способ передачи информации и как работает этот протокол. Данная настройка является лишь частью для создания полной рабочей компьютерной сети. Задача заключается в создании кампусной офисной сети с использованием протокола Frame Relay. Офис разделен на 4 отдела и имеет одну бытовую комнату. Нехватка оборудования, слишком простые маршруты передачи данных, все это требовалось переделывать, иначе не получилось бы достичь цели, которую мы поставили. Чтобы решить данную задачу, потребовалось добавлять новые подсети в соответствии с отделами, которые расположены в офисе, также настраивать конфигурацию, чтобы каждый отдел был связан с другим и для этого пришлось настраивать соответствующую маршрутизацию. В соответствии с поставленной задачей, было добавлено оборудование, настроены подсети для каждого отдела и соответствующая маршрутизация между отделами, в результате получилась кампусная офисная сеть, связанная между отделами, задача решена.

Процесс разработки модели на базе данной структуры, включает несколько этапов, а именно:

1) Построение модели сети.

Расположение составляющих компьютерной сети (ПК, коммутаторы, роутер, маршрутизаторы, сервера, выбор кабеля, облако провайдера)

2) Настройка конфигурации оборудования сети.

Назначение IP-адреса, DNS-сервер, шлюз по умолчанию соответствующему оборудованию пользователя (PC0-PC13, Laptop0, Laptop1, Server0-Server2). Настройка коммутаторов (Switch0-Switch2), соединение их с оборудованием пользователя. Настройка роутера, назначение IP-адреса и подключение соответствующих ПК к роутеру по беспроводному соединению (Wi-Fi). Настройка маршрутизатора (Router0, Router2, Router3), назначение IP-адреса соответствующему разъему подключения, назначение рабочей подсети.

3) Конфигурация Frame Relay.

Установка соединения между Cloud0 и Router0, Router2. Настройка маршрутизации в Router0 и Router2 через Cloud0.

4) Конфигурация IP-туннеля.

В Router3 установка IP-адреса туннеля и назначение рабочей подсети. Точно также установка IP-адреса туннеля и назначение рабочей подсети в Router0.

5) Настройка DNS сервера.

В качестве DNS-сервера используется Server0-Server2. Настройка конфигурации Server0-Server2 и создание HTTP протокола.

6) Тестирование.

Проверка соединений с помощью команды “ping” на всех PC, проверка маршрутизации между Router0, Router2, Router3. Проверка работы DNS-сервера.

В результате проведенной модернизации исходной сетевой структуры получена интегральная офисная сеть ISDN (рис. 3), использующая технологию трансляции кадров.

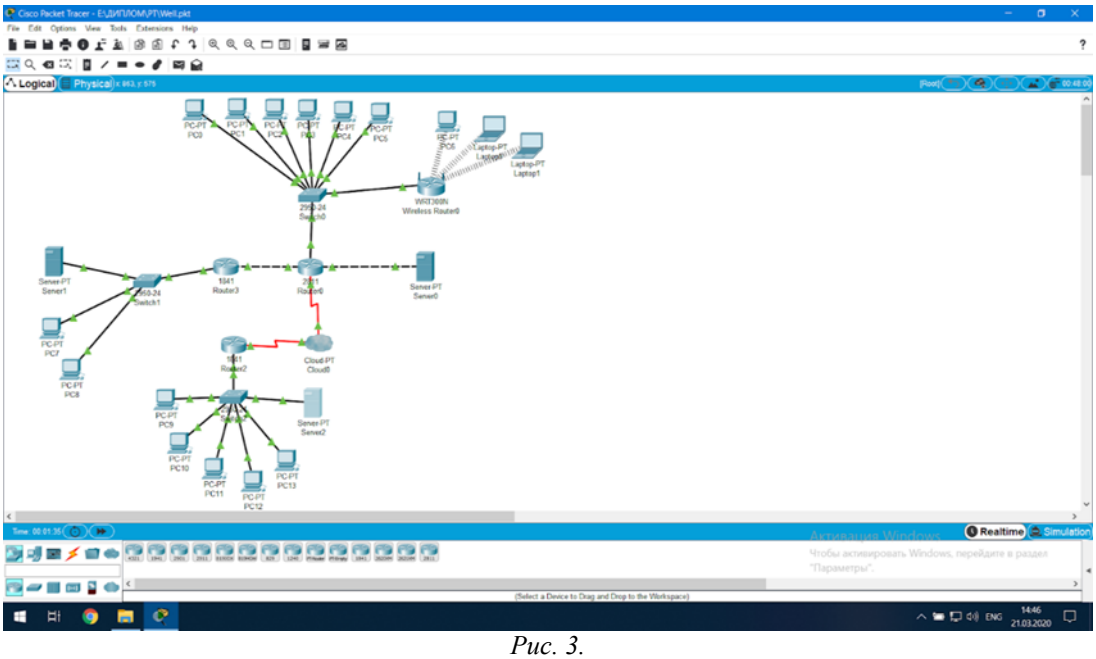


*Рис. 3.*

Разработанная модель сети представляет собой офисную сеть, включающую 4 подсети, в каждой из которых сетевое оборудование (маршрутизаторы, коммутаторы, сервера, роутеры, облако провайдера и ПК (14шт PC-PT, 2шт. Laptop-PT, 3шт. Server PT, 1шт. Cloud-PT, 2шт. маршрутизаторов 1841, 1шт, маршрутизатор 2811, 3шт. коммутаторов 2950-24, 1шт. роутер WRT300N) привязано к своему коммутатору и имеет возможность обмениваться с другими подсетями по интегральному каналу.

Таким образом, благодаря использованию технологии Cisco для построения интегрального канала ISDN в сети Frame Relay получены сетевые решения, позволяющие с минимальными затратами построить локальную вычислительную сеть без выполнения сложных и дорогостоящих проектных работ.

С учетом использования апробированных программных и технических решений технологии Cisco обеспечивается повышение надежности функционирования интегральных каналов передачи информации при сохранении требуемой производительности сети.

Модель может быть рекомендована для использования в учебном процессе для проведения практических видов занятий по учебной дисциплине «Компьютерные сети», а также в фирмах, специализирующихся на проектировании телекоммуникационных систем.

ЛІТЕРАТУРА

1. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы / В.Г. Олифер, Н. А. Олифер. // Учебник для вузов. – 4-е изд. – СПб.: Питер, 2010. – 944с.: ил.
2. Таненбаум Э. Компьютерные сети / Э. Таненбаум, Д.Уэзеролл. – 5-е изд. – СПб.: Питер, 2012. – 960 с.
3. Одом, Уэнделл. Официальное руководство по подготовке к сертификационным экзаменам CCENT/CCNA ICNDI, 2-е изд.: Пер. с англ. – М.: ООО “И. Д. Вильямс”, 2010. – 672с.: ил. – Парал. тит. Англ

**ВОЕВОДА Вячеслав Романович –** студент группы КУ-41 факультета компьютерных наук; Харьковский национальный университет имени В.Н. Каразина, майдан Свободы, 4, Харків-22, Украина, 61022; e-mail: voevodavyacheslav@gmail.com; ORCID: 0000-0002-3370-2485.

**БЕРДНИКОВ Анатолий Георгиевич** – к. т. н., доцент, доцент кафедры теоретической и прикладной системотехники; факультету компьютерных наук; Харьковский национальный университет имени В.Н. Каразина, майдан Свободы, 4, Харків-22, Украина, 61022; e-mail: tps@karazin.ua.

Научные интересы: - *Программирование, моделирование, проектирование.*

**УДК 004.056.52**

ГАРМАШ Д.В., МАЛЄЄВА Г.А., ГОРБЕНКО І.Д.

# ПОРІВНЯННЯ ПЕРСПЕКТИВНИХ АЛГОРИТМІВ ЕЛЕКТРОННОГО ПІДПИСУ НА ОСНОВІ MQ-ПЕРЕТВОРЕНЬ

Наразі в процесі розроблення посквантових алгоритмів цифрового підпису певне визнання отримали криптографічні перетворення в квадратичних полях[1,2]. Попередній аналіз вказаних джерел дозволив зробити висновок про перевагу алгоритмів Rainbow та LUOV. Метою цієї доповіді є їх порівняльний аналіз.

В першу чергу потрібно звернути увагу на основні принципи роботи з мультиваріативними квадратичними перетвореннями.

Мультиваріативні перетворення грунтуються на використанні кінцевого поля $\mathrm{k}-\frac{\mathrm{GF}[2]}{\mathrm{x}^2+\mathrm{x}+1}$, в якому $2^2$ елементів.

Для спрощення вони позначаються множиною чисел $\{0,1,2,3\}$. Причому 0 є нулем у полі k, 1 є одиницею, 2 є поліномом x, а 3 є поліномом 1+x. По суті, коефіцієнти квадратичних поліномів приймають значення над полем $4=2^2$. В якості квадратичних поліномів вибираються такі

$$\mathrm{G}_0=(\mathrm{x}_1,\mathrm{x}_2,\mathrm{x}_3)=1+\mathrm{x}_2+2\mathrm{x}_0\mathrm{x}_2+3\mathrm{x}_1^2+3\mathrm{x}_1\mathrm{x}_2+\mathrm{x}_2^2$$

$$\mathrm{G}_1=(\mathrm{x}_1,\mathrm{x}_2,\mathrm{x}_3)=1+3\mathrm{x}_0+2\mathrm{x}_1+\mathrm{x}_2+\mathrm{x}_0^2+\mathrm{x}_0\mathrm{x}_1+3\mathrm{x}_0\mathrm{x}_2+\mathrm{x}_1^2 \quad (1)$$

$$\mathrm{G}_2(\mathrm{x}_1,\mathrm{x}_2,\mathrm{x}_3)=3\mathrm{x}_2+\mathrm{x}_0^2+3\mathrm{x}_1^2+\mathrm{x}_1\mathrm{x}_2+3\mathrm{x}_2^2$$

Вважається, що наведені поліноми є багатовимірними поліномами над кінцевим полем та вони можуть застосовуватись в багатофакторній криптографія при розробці асиметричних криптографічних примітивів [1].

Алгоритм Rainbow побудований з використанням багатовимірних перетворень. Цей механізм базується на схемі Ойла та Вінежера. Сутність таких схем полягає у тому, що існують два типи змінних – vinegar та oil. Перші при обчисленні центрального відображення обираються випадковим чином, а другі використовуються як значення геш-функції від повідомлення. Особливістю схему UOV є те, що зазвичай кількість $v$ змінних vinegar повинна складати $v=2o...3o$ від кількості $o$ змінних oil.

Алгоритм LUOV є також, на наш погляд, одним з основних кандидатів на стандарт ЕП для пост-квантового періоду. Він ґрунтується на багатовимірній криптографії. Його стійкість грунтується на складності рішення систем багатовимірних поліноміальних рівнянь. Аналіз показує, що особливістю багатоваріантної криптографії є допустима, у порівнянні з іншими методами, складність постквантового ЕП. По суті вона швидка і вимагає лише помірних обчислювальних ресурсів, що робить його привабливим для застосування у недорогих пристроях. Алгоримтм LUOV є простим удосконаленням схеми UOV, що значно зменшує розмір відкритих ключів [2].

Аналіз показав, що в алгоритмі ЕП LUOV використовується функція одностороннього відображення $\rho: F_{2^r}^n \to F_{2^r}^m$ яка є багатоваріантною квадратичною поліноміальною платформою змінних $n=m+v$ з коефіцієнтами у бінарному підполі $F_2 \subset F_{2^r}$. В алгоритмі основою є факторизація типу $P=F\circ T$, де $T: F_{2^r}^n \to F_{2^r}^n$ є зворотнє відображення .

Анліз показав, що , алгоритм LUOV може бути використана в двох режимах. Одним з варіантів є звичайний режим ЕП, де повідомлення автентифікуються шляхом додавання підпису. Інший варіант - це режим відновлення повідомлень, який можна використовувати для

зменшення розміру пари підпис та повідомлень. У режимі відновлення повідомлення (частина) повідомлення не передаються, але відновлюються з підпису.

Можна зазначити, що алгоритм LUOV має малий розмір підпису, нескладний принцип обчислення, визначений підпис та гнучкий стан, але великий розмір ключів. У той же час, алгоритм LUOV має ще більший розмір ключів та розмір підпису, він є більш складним, але швидшим.

*Табл.1Розміри ключів та підпису алгоритмів LUOV та Rainbow.*

| **Submission** | **sk (байт)** | **pk (байт)** | **sign (байт)** |
|---|---|---|---|
| LUOV | 32 | 7 536 | 1 746 |
| LUOV | 32 | 19 973 | 3 184 |
| LUOV | 32 | 40 248 | 4 850 |
| LUOV | 32 | 100 989 | 521 |
| LUOV | 32 | 15 908 | 319 |
| LUOV | 32 | 46 101 | 441 |
| Rainbow | 100 209 | 152 097 | 64 |
| Rainbow | 114 308 | 163 185 | 78 |
| Rainbow | 143 385 | 192 241 | 104 |
| Rainbow | 409 463 | 564 535 | 112 |
| Rainbow | 537 781 | 720 793 | 156 |
| Rainbow | 376 141 | 565 489 | 92 |

ЛІТЕРАТУРА

1. ETSI White Paper №8: Qantum safe cryptography and security. – 2015
2. Горбенко І.Д. Аналіз проблем криптографічного захисту інформації у пост-квантовий період та можливі шляхи їх вирішення// Матеріали V-ої міжнародної науково-технічної конференції «Захист інформації і безпеки інформаційних систем». – Львів, 2016 (02-.06 – 03.06). – С. 52.

**ГАРМАШ Дмитро Васильович** – аспірант факультету комп’ютерних наук, кафедра Безпеки інформаційних систем і технологій; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: donni.dima@gmail.com.

Наукові інтереси:

*– прикладна криптологія, криптографічні системи та протоколи.*

**МАЛЄЄВА Ганна Андріївна** – аспірант факультету комп’ютерної інженерії та управління, кафедра Безпеки інформаційних технологій; Харківський національний університет радіоелектроніки, проспект Науки, 14, Харків, Україна, 61166; e-mail: hanna.malieieva@nure.ua.

Наукові інтереси:

*– криптографія, криптоаналіз та їх застосування з метою захисту інформації..*

**ГОРБЕНКО Іван Дмитрович** – д. т. н., професор, головний конструктор АТ «Інститут інформаційних технологій», викладач кафедри Безпеки інформаційних систем і технологій, Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: halina@iit.kharkov.ua.

Наукові інтереси:

*– прикладна криптологія, криптографічні системи та протоколи, проектування та розробка систем, комплексів та засобів криптографічного захисту інформації.*

**УДК 004.9 + 681.51+ 721**

ГЕРАСИМЕНКО Л.В.

# МОДЕЛЮВАННЯ РОЗТАШУВАННЯ З УРАХУВАННЯМ ВИМОГ САНАЦІЇ

Проблема розташування з урахуванням вимог санації розглядається на прикладі організації оптимального планувального рішення будинків. Розробляється модель, яка враховує додаткові умови, викладені в документах предметної області. Запропоновано реалізацію алгоритму розв'язання задачі у вигляді програмного пакету Revit, Lara, ArchiCAD.

**Ключові слова:** математична модель, нерегулярне розміщення геометричних об'єктів, планувальне рішення, санація.

**Вступ**

Гігантський розмах будівництва в усіх країнах забезпечує особливі вимоги до якості проектів. Маються на увазі не тільки естетичні, планувальні та конструктивні переваги проектів, а й забезпечення оптимального мікроклімату приміщень як одного з основних факторів, що впливають на здоров'я і працездатність людей.Інсоляція а саме висвітлення прямими сонячними променями, робить істотний вплив на мікроклімат приміщень.

Організація моделювання плану будівель із застосуванням технології Revit відноситься до завдань безперервного математичного програмування. Зокрема до задачі нерегулярного прямокутного розміщення геометричних об'єктів із змінними метричними характеристиками на ділянці з урахуванням вимог санації. Для знаходження точного рішення, подібного роду завдання пропонують застосування технологій Revit, Lara, ArchiCAD.

**Матеріали і методи досліджень**

Вивчення питань природного освітлення та інсоляції, а також викладання пов'язаних з ними дисциплін ведеться на кафедрах різних університетів і інститутів по всьому світу. Розробляються альтернативні методи і алгоритми розрахунку природної радіації, засновані на сучасних знаннях фізиків, біологів, інженерів і програмних розробників. У світі існує кілька комп'ютерних програм для розрахунку інсоляції: японська MicroShadow for ArchiCAD; вітчизняного виробництва -Lara, CITIC: Солярис.

У публікації наведено вирішення проблеми оптимізації прямокутного розміщення геометричних предметів з модифікується метричними параметрами.

**Результати дослідження**

Аналіз вітчизняного та зарубіжного нормування інсоляції виявлено, що в будівництві будівель застосовуються часовий показник і психо-емоційна оцінка.

Були проведені розрахунки тривалості інсоляції в будь-який день року, з урахуванням географічного положення та были зроблені такі розрахунки:

- розрахунок інсоляції майданчика;
- розрахунок інсоляції в точці;
- створення кадрів анімації руху тіні взаданий день;
- редагування геометрії об'єктів міського простору з урахуванням результатів розрахунків інсоляції в інтерактивному режимі.

Розрахунковим майданчиком являється як: площина землі, план будівлі так і фасаду. Площина землі може містити модель будівлі. Дані розрахунки були зроблені для прямокутної в плані будівлі, але можна зробити розрахунки для будь-якої криволінійної форми.

Розрахункова точка може бути як розрахункової точкою вікна, так і будь-який інший точкою простору. В результаті розрахунку інсоляції було визначено:

- визначено інтервали інсоляції в точці, час початку і закінчення кожного інтервалу, сумарну тривалість інсоляції;

- визначено відповідність ПІ в точці нормативної ПІ;
- сформувано таблицю результатів з поясненням параметрів розрахунку і нормування;
- виконати научно графічну візуалізацію результатів розрахунку з відображенням на ній секторів інсоляції.

Розрахунок інсоляції майданчиків проводиться для кожної точки майданчика по тій же схемі, що і для вікон. В кінці розрахунку визначається кількість точок, тривалість інсоляції яких відповідає нормам (якщо інсолюються мінімум половина точок - інсоляція майданчика виконується).Результати розрахунку інсоляції представляються у вигляді звітних таблиць, графіків інсоляції вікон, Інсоляціонний кутів. Тіні від об'єктів, затінюють межі об'єктів можна подивитися на екрані або винести в друкований звіт (безпосередньо на принтер або в графічні файли). Для бібліотечних об'єктів існує можливість формування в файл MS Word докладного звіту, що містить результати розрахунків і обґрунтовані висновки про виконання норм інсоляції в кімнатах і квартирах.

Отримані результати розрахунку часу інсоляції повинні бути інтерпретовані експертом в укладанні на відповідність нормам. Тут треба звернути увагу на наступне. Норми за часом інсоляції іноді змінюються. Як правило, це відбувається в сторону зменшення, що дозволяє ущільнювати забудову. У нормативних і ще діючих документах є суперечності щодо необхідної величини часу інсоляції. Вимоги щодо нормативного часу інсоляції можуть залежати від місцевого законодавства, деяких приміток в нормативних документах, наприклад, історична забудова, центр міста і т. д.

Питання про забезпечення інсоляції дуже важливе на самих різних етапах нового будівництва або реконструкції.

ЛИТЕРАТУРА

1. Расчет инсоляции зданий. Штейнберг А.Я. 1975
2. Расчет инсоляции зданий. Киев, «Буддвельник», 1975
3. http://arx.novosibdom.ru/node/186
4. https://bouw.ru/term/insolyatsiya

**ГЕРАСИМЕНКО Лада Володимирівна** – студентка, Харківский національний університет будівництва та архітектури, (вул. Сумська 40, Харків, Україна; 61195); e-mail: Lada.gerasimenkoo@gmail.com; ORCID: 0000-0002-6755-4193.

**УДК 004.032.26:548.734.032**

ГОЛУБНИЧИЙ В.О, СТРІЛЕЦЬ В.Є.

# МЕТОД РОЗПІЗНАВАННЯ ТА АНАЛІЗУ РЕНТГЕНОГРАМ ГРУДНОЇ КЛІТИНИ НА ОСНОВІ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

**Проблема аналізу рентгенограм грудної клітини**

Використання рентгенограм грудної клітини є одним з найкращих способів діагностування легеневих патологій, але невчасна та неточна постановка діагнозу може привести до смерті пацієнта, незважаючи на якісний знімок. Це може бути зумовлено багатьма причинами, серед яких має місце як невеликий досвід лікаря, так і надзвичайно велике навантаження, якому лікарі піддаються. На допомогу в таких випадках приходять системи комп'ютерної діагностики, які здатні полегшити роботу радіологам шляхом автоматизації процесу аналізу та обробки рентгенограм. Такі системи повинні мати достатню швидкість роботи та гарну точність, щоб забезпечити відповідну якість роботи та спростити проблему постановки вірного діагнозу замість того, щоб додати додаткові складності в роботу радіологів.

Робота спрямована на розробку систему комп'ютерного діагностування легеневих патологій шляхом автоматизованого аналізу рентгенограм, що буде задовольняти потреби як пацієнтів, так і лікарів. Останнім часом було зроблено значний прогрес у розробці таких систем [1], але варто зазначити, що процес створення алгоритму діагностики містить декілька підводних каменів: патології часто візуально схожі одна з одною і тому навіть найкращі радіологи можуть допустити помилку у своїх судженнях [2]. Іншою проблемою є недостатня кількість даних, які потрібні для побудови високоточної системи комп'ютерного діагностування патологій. З появою нових наборів даних, таких як CheXpert [3], зменшується негативний вплив проблеми, але, на жаль, не нівелює його повністю. Варто зазначити, що значна кількість дослідників концентрується на одній певній хворобі [4, 5] замість того, щоб розробити один універсальний детектор, який потребує значних зусиль та великої кількості досліджень. На відміну від них, алгоритм діагностики, який пропонується, є спроможним знаходити декілька патологій одночасно.

Таким чином задачу розпізнавання та аналізу рентгенограм можливо сформулювати як задачу багатоміткової класифікації, де вхідними даними є нормалізована рентгенограма грудної клітини, а вихідними даними є бінарний вектор ознак, де наявність одиниці або її відсутність у певній комірці говорить про наявність або відсутність відповідної патології. Задача класифікації зображень належить до задач комп'ютерного зору та може бути вирішена шляхом використання машинного навчання з учителем, а саме його новітнім підрозділом – глибоким навчанням шляхом використання згорткових нейронних мереж.

**Використання згорткових нейронних мереж для аналізу рентгенограм**

Згорткові нейронні мережі це – нейронні мережі, які поєднують три архітектурні ідеї, щоб забезпечити в тій чи іншій мірі стійкість до зміщення, зміни розміру та спотворення даних: локальне рецепторне поле, спільні параметри (реплікація параметрів) та просторова підвибірка даних. Першою згортковою мережею є LeNet-5 [6], ідеї та архітектурні особливості якої лягли в основу всіх наступних архітектур згорткових мереж. Прикладом таких ідей є послідовне застосування згорткових шарів разом з агрегаційними шарами, зменшення просторових розмірів мап ознак зі збільшенням їхньої кількості та наявність шару щільно з'єднаних нейронів, які формують кінцевий результат роботи мережі. При цій конфігурації, згорткові шари можливо розглядати як екстрактор важливих ознак із зображення, а щільно з'єднані шари як класифікатор, що обробляє високорівневу інформацію, отриману від згорткових шарів. Наступним етапом розвитку мереж цього типу є створення AlexNet [7], що відрізнялася від LeNet більшою глибиною мережі, використанням іншої функції активації та більш складним механізмом тренування. Успіх AlexNet на ImageNet-2012 призвів до збільшення зацікавленості дослідників до згорткових мереж та появи більш складних їх архітектур [8, 9]. Результатом цього розвитку є

проходження межі людської точності у роботі [10] Kaiming He та інших. Таким чином, згорткові мережі зарекомендували себе як найкращий засіб для аналізу зображень, що й вплинуло на вибір алгоритму для дослідження. У роботі використовувалась архітектура щільних нейронних мереж DenseNet [11], що була запропонована як подальший розвиток ResNet [9] та ResNetV2 [12].

Головною відмінністю DenseNet від інших архітектур є використання щільних блоків (Dense Block) та блоків передачі даних між ними (Transition Block). Мережа накопичує загальну інформацію про зображення у щільних блоках шляхом з'єднання всіх наступних мап ознак із попередніми, це дозволяє спростити проходження градієнта функції похибки вверх по мережі та дає можливість використовувати як високо-, так і низькорівневу інформацію із усіх шарів мережі. Блоки передачі відіграють роль агрегатора, що стискає просторові розміри мап ознак та зменшує їхню кількість для більш ефективного використання наявних ресурсів. Ми використовуємо модифікацію щільної мережі під назвою DenseNet-BC-169 з незмінними початковими гіперпараметрами. Єдиною відмінністю різних видів DenseNet-BC є різна кількість згорткових шарів у щільних блоках.

Для більшої гнучкості в дослідженні та здатності змінювати мережу в процесі роботи була створена мережа DenseNet для проведення експериментів на основі фреймворку машинного навчання PyTorch.

**Збір даних для тренування**

Для тренування згорткових нейронних мереж потрібна значна кількість навчальних даних, щоб запобігти виникненню явища, коли нейронна мережа починає сприймати випадковий шум в рентгенограмах як важливі ознаки. Для цього був використаний набір даних CheXpert, що містить в цілому 224316 рентгенограмм 65240 пацієнтів. Усі зображення поділені на 14 категорій (табл. 1).

Цей набір даних має декілька особливостей, які потрібно врахувати перед початком роботи з ним:

– мітки були отримані з використанням засобу автоматичної обробки тексту за допомогою сформованих раніше правил;

– засіб автоматичної обробки тексту не ставить мітку, якщо патологія не була згадана в радіологічному дослідженні, що говорить про її відсутність;

– набір даних містить рентгенограми, що були зроблені з різних сторін тіла;

– мітки містять не лише 0 та 1, тобто наявність або відсутність патології, а й мітки невпевненості (-1);

– категорії надзвичайно зміщені відносно одна одної за кількістю елементів;

– кількість даних з негативними мітками значно перевищує кількість даних з позитивними мітками.

*Табл. 1. Схема набору даних*

| Pathology | Positive (%) | Uncertain (%) | Negative (%) |
|---|---|---|---|
| No Finding | 16627 (8.86) | 0 (0.0) | 171014 (91.14) |
| Enlarged Cardiom. | 9020 (4.81) | 10148 (5.41) | 168473 (89.78) |
| Cardiomegaly | 23002 (12.26) | 6597 (3.52) | 158042 (84.23) |
| Lung Lesion | 6856 (3.65) | 1071 (0.57) | 179714 (95.78) |
| Lung Opacity | 92669 (49.39) | 4341 (2.31) | 90631 (48.3) |
| Edema | 48905 (26.06) | 11571 (6.17) | 127165 (67.77) |
| Consolidation | 12730 (6.78) | 23976 (12.78) | 150935 (80.44) |
| Pneumonia | 4576 (2.44) | 15658 (8.34) | 167407 (89.22) |
| Atelectasis | 29333 (15.63) | 29377 (15.66) | 128931 (68.71) |
| Pneumothorax | 17313 (9.23) | 2663 (1.42) | 167665 (89.35) |
| Pleural Effusion | 75696 (40.34) | 9419 (5.02) | 102526 (54.64) |
| Pleural Other | 2441 (1.3) | 1771 (0.94) | 183429 (97.76) |
| Fracture | 7270 (3.87) | 484 (0.26) | 179887 (95.87) |
| Support Devices | 105831 (56.4) | 898 (0.48) | 80912 (43.12) |

Перед тим як почати роботу з CheXpert, всі мітки невпевненості (-1) були замінені на позитивні мітки (1) згідно з рекомендаціями авторів набору даних, встановлені негативні мітки там, де міток не було. Весь набір даних був розбитий на 3 частини: тренувальний (221600 рентгенограм), кросс-валідаційний (1024 рентгенограм) та тестовий (1024 рентгенограм) набори даних. Тренувальний набір використовувався для знаходження оптимальних параметрів, кросс-валідаційний набір потрібен для тестування гіперпараметрів, тестовий набір даних був створений для оцінки кінцевої моделі.

**Підготовка мережі та вибір метрики оцінювання ефективності**

Для забезпечення якісних результатів тренування рентгенограми були нормалізовані шляхом віднімання від кожного каналу їхнього середнього значення та діленням на стандартне відхилення, які були взяті із набору даних ImageNet. Всі зображення в подальшому були зжаті до розмірів (320x320). Як алгоритм навчання був обраний Nadam [13] завдяки його швидкості сходження, малій кількості шуму при тренуванні та здатності заглянути на ітерацію вперед для коригування наступного кроку. Параметри $\beta_1$ та $\beta_2$, що відповідають за швидкість накопичення були залишені зі своїми початковими значеннями. Для регуляризації мережі була обрана l2-регуляризація з параметром $\lambda$=1e-5 та алгоритм Dropout [14] з параметром p=0.9. Коефіцієнт швидкості навчання $\alpha$ був встановлений на рівні 1e-3. Враховуючи наявну кількість даних, нейронна мережа тренувалась протягом 5 епох. Всі гіперпараметри були підібрані експериментально шляхом пошуку оптимального значення між 2 межами.

Для оцінювання ефективності роботи мережі була обрана AUC (area under curve) метрика, що оцінює площу під ROC (receiver operating characteristic) кривою. Дана крива відображає відношення між кількість позитивно та негативно-позитивно класифікованих зображень при послабленні межі, що відповідає за позитивну відповідь. Ця метрика були обрана завдяки своїй здатності охарактеризувати якість роботи готової моделі незалежно від вибору межі позитивної класифікації, що неможливо сказати про метрику точності та F1 метрику.

**Результати тренування та їх аналіз**

Після тренування моделі були отримані наступні результати на тестовому наборі даних (табл. 2).

Згідно з таблицею результатів та схемою набору даних, нейронна мережа показала найгірші результати у більшості випадків в тих категоріях, де кількість даних найменша, тобто мережа не має достатньо інформації для правильної класифікації. Середнє значення AUC знаходиться на рівні 0.74, але при цьому спостерігаються значні відхилення від середнього значення ($\sigma$ = 0.071), що робить неможливим використання моделі у реальних умовах на даний момент, але варто зазначити, що завдяки подальшим покращенням моделі буде можливо покращити метрику, враховуючи значний простір для майбутніх досліджень.

*Табл. 2 Результати тренування*

| Категорія | AUC | Категорія | AUC |
|---|---|---|---|
| No Finding | 0.869 | Pneumonia | 0.731 |
| Enlarged Cardiom. | 0.616 | Atelectasis | 0.677 |
| Cardiomegaly | 0.799 | Pneumothorax | 0.728 |
| Lung Opacity | 0.712 | Pleural Effusion | 0.833 |
| Lung Lesion | 0.684 | Pleural Other | 0.714 |
| Edema | 0.829 | Fracture | 0.657 |
| Consolidation | 0.692 | Support Devices | 0.817 |
| **Average** | 0.74 | | |

Існує значна кількість способів покращення точності моделі, які будуть розглянуті у подальших дослідженнях та експериментах.

ЛІТЕРАТУРА

1. C. Qin, D. Yao, Y. Shi. Computer-aided detection in chest radiography based on artificial intelligence: A survey, Biomedical Engineering Online, 2018. doi:10.1186/s12938-018-0544-y (Last accessed: 08.03.2020).
2. L. Delrue, R. Gosselin, B. Ilsen. Difficulties in the interpretation of chest radiography, in: Comparative Interpretation of CT and Standard Radiography of the Chest, 2011. doi:10.1007/978-3-540-79942-9_2 (Last accessed: 08.03.2020).
3. J. Irvin, P. Rajpurkar, M. Ko. CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison, 2019. doi:10.1609/aaai.v33i01.3301590 (Last accessed 05.03.2020).
4. F. Pasa, V. Golkov, F. Pfeiffer. Efficient Deep Network Architectures for Fast Chest X-Ray Tuberculosis Screening and Visualization, 2019. doi:10.1038/s41598-019-42557-4 (Last accessed: 08.03.2020).
5. P. Lakhani, B. Sundaram, Deep learning at chest radiography: Automated classification of pulmonary tuberculosis by using convolutional neural networks, 2017. doi:10.1148/ radiol.2017162326 (Last accessed: 09.03.2020).
6. Y. LeCun, L. Bottou, Y. Bengio. Gradient-based learning applied to document recognition, 1998. doi:10.1109/5.726791 (Last accessed: 07.03.2020).
7. A. Krizhevsky, I. Sutskever, G. Hinton. ImageNet classification with deep convolutional neural networks. 2017. doi:10.1145/3065386 (Last accessed: 04.03.2020).
8. C. Szegedy, W. Liu. Going Deeper With Convolutions, 2014. URL: https://arxiv.org/abs/ 1409.4842 (Last accessed: 08.03.2020).
9. K. He, X. Zhang, S. Ren. Deep Residual Learning for Image Recognition, *2016.* doi:10.1109/CVPR.2016.90 (Last accessed: 07.03.2020).
10. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, 2015. URL: https://arxiv.org/abs/ 1502.01852 (Last accessed: 08.03.2020).
11. G. Huang, Z. Liu, K. Q. Weinberger. Densely Connected Convolutional Networks, 2017. doi: 10.1109/CVPR.2017.243 (Last accessed: 06.03.2020).
12. K. He, X. Zhang, S. Ren. Identity Mappings in Deep Residual Networks, 2016. doi: 10.1007/978-3-319-46493-0_38 (Last accessed: 05.03.2020).
13. T. Dozat. Incorporating Nesterov Momentum Into Adam, 2015. URL: http://cs229.stanford.edu/ proj2015/054_report.pdf (Last accessed: 08.03.2020).
14. N. Srivastava., G. Hinton, A. Krizhevsky. Dropout: a simple way to prevent neural networks from overfitting, 2014. doi:10.5555/2627435.2670313 (Last accessed: 06.03.2020).

**ГОЛУБНИЧИЙ Вадим Олександрович –** студент групи КІ-41 факультету комп'ютерних наук; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: illuminataldus@gmail.com; ORCID: 0000-0001-6482-2765.

Наукові інтереси:

- *штучний інтелект;*
- *методи машинного навчання;*.
- *комп'ютерний зір;*

**СТРІЛЕЦЬ Вікторія Євгенівна** – доцент кафедри теоретично та прикладної системотехніки, факультет комп'ютерних наук, Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: striletsvictoria@gmail.com; ORCID: 0000-0002-2475-1496.

Наукові інтереси:

- *методи машинного навчання;*
- *штучний інтелект;*
- *комп'ютерне та математичне моделювання.*

**УДК 004.056.55**

ГОРБЕНКО І.Д., КАЧКО О.Г., ЄСІНА М.В, ПОНОМАР В.А.

# СТАН ТА ПРОБЛЕМНІ ПИТАННЯ РОЗРОБЛЕННЯ ТА ВПРОВАДЖЕННЯ ПЕРСПЕКТИВНОГО СТАНДАРТУ ЦИФРОВОГО ПІДПИСУ

**Вступ**

Наразі, та, очевидно, в деякій перспективі, для надання послуг цілісності, справжності, доступності та неспростовності застосовуються методи цифрового підпису (ЦП), що орієнтовані на застосування існуючих, стандартизованих ЦП. Вони є суттєвою складовою забезпечення кібербезпеки та безумовними складовими децентралізованих технологій блокчейн (БЧ) тощо. Але є обґрунтовані підозри, що у постквантовий період існуючі стандарти ЦП, будуть зламуватись за допомогою квантових криптоаналітичних систем криптоаналітиком 3-го рівня. Важливою особливістю постквантового періоду є значна невизначеність щодо вихідних даних для криптоаналізу та протидії в частині можливостей квантових комп'ютерів (КВК), їх математичного та програмного забезпечень [1, 2]. Застосування існуючих ЦП, на наш погляд, необхідно розглядати враховуючи можливе існування перехідного та постквантових періодів. У перехідний період, можуть бути застосовані деякі існуючі нині ЦП, але з максимально можливими чи збільшеними довжинами та властивостями загальносистемних параметрів та ключових даних, в тому числі при використанні в системі БЧ. У постквантовий період існуючі ЦП з точки зору криптографічної стійкості скоріше всього не будуть застосовуватись, якщо суттєво будуть удосконалені КВК з необхідними для успішного криптоаналізу ЦП довжинами регістрів (в кубітах) та буде розроблене необхідне для їх реалізації математичне та програмне забезпечення.

Наразі вже практично створені та застосовуються квантові комп'ютери, стан створення та можливості застосування КВК наступні [1, 2]:

- IBM повідомила про план запуску в жовтні 2019 53-кубітного КВК, він має нову конструкцію процесора, масштабуємий, знижена ймовірність помилок, надійний в хмарі тощо;
- уже наразі 72-кубітний КВК Google за 3,5 хвилини виконує еквівалент роботи 10 тис. р. найпотужнішого кластера.

Метою доповіді є аналіз стану захищеності існуючих ЦП та обґрунтування необхідності, аналіз стану, створення, дослідження та прийняття постквантових стандартів ЦП, аналіз існуючої проблеми постквантової стандартизації та визначення шляхів її вирішення на міжнародному та національному рівнях як в теоретичному, так і практичному плані.

**Сутність та стан вирішення проблеми перспективних ЦП на світовому та національному рівнях**

NIST США організував та провів 1-й етап конкурсу щодо кандидатів на стандарти постквантових ЦП. Із 20 кандидатів 1-го етапу до 2-го етапу рекомендовано 9. Суттєві досягнення зі створення математичних та програмних моделей перетворень на квантових комп'ютерах за оцінками провідних світових фахівців в галузі кібербезпеки, призвели до істотного прогресу в області криптоаналізу сучасних стандартизованих ЦП, особливо у постквантовий період.

Основні вимоги до кандидатів на стандарти постквантових ЦП можна конкретизувати у трьох напрямках:

- вимоги з безпеки (вимоги до стійкості щодо криптографічного аналізу);
- техніко-економічні вимоги (в основному щодо часової та просторової складностей);
- техніко-експлуатаційні вимоги тощо.

Вимоги до стійкості ЦП сформульовані у відповідності з моделлю загроз, в умовах дії моделі EUF, тобто забезпечення захисту від екзистенційної підробки при атаках на основі адаптивно підібраного повідомлення. В ході початкових досліджень на 2-му етапі підтримано та продовжуються дослідження: три ЦП на основі алгебраїчних решіток; чотири ЦП на основі

багатовимірних перетворень; два ЦП на симетричній основі (блокові шифри та функції гешування).

**Аналіз стійкості існуючих кандидатів на постквантові стандарти ЦП**

До основних алгоритмів криптоаналізу, що можуть бути застосованими на квантовому комп'ютері необхідно віднести [1]: квантовий алгоритм факторизації Шора; квантовий алгоритм Гровера пошуку елемента в несортованій базі; квантовий алгоритм Шора для розв'язку дискретного логарифму в скінченому полі; квантовий алгоритм розв'язку дискретного логарифму в групі точок ЕС Шора; квантові алгоритми криптоаналізу для перетворень в фактор кільці; квантовий алгоритм криптоаналізу Ксіонга та Ванга та його вдосконалення тощо.

Криптографічна стійкість перспективних проектів повинна бути такою:

1) Стійкість проти класичних та квантових атак – класична та квантова безпека.

2) Базування на задачах, які мають високу складність обчислення. Можливе ігнорування зниження рівня складності, за умови, що практична стійкість не зміниться – доказова безпека.

3) Стійкість проти атак з адаптивним підбором – активна безпека.

4) Стійкість проти атак спеціального виду (сторонніми каналами).

Для прикладу результати аналізу складності дискретного логарифмування для класичного та квантового алгоритмів наведені в таблиці 1. Вони підтверджують загрози щодо застосування ЦП в групі точок еліптичних кривих.

*Табл. 1 Результати аналізу складності стандартів ЦП засобом дискретного логарифмування в групі точок*

| Алгоритм розв'язку дискретного логарифмічного рівняння | | | |
|---|---|---|---|
| Порядок базової точки | Необхідні кубіти $f(n)=7n+4\log_2 n+10$ | Складність квантового алгоритму $360n^3$ | Складність класичного алгоритму |
| 256 | 1834 | $6\cdot10^9$ | $3{,}4\cdot10^{38}$ |
| 571 | 4016 | $6{,}7\cdot10^{10}$ | $8{,}8\cdot10^{85}$ |
| 1024 | 7218 | $3{,}8\cdot10^{11}$ | $1{,}3\cdot10^{154}$ |

До 2-го раунду пройшло 9 схем ЦП: CRYSTALS-DILITHIUM, FALCON, GeMSS, LUOV, MQDSS, Picnic, qTESLA, Rainbow та SPHINCS+. Три з них (Dilithium, FALCON, qTESLA) засновані на стійкості алгебраїчних решіток, чотири (GeMSS, LUOV, MQDSS, Rainbow) – на основі MQ-перетворень, одна (SPHINCS+) – на стійкості геш-функції, одна (Picnic) – на стійкості геш-функції та симетричного блокового шифру. Предметом наших досліджень є ЦП, що створюються на застосуванні алгебраїчних решіток. Результати їх досліджень дозволяють визначити їх властивості таким чином [1, 2].

1. Механізм на основі CRYSTALS – Dilithium [1] (далі Dilithium). Механізм Dilithium є схемою підпису на основі решітки, побудованої з використанням евристики Фіат-Шаміра, його безпека заснована на складності задачі MLWE. Dilithium є частиною пакету CRYSTALS разом з механізмом обміну ключами Kyber. Основною новизною Dilithium є те, що розмір відкритого ключа зменшений за рахунок не включення деяких бітів нижнього порядку, щоб компенсувати це, кожен підпис включає додаткову «підказку», що дозволяє верифікатору перевіряти підпис. Dilithium забезпечує досить хороші показники і є досить простим у реалізації.

Найбільш відомі атаки проти Dilithium засновані на зведенні базису решітки, без значного використання алгебраїчної структури задачі MLWE. Параметри вибору Dilithium базуються на консервативних оцінках витрат цих атак. Dilithium має формальне підтвердження стійкості в класичній моделі випадкового оракула. Цей доказ є нетривіальним, і він не діє у квантовій моделі випадковій оракула; проте ніяких атак поки не відомо.

2. Механізм на основі ЦП Falcon [1]. Механізм Falcon представляє схему підпису на решітках, яка заснована на GPV (Gentry-Peikert-Vaikuntanathan) вибірці Гауса, яка використовує решітку NTRU. Основною її новизною є дуже швидкий рекурсивний алгоритм для вибірки Гауса, використовуючи структуру даних дерево («Falcon tree»). Стосовно Falcon найбільш відомі атаки засновані на зведенні базису решітки, без істотного використання спеціальної структури решітки NTRU. Автори Falcon дали формальний доказ безпеки у квантовій моделі випадкового оракулу [1]. Аналіз показав, що Falcon відповідає практично усім вимогам. Проте реалізувати його досить складно, оскільки він значною мірою спирається на структуру числових полів і розкладання

полів та вимагає арифметики подвійної точності з плаваючою точкою. Також потрібно провести подальші дослідження щоби пересвідчитись, що алгоритм ЦП Falcon є захищеним від атак сторонніми каналами.

3. Механізм на основі qTESLA [1]. Механізм qTESLA є ЦП, розроблений на основі решітки, яка використовує припущення, що розподіли RLWE не відрізняються від випадкових. Відкритий ключ механізму qTESLA, грубо кажучи, є варіантом моделі RLWE. В ньому підписувач зберігає секретну інформацію про нього і використовує цю інформацію разом з геш-функцією для вироблення ЦП. Перевірка підпису передбачає використання деякої простої арифметики в межах вибраного кільця, а потім повторне обчислення геш-функції. Механізм qTESLA має досить хороші параметри продуктивності, які порівнюються з іншими схемами ЦП на основі решітки. Розробники qTESLA заявили про повний доказ стійкості на схемі у квантовій моделі випадкового оракула. Також відзначено, що помилка у доказі стійкості вимагає коригування параметрів (що знижує його ефективність).

У таблиці 2 наведено перелік кандидатів на ЕП, що визначені на семінарі 2-го етапу конкурсу.

*Табл. 2. Постквантові кандидати на ЕП за 2-й етап*

| Математичні методи ЦП | Кандидати в стандарти ЦП | Число |
|---|---|---|
| Алгебраїчні решітки | Crystals-Dilithium, Falcon, qTesla | 3 |
| Геш-перетворення | Sphincs+ | 1 |
| Кінцеві автомати | Picnic | 1 |
| Багатовимірні перетворення | Gemss, Luov, Mqdss, Rainbow | 4 |
| Усього | | 9 |

Згідно прийнятої методики [3] було проведено оцінки та порівняльний аналіз кандидатів на постквантовий стандарт ЦП. Відносні переваги вказаних ЦП наведені в таблиці 3. Графічне подання переваг наведено на рис 1.

*Табл. 3. Відносна перевага алгоритмів ЦП*

| Алгоритми | $I_{ст.}$ | $l_{в.к.}$ | $l_{о.к.}$ | $l_{рез.}$ | $T_{пр.}$ | $T_{зв.}$ | $T_{гк.}$ |
|---|---|---|---|---|---|---|---|
| Dilithium_very_high | 0,0248 | 0,0723 | 0,0328 | 0,0783 | 0,3727 | 0,3309 | 0,2924 |
| falcon1024 | 0,1326 | 0,0723 | 0,0204 | 0,1396 | 0,1577 | 0,2058 | 0,0266 |
| qTesla_256 | 0,1326 | 0,0375 | 0,0204 | 0,0509 | 0,2503 | 0,1764 | 0,1053 |
| sphincs | 0,1326 | 0,2767 | 0,0943 | 0,0217 | 0,0129 | 0,0604 | 0,0142 |

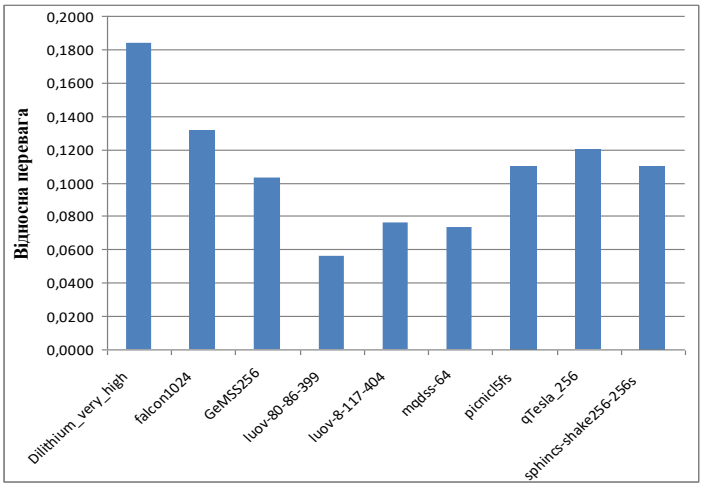



*Рис. 1. Переваги алгоритмів ЦП*

**Висновки**

1. На конкурс NIST США щодо стандарту ЕП було подано механізми ЦП, що ґрунтуються на алгебраїчних решітках – Crystals-Dilithium, Falcon та qTESLA. Наразі вони досліджуються на другому етапі конкурсу NIST США.

2. Наші дослідження можливостей створення постквантового ЦП, що може бути стійким у постквантовий період, підтвердили, що надійною їх математичною основою є алгебраїчні решітки.

3 Результати порівняльного аналізу механізмів Crystals-Dilithium, Falcon та qTESLA показали, що механізм Crystals-Dilithium згідно безумовних, умовних та прагматичних критеріїв має суттєві переваги (рис. 1).

4. При подальших дослідженнях необхідно провести детальні дослідження стосовно можливостей реалізації механізмів ЦП з 5-7 рівнями стійкості та доведення стійкості стосовно атак сторонніми каналами.

5. Найбільш складними задачами стосовно реалізації 5-7 рівнів стійкості є побудування загальних параметрів та ключів для механізму Crystals-Dilithium.

ЛІТЕРАТУРА

1. Post-Quantum Cryptography. – URL: https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

2. Post-Quantum Cryptography. Workshops and Timeline. URL: https://csrc.nist.gov/Projects/post-quantum-cryptography/workshops-and-timeline.

3. Yesina Maryna, Olga Akolzina, Olena Kachko (supervisor). Proposals of the expert estimations technique usage for the comparing and estimation NTRU-like cryptographic systems // Inżynier XXI wieku (“Engineer of XXI Century” – the VII Inter University Conference of Students, PhD Students and Young Scientists: University of Bielsko-Biala, Poland, December 08, 2017). – Bielsko-Biała: Wydawnictwo Naukowe Akademii Techniczno-Humanistycznej w Bielsku-Białej, 2017. – P. 383–398. – ISBN 978-83-65182-81-4 (Tom 2) – Chapter in monograph.

**ГОРБЕНКО Іван Дмитрович** – д.т.н, професор, професор кафедри безпеки інформаційних систем і технологій, факультет комп’ютерних наук, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків, Харківська область, 61000; e-mail: gorbenkoi@iit.kharkov.ua.

Наукові інтереси:

*– криптографія, криптоаналіз, постквантова криптографія, захист інформації.*

**КАЧКО Олена Григорівна** – к.т.н., професор кафедри, ПІ ХНУРЕ, проспект Науки, 14, Харків, Харківська область, 61000; e-mail: ekachko@gmail.com; ORCID: 0000-0001-9249-0497.

Наукові інтереси:

*– криптографія, криптоаналіз, паралельні обчислення.*

**ЄСІНА Марина Віталіївна** – к.т.н., старший викладач кафедри безпеки інформаційних систем і технологій, факультет комп’ютерних наук, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків, Харківська область, 61000; e-mail: m.v.yesina@karazin.ua; ORCID: 0000-0002-1252-7606.

Наукові інтереси:

*– захист інформації, постквантова криптографія.*

**ПОНОМАР Володимир Андрійович** – к.т.н., науковий співробітник кафедри безпеки інформаційних систем і технологій, факультет комп’ютерних наук, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків, Харківська область, 61000; e-mail: Laedaa@gmail.com.

Наукові інтереси:

*– криптографічні перетворення, безпечне програмування, захист криптографічних засобів інформації.*

**УДК 533.9**

ГРАДИСЬКИЙ О.Ю., КАРАСЬ І.В.

# КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ НАГРІВУ ПЛАЗМИ МІКРОХВИЛЬОВИМ ВИПРОМІНЮВАННЯМ ЗІ СТОХАСТИЧНИМИ СТРИБКАМИ ФАЗИ

У стохастичних електромагнітних полях може відбуватися багато корисних процесів, таких як нагрів плазми або прискорення заряджених частинок. При цьому відбувається обмін енергією між надвисокочастотними стохастичними електромагнітними полями та зарядженими частинками. Частотою зіткнень при цьому виступають випадкові стрибки фази стохастичних коливань, а набрана енергія пропорційна частоті стрибків фази.

**Набір енергії частинкою в полі хвилі зі стохастичними стрибками фази та пружними зіткненнями з молекулами газу.**

Основною темою проведених досліджень є розгляд руху електрона в полі поперечної електромагнітної (ТЕМ) хвилі, яка поширюється в коаксіальному плазмовому хвилеводі.

В такому випадку відоме рівняння руху електрона:

$$m\frac{d\vec{\upsilon}}{dt} = e\vec{E} + \frac{e}{c}\left[\vec{\upsilon}\vec{H}\right] - \nu_m\vec{\upsilon}\,, \tag{1}$$

де $\upsilon$ – швидкість електрона,

t – час,

e – заряд електрона,

m –маса електрона,

$\nu_m$ – частота пружних зіткнень,

E та H – напруженості електричного та магнітного поля.

Якщо зовнішнє магнітне поле відсутнє, в поперечно-електромагнітній хвилі будуть присутні лише компоненти полів $E_r$ и $H_\varphi$ [1].

$$m\frac{d\upsilon_z}{dt} = \frac{e}{c}\upsilon_r H_\varphi - \nu_m\upsilon_z\,, \tag{2}$$

$$m\frac{d\upsilon_r}{dt} = eE_r + \frac{e}{c}\upsilon_z H_\varphi - \nu_m\upsilon_r\,, \tag{3}$$

$$m\frac{d\upsilon_\varphi}{dt} = 0 \tag{4}$$

З наведених рівнянь можна бачити, що рух електрона відбуватиметься в площині r0z.

В узагальненому випадку потрібно розглядати рух електрона у схрещених неоднорідних полях:

$$E_r = E_0\cos(\omega t - k_3 z + \varphi(t))\,, \tag{5}$$

$$H_\varphi = H_0\cos(\omega t - k_3 z + \varphi(t)) \tag{6}$$

Оцінки кінетичної енергії електрона демонструють, що швидкість $\upsilon$ електрона складає близько $2\cdot10^8$ см/с. В експериментах по вивченню розряду ініційованого мікрохвильового випромінювання зі стохастичними стрибками фази частота мікрохвильового випромінювання складала 500 МГц, амплітудне значення напруженості електричного поля складало не більше 100 В/см [2]. І в такому випадку амплітуда зсуву електрона в напрямку осі z за час, що дорівнює періоду хвилі складає близько $\Delta z$=0,4 см, з чого можна зробити висновок, що значення зсуву $\Delta z$ набагато менші, ніж довжина хвилі мікрохвильового випромінювання $\lambda$=60 см. Згідно вище зазначеного, можна вважати поля однорідними та не враховувати величину $k_3\Delta z$ порівняно з величиною стрибка $\Delta\varphi$. Другий же доданок в рівнянні руху електрона менший за перший на два порядки через дуже мале відношення швидкості до швидкості світла. Тому можна розглядати рух електрона в полі виду:

$$E = E_0 \cos(\omega t + \varphi(t)), \tag{7}$$

де $E_0$ – амплітуда напруженості електричного поля хвилі,

ω – частота хвилі,

φ(t) – значення фази, що змінюється у випадковий момент часу t на випадкову величину від –π до π.

Почергово розглянемо рух електрона з різними умовами.

**Набір енергії електроном при наявності пружних зіткнень.**

У цьому випадку фаза електрона у випадкові моменти часу змінюється на випадкову величину. Рівняння руху частинки тоді виглядатиме наступним чином [3, 4]:

$$\frac{d\upsilon}{dt} = \frac{eE}{m}\cos(\omega t + \varphi(t)) - \nu_m \upsilon \tag{8}$$

Можна перейти до безрозмірних величин:

$$V = \frac{\upsilon}{\upsilon_0}, \tag{9}$$

де $\upsilon_0 = \frac{eE}{m\omega}$ – осциляторна швидкість,

$\tau = \frac{t}{T}$ – час рахується у періодах високочастотної хвилі,

ν – частота пружних зіткнень, нормована на зворотній період хвилі.

У безрозмірних величинах рівняння матиме наступний вигляд:

$$\frac{dV}{d\tau} + \nu V = 2\pi \cdot \cos(2\pi\tau + \varphi(t)) \tag{10}$$

При відсутності стрибків фази (де φ(τ)=const, хвиля регулярна), рівняння можна розв'язати аналітично:

$$V(\tau) = \frac{4\pi^2}{4\pi^4 + \nu^2}\left[\sin(2\pi\tau) + \frac{\nu}{2\pi}\cos(2\pi\tau)\right] - \frac{2\pi\nu}{4\pi^2 + \nu^2}e^{-\nu\tau} \tag{11}$$

При відсутності зіткнень (при $\nu = 0$) з цього рівняння видно наступне:

$$V(\tau) = \sin(2\pi\tau) \tag{12}$$

Тоді зміна енергії буде такою:

$$\frac{d\hat{\varepsilon}}{d\tau} = 4\pi V(\tau)\cdot\cos(2\pi\tau) = 4\pi\cdot\cos(2\pi\tau)\cdot\sin(2\pi\tau), \tag{13}$$

де $\hat{\varepsilon}$ – енергія електрона, нормована на величину $\varepsilon_0 = m\upsilon^2/2$.

Після інтегрування отримуємо:

$$\Delta\hat{\varepsilon} = \frac{1}{2}\cdot\cos(4\pi\tau) \tag{14}$$

Це означає, що при відсутності зіткнень змін енергії в середньому за період немає.

**Набір енергії електроном в полі хвилі зі стрибками фази.**

Якщо зіткнення відсутні, але присутні стрибки фази в хвилі, рівняння виглядає наступним чином:

$$\frac{dV}{d\tau} = 2\pi\cdot\cos(2\pi\tau + \varphi(t)) \tag{15}$$

Переходячи до безрозмірних величин в рівнянні для енергії маємо:

$$\frac{d\hat{\varepsilon}}{d\tau} = 4\pi V(\tau)\cdot\cos(2\pi\tau + \varphi(\tau)) \tag{16}$$

**Набір енергії електроном в полі хвилі зі стрибками фази та пружними зіткненнями з молекулами газу.**

У цьому випадку до набору енергії електроном ще включаються пружні зіткнення. Це враховується при розв'язанні системи рівнянь:

$$\frac{dV}{d\tau} = 2\pi \cdot \cos(2\pi\tau + \varphi(t)) - \nu V\,, \tag{17}$$

$$\frac{d\hat{\varepsilon}}{d\tau} = 4\pi V(\tau) \cdot \cos(2\pi\tau + \varphi(\tau)) - \nu \frac{m}{M}\hat{\varepsilon}\,, \tag{18}$$

де ν – частота пружних зіткнень, нормована на зворотній період хвилі,
M – маса молекули.

При чисельних розрахунках зміни енергії, доданок $\nu \frac{m}{M}\hat{\varepsilon}$, що пов'язані з втратою енергії в результаті пружних зіткнень не враховуються.

Залежність частоти пружних зіткнень $\nu_m$ від енергії розраховується за формулою:

$$\nu_m = N \cdot \sigma(\varepsilon) \cdot \upsilon\,, \tag{19}$$

де N – концентрація газу,
σ – переріз розсіювання для пружних зіткнень,
υ – швидкість електрона.

Газ, що використовується для дослідів – повітря.

При розрахунку залежності пружних зіткнень від енергії для повітря враховувалося, що відсоткове співвідношення азоту та кисню в повітрі складають 80% та 20% відповідно.

При вказаному відсотковому співвідношенні азоту та кисню, переріз для азоту є визначальним. Тому розрахунок зміни енергії електрона в полі хвилі зі стрибками фази при наявності пружних зіткнень проводиться до відносної енергії $\varepsilon_{max}$, що відповідає максимуму перерізу іонізації азоту.

Результати числових розрахунків наведені у відповідності до експериментальних залежностей по вимірюванню електричного поля пробою у діапазоні від 20 до 160 В/см [8, 9].

Частота стрибків фази в експерименті в оптимальному режимі роботи пучково-плазмового генератора лишалася постійною і складала один стрибок на періоді високочастотної хвилі. Потужність регулювалася від 1 кВт до 28 кВт завдяки використанню широкосмугового відгалужувача [10-15]. При проведенні числових розрахунків використовувалося таке ж значення частоти стрибків фази.

Завдяки значенню поля E можна знайти нормувальну константу $\upsilon_0$,та відповідно $\varepsilon_0$. Значення тиску P для кожної фіксованої точки дають значення концентрації N з формули (19), що дозволяють пов'язати частоту пружних зіткнень з енергією.

Таким чином, в програмі можна буде врахувати залежність середнього часу між зіткненнями від набраної електроном енергії.

**Зміни енергії електрона в полі хвилі зі стрибками фази, при відсутності зіткнень.**

Розглянемо результати роботи програми при розрахунку зміни енергії електрона в полі хвилі зі стрибками фази та при відсутності зіткнень. Час вимірюється в періодах високо-частотного поля. На графіку представлені результати розрахунків усереднені по 50 реалізаціям.


*Рис. 1 Залежність безрозмірної енергії електрона E від безрозмірного часу за відсутності зіткнень.*

З рис. 1 можна побачити, що при відсутності пружних зіткнень, але при наявності стрибків фази високо-частотного поля енергія електрона в середньому зростає, на відміну від

випадку регулярної хвилі (за відсутності стрибків фази), коли набір енергії електроном можливий лише при наявності зіткнень.

**Зміна енергії електрона в полі хвилі зі стрибками фази та при наявності пружних зіткнень.**

На рис. 2 зображена залежність середнього часу між зіткненнями, що відрахований в періодах електромагнітної хвилі, від безрозмірної енергії електрона для точок 1, 1', 1'' з рис. 5. Точка 1 відповідає значенню тиску P=5 Па, 1'– P=13 Па, 1''– P=33 Па.

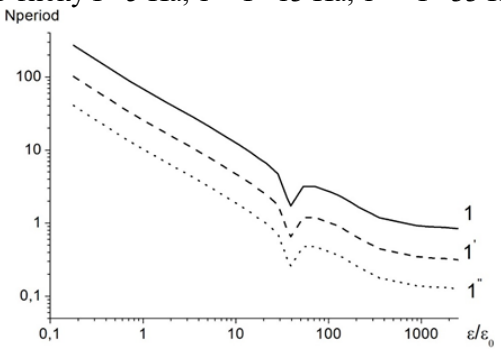



*Рис. 2 Залежність середнього часу між зіткненнями, що відрахований в періодах електромагнітної хвилі, від безрозмірної енергії електрона, для значень поля E=25 В/см, тиску P=5 Па для точки 1, P=13 Па – 1', P=33 Па – 1''ь.*

З оцінки перерізу пружних зіткнень (рис 2) можна побачити, що при значеннях ε, що наближаються до $\varepsilon_{max}$ для P=5 Па, зіткнення відбуваються в середньому один раз в період високочастотної хвилі, так само як і стрибки фази. Це означає, що роль зіткнень у наборі енергії електроном порівнювана з роллю стрибків фази. При великому тиску зіткнення ж відбуватимуться частіше, ніж стрибки фази: При P=13 Па приблизно три рази за період, при P=33 Па – 6 разів за період. Тобто стрибки фази відіграють важливу роль лише при низьких тисках. При енергіях електрона порядку осциляторної (ε = 1) пружні зіткнення відбуваються з частотою приблизно один раз у 100 періодів.

Використовуючи значення залежності середнього часу між зіткненнями від набраної електроном енергії з рис. 2, розглянемо результати роботи програми при розрахунку зміни енергії електрона в полі хвилі зі стрибками фази та при наявності пружних зіткнень, для значень поля E=25 В/см, тиску P=5 Па. Час вимірюється в періодах високочастотного поля.

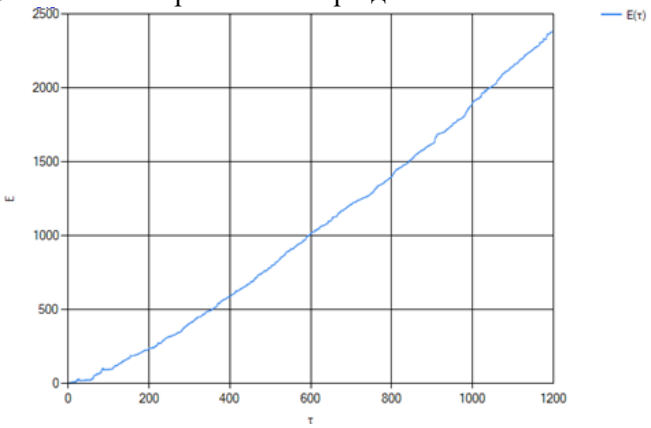



*Рис. 3 Усереднений розрахунок набору енергії електроном для поля E=25 В/см, тиску P=5 Па.*

На рис 3 зображені результати розрахунків енергії, що була набрана електроном, результати розрахунків усереднені по 50 реалізаціям. При значенні поля E=25 В/см осциляторна швидкість електрона $\upsilon_0=1{,}4\cdot10^7$ см/с, та відповідно нормувальне значення енергії $\varepsilon_0=5{,}6\cdot10^{-2}$ еВ. Енергія, яку необхідно набрати електрону дорівнює максимуму перерізу іонізації азоту і складає приблизно 120 еВ. Тож, розрахунок проводиться до відносної енергії, що дорівнює $\varepsilon_{max}$ = 2150. Характерний час, за який електрон набирає таку енергію дорівнює $t_{ion}$=2,2 мкс, що значно менше, ніж тривалість високочастотного імпульсу.

ЛИТЕРАТУРА

1. Karas` I.V. Electromagnetic modes of a coaxial plasma waveguide in an external magnetic field / I.V. Karas`, I.A. Zagrebelny // Problems of Atomic Science and Technology. Series: Plasma Electronics and New Acceleration Methods. – 2015. – № 4 (98). – P. 36-42..
2. Пучково-плазменный генератор стохастических колебаний дециметрового диапазона / А.К. Березин, Я.Б. Файнберг, А.М. Артамошкин [и др.] // Физика плазмы. – 1994. – Т.20, №9. – С. 782-789.
3. Карась В.И. Набор энергии электронами в поле волны со стохастическими скачками фазы при наличии упругих и неупругих столкновений / В.И. Карась, И.А. Загребельный // Инженерная физика. – 2015. – №11. – С. 46-52.
4. Ландау Л.Д. Теория поля / Л.Д. Ландау, Е.М. Лифшиц – М.: Наука, 1973. – 504 с.
5. Cross Sections for Collisions of Electrons and Photons with Nitrogen Molecules / Y. Itikawa, M. Hayashi, A. Ichimura [et. al.] // Journal of Physical and Chemical Reference Data. – 1986. – V.15, № 3. – P. 985-1010.
6. Cross Sections for Collisions of Electrons and Photons with Oxygen Molecules / Y. Itikawa, A. Ichimura, K. Onda [et. al.] // Journal of Physical and Chemical Reference Data. – 1989. – V.18, №1. – P. 23-42..
7. Физические величины: Справочник / А.П. Бабичев, Н.А. Бабушкина, А.М. Братковский [и др.]. Под ред. И.С. Григорьева, Е.З. Мейлихова. – М.: Энергоатомиздат, 1991. – 1232 с.
8. Experimental investigations of propagation of microwave radiation with stochastic jumping phase in overdense plasma / A.F. Alisov, A.M. Artamoshkin, I.A. Zagrebelny [et. al.] // Problems of Atomic Science and Technology. Series: Plasma Electronics and New Acceleration Methods. – 2003. – №4(3). – P. 69-73.
9. Пробой и разряд в газе низкого давления, создаваемый микроволновым излучением со стохастически прыгающей фазой (I) / В.И. Карась, А.Ф. Алисов, А.М. Артамошкин [и др.] // Вопросы атомной науки и техники. Серия: Плазменная электроника. – 2006. – № 5(5). – С. 54-58.
10. Взаимодействие микроволнового излучения со стохастически прыгающей фазой с плазмой или газом / В.И. Карась, Я.Б. Файнберг, А.Ф. Алисов [и др.] // Физика плазмы. – 2005. – Т. 31, №9. – С. 810-822.
11. Зависимость пороговой мощности пробоя от давления газа в различных режимах работы генератора СВЧ-излучения со скачками фазы / А.Ф. Алисов, А.М. Артамошкин, В.И. Голота [и др.] // Светотехника и электроэнергетика. – 2009. – №3(19). – С. 4-8.
12. Special Features of Low–Pressure Discharge Initiated by Microwave Radiation With Stochastic Jumping Phase / V.I. Karas`, A.M. Artamoshkin, A.F. Alisov [et. al] // IEEE Transaction on Plasma Science. – 2013. – V.41, № 9. – P. 2458-2463.
13. Разряд низкого давления, индуцированный микроволновым излучением со стохастически прыгающей фазой / А.Ф. Алисов, А.М. Артамошкин, О.В. Болотов [и др.] // Доповіді НАН України. – 2010. – № 8. – С. 74-82.
14. Low pressure discharge initiated by microwave radiation with stochastically jumping phase / V.I. Karas`, A.M. Artamoshkin, A.F. Alisov [et. al.] // Problems of Atomic Science and Technology. Series: Plasma Physics. – 2012. – № 6 (82). – P. 142-145.
15. Optical radiation special features from plasma of low pressure discharge initiated by microwave radiation with stochastic jumping phase / A.F. Alisov, O.V. Bolotov, V.I. Golota [et. al.] // Problems of Atomic Science and Technology. Series: Plasma Electronics and New Acceleration Methods. – 2013. – № 4 (86). – P. 183-188.

**ГРАДИСЬКИЙ Олександр Юрійович** – студент 4 курсу, факультет комп'ютерних наук, Харківський національний університет імені В.Н. Каразіна.

**КАРАСЬ Ірина В'ячеславівна** – кандидат фіз.-мат. наук, доцент; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: irakaras22@ukr.net; ORCID: 0000-0001-8872-8204.

**УДК 004.056.55**

ГУРЬЕВА Е.А., ПОПОВА М.В., ЕСИНА М.В.

# ПРОТОКОЛ КОНСЕНСУСА POW И ЕГО УЯЗВИМОСТИ

**Основные принципы функционирования протокола**

Консенсус – это соглашение, которое удовлетворяет каждую из вовлеченных сторон. В контексте криптографии консенсус является процедурой принятия решения. Его цель – обеспечить всех участников сети возможностью согласования своего текущего состояния после добавления новой информации, блока данных или пакета транзакций. Иными словами, протокол консенсуса гарантирует, что сформированная цепь верна и подтверждает честность (легитимность) ее участников. Это важная структура для предотвращения ситуации, когда кто-то один контролирует всю систему, и она гарантирует то, что все участники соблюдают правила сети.

Протокол – это набор правил. Протоколы помогают:

- обеспечить стабильные условия для осуществления транзакций в сети;
- устранить возможность двойной траты;
- удостовериться, что все участники соблюдают предусмотренные правила.

Роль консенсусных алгоритмов заключается в обеспечении требуемого уровня надежности сети, построенной на серии узлов (устройств, соединённых с другими устройствами, как часть компьютерной сети). Консенсусные алгоритмы должны быть достаточно развитыми, чтобы успешно предсказывать любые возможные сбои коммуникации внутри сети. Алгоритм автоматически спрогнозирует, что некоторые процессы и системы будут недоступны, и что в результате этого некоторые коммуникации будут потеряны. Чтобы противостоять этому, консенсусный алгоритм должен быть отказоустойчивым и работать для достижения заранее определенного консенсуса или одобрения, по крайней мере, от большинства узлов. Блокчейн-системы, могут обладать только двумя из трёх возможных свойств: децентрализация, масштабируемость, безопасность. Каждый согласованный алгоритм имеет свой собственный сценарий применения, а выбор того, какой конкретно консенсус использовать для реализации блокчейна, зависит от типа сети и данных. Чтобы транзакция была действительной в большинстве криптовалютных сетей, эта транзакция должна собрать определенное количество подтверждений (часто равных включению в блок цепочки блоков) из сети. Например, процесс получения 10 подтверждений означает просмотр конкретной транзакции в одном и 9 последовательных блоках.

Механизм достижения консенсуса PoW стал популярным благодаря криптовалютам. Он является самым простым по устройству и при этом самым надежным в условиях полной децентрализации и анонимности. заключается в доказательстве о выполненной работе. Он разработан для ограничения нападений с целью отказа в обслуживании [1].

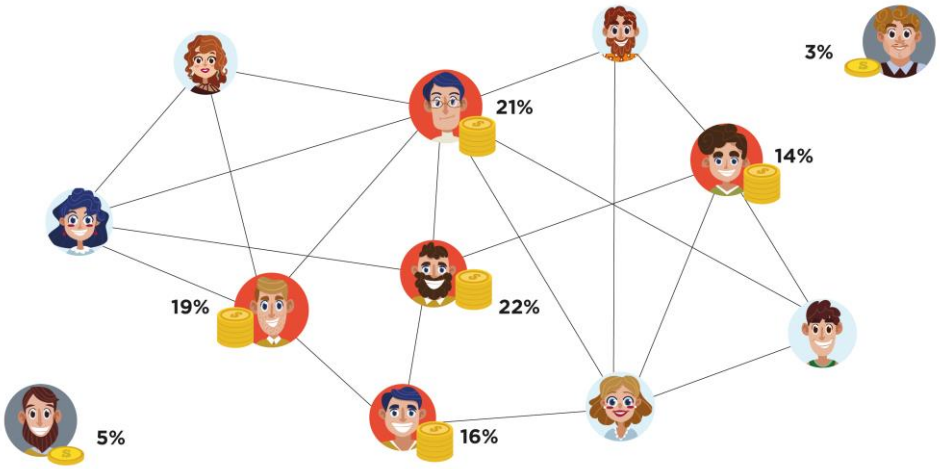



*Рис. 1 – Распределение мощностей на примере цифровых монет*

Сначала добыча Bitcoin опиралась на вычислительную мощность стандартных компьютеров, поэтому каждый мог стать валидатором. С 2014 года в добыче доминируют специально разработанные компьютерные микросхемы, известные как интегральные микросхемы (ASIC) [2]. Валидаторы все больше объединялись в коалиционные пулы, чтобы оценить риски и максимизировать прибыль. Как следствие, добывающая мощность постоянно становится более централизованной в «пулах добычи». Это породило направление исследований, используя методы теории игр и конструкции механизмов, чтобы препятствовать формированию централизованных пулов и уменьшению их влияния на систему [3].

Требования к задаче на формирование PoW:

- Возможность самостоятельно выбрать входные данные для решения задачи
- Проверка решения должна быть очень быстрой
- Проверить решение может любой желающий
- Хэш-значение предыдущего блока заранее не известно
- Входные данные к задаче для следующего блока отличаются для всех валидаторов
- Сложность задачи одинакова для всех участников
- Затраты ресурсов для нахождения решения определяются параметром сложности

PoW показали, что они могут масштабироваться до большого количества пользователей, однако скорость транзакций может не подходить для некоторых случаев использования [4]. Например, оригинальная версия Bitcoin способна обрабатывать около 7 транзакций в секунду [5], 1 блок каждые 10 минут и может потребовать в среднем до 1 часа для достижения подтверждения. Следует отметить, что на практике время подтверждения может меняться, поскольку генерация блока не является детерминированной. На самом деле время подтверждения зависит от объема сетевой активности и стоимости операций. Узлы должны хранить несколько цепочек, когда они появляются, однако, чем старше блок в цепочке, тем маловероятно, что он будет изменен. Типичное количество подтверждений, принятых большинством поставщиков Bitcoin сообщества и кошельков, составляет 6 подтверждений, то есть примерно 1 минута для получения блока [6]. Ранние блокчейны, разработанные в платформе Ethereum, которые используют PoW, могут обрабатывать до 20 транзакций в секунду [7]. Visa, с другой стороны, считается способной поддерживать 24000 транзакций в секунду (текущее среднее значение 1700 транзакций в секунду). Разработчики системы блокчейн постоянно работают над повышением скорости и масштабируемости. Исследуются решения этих вопросов, такие как увеличение размера блока [2], а также использование шардинга, боковых цепей и платежных каналов, которые обещают мгновенное завершение. Такие решения имеют потенциал для значительного улучшения времени, однако, это увеличивает влияние валидаторов, что может привести к нежелательной централизации [8]. Основной критикой является то, что PoW отвечает за расходование больших объемов реальных ресурсов, таких как электроэнергия. Например, страница Wiki Ethereum утверждает, что Bitcoin и Ethereum сжигают более $ 1 млн. на электроэнергию аппаратными средствами в день для выполнения своего консенсусного механизма [9]. Pilkington [4] провел медиа-релиз под названием калькулятор Bitcurrency, который показывает, что Bitcoin может потреблять за раз до 60% мирового производства электроэнергии, что эквивалентно 13000 TWh, что питает 1,5 миллиарда домов. Другие источники сообщают, что Bitcoin может потреблять столько электроэнергии, как Дания [10] до 2022 года, с проверкой одной транзакции Bitcoin, которая потребляет 200 кВт/ч электроэнергии [11]. Эта стоимость может не быть оправдана для операций с низким уровнем риска, где пользователям можно доверять, или существуют установленные методы для предотвращения вредоносного поведения [8].

Валидаторы конкурируют друг с другом, чтобы добавить новый блок в существующей блокчейн, решая криптографическую задачу генерирования хэш-выхода, который начинается с нескольких последовательных нулей в наиболее значимых позициях. Используемый способ добавляет к блоку случайное число, которое может использоваться только один раз, и вычисляет хэш-выход заголовка блока.

Заголовок блока содержит такую информацию, как хэш предыдущего проверенного блока и специальный хэш всех транзакций, содержащиеся в блоке (дерево Merkle). Целью

является поиск хэш-выхода, который является правильным. Валидаторы не имеют возможности прогнозировать или влиять на результат, поэтому единственное возможное действие – попытка. Эта процедура требует вычислительных усилий, которые увеличиваются экспоненциально с числом конечных нулей. Когда найдено правильный хэш-выход, блок возвращается в сеть Bitcoin и принимается другими узлами, если все транзакции действительны и неизрасходованные, а успешный валидатор принимает финансовое вознаграждение.

Другие валидаторы принимают вновь блок, начиная работу над последующим блоком. Важно, что все последующие блоки содержат хэш-выходы из всех предыдущих блоков. Поскольку генерирования хэш-выходов случайно и выполняется параллельно многими валидаторами, могут появиться многочисленные цепочки. В этом случае сеть сохраняет все полученные цепи. Участники сети в конце концов отказываются от меньших по длине, принимая за правильную ту цепь, которая, как предполагают, была произведена большинством вычислительных мощностей и представляют наиболее вероятное состояние книги. Как следствие, злоумышленники могут постоянно опережать честную часть сети, если они не могут контролировать более 51% вычислительной мощности в сети. В случае атаки на 51%, злонамеренные узлы могут переписать всю историю транзакций. Нарушения в безопасности могут быть введены пользователями, валидаторами, хакерами или атаками «человек-по-середине».

**Уязвимости протокола достижения консенсуса**

Безопасность криптовалют является одним из основных преимуществ технологии блокчейн, но нет ничего совершенного. У каждого алгоритма есть свои уязвимости, которые находят взломщики. Принцип безопасности основан на том, что информация о транзакциях подтверждается другими участниками сети, незнакомыми друг с другом, что и используют атакующие, перехватывая информацию о транзакциях.

Блокчейн, использующий алгоритм proof-of-work, может быть подвержен атакам, основанным на атаке «двойного расходования» – повторной трате средств. Данная атака может быть успешной, если в сети блокчейна используется малое число подтверждений блоков или, если у злоумышленника находится большее число вычислительной мощности, чем у всех остальных майнеров. В качестве общей защиты от атаки «двойного расходования» предлагается использовать такие методы, как мониторинг сети блокчейна, использование «черных списков» и ожидание множественных подтверждений транзакций для подтверждения совершения операций [12, с.10].

Атака Финни (Finney Attack) является вариацией атаки «двойного расходования», когда для совершения сделки ожидается не более одного подтверждения транзакции. Атакующий готовит транзакцию с оплатой товара и вместе с ней готовит блок, содержащий транзакцию на перевод этих средств на другой свой счет, но не публикует этот блок в сети. Как только транзакция с оплатой подтверждается одним из майнеров и злоумышленник получает товар, он незамедлительно публикует заранее подготовленный блок в сеть. В этом случае в сети оказывается две цепочки блоков одинаковой длины. И если остальные майнеры будут развивать вторую цепочку, содержащую транзакцию на перевод денег на счет атакующего, то транзакция перевода денег продавцу будет отклонена, и, следовательно, продавец потеряет деньги, так как товар уже был отправлен. Защитой в данном случае является ожидание продавцом некоторого достаточного числа подтверждений транзакций, что уменьшает вероятность данной атаки, но не устраняет ее полностью [2, с. 11].

Если атакующий имеет контроль над $n$ узлами сети, а продавец ожидает меньшее число подтверждений транзакций, то используя атаку Финни, атакующий может создать более длинную цепочку с транзакцией, переводящий средства на контролируемый им счет. После публикации цепочки в сеть, майнеры продолжат работать над более длинной цепочкой, содержащей блок с необходимой атакующему транзакцией. Данная атака называется атакой грубой силы. Для защиты необходимо проводить мониторинг сети блокчейна.

Атака 51% предполагает, что у злоумышленника сосредоточено более 50% вычислительной мощности сети блокчейна. В данном случае атакующий, обладая большинством голосов, может отклонять или подтверждать необходимые ему транзакции, создавая новые блоки быстрее, чем остальные майнеры. Защититься от такой атаки невозможно. Ее можно

только предотвратить путем мониторинга сети. В закрытых блокчейнах, необходимо удостовериться, что узлы сети находятся в доверенных местах.

При помощи атаки 51% можно создать новую криптовалюту. Алгоритм консенсуса PoW разрабатывался для доказательства целостности цепочки, не для предотвращения появления ответвлений.

Предположим, атакующие скрытно майнят несколько блоков, а затем «сбрасывают» их на основную сеть. Если за атакующим нет поддержки сообщества, честное меньшинство из остальных 49% отвергнет такую цепочку. Но несколько тайно найденных блоков позволяют атакующему отделиться от сети и продолжать майнить свою собственную цепочку, тогда как остальные майнеры продолжат старую. Так появляется два актива, один — всем известен, а другой – новый.

Пока майнеров достаточно, чтобы блокчейн работал, даже образуемые в результате хардфорка новые блокчейны не нанесут существенного вреда.

Противники POW-подхода, помимо ряда потенциальных проблем с безопасностью, выделяют следующие недостатки:

- Вероятность успешного создания следующего блока майнером прямо пропорциональна вычислительным мощностям, которыми он обладает, что приводит к постоянному наращиванию количества и качества оборудования каждого участника сети. Таким образом, майнинг с применением POW алгоритмов требует чрезвычайно много электроэнергии. Поэтому POW-подход является не самым лучшим решением с точки зрения энергоэффективности.
- Результаты вычисления хэш-функций нигде, кроме как в самой сети, не нужны. С момента появления технологии сообщество пыталось придумать способ направить все вычислительные ресурсы сети на решение какой-либо полезной математической или промышленной задачи, но в чистом виде это не удалось реализовать.

Попытки избавиться от недостатков POW привели к появлению многочисленных гибридных вариантов. PoW остается самым «старым» и надёжным методом обеспечения работы криптовалютных проектов. Но многие считают его эволюционно устаревшим, и появляющиеся в последнее время криптовалюты используют альтернативные варианты консенсусов — Proof of Stake, Proof of Capacity (майнинг на жестких дисках) и др. Эти консенсусы требуют меньших вложений для конечного потребителя и более доступны.

Если они покажут себя достаточно успешными, PoW-технологии майнинга и подтверждения могут в перспективе отойти на задний план.

## ЛИТЕРАТУРА

1. Back A. Hashcash-a denial of service counter-measure.〈http://www.hashcash.org/papers/hashcash.pdf〉, (дата обращения: 02.03.2020).
2. Xethalis G, Moriarty K, Claassen R, Levy J. An introduction to Bitcoin and blockchain technology, (дата обращения: 02.03.2020).
3. Eyal I, Sirer EG. Majority is not enough: Bitcoin mining is vulnerable, 2014.
4. Pilkington M. Blockchain technology: principles and applications. 〈https://ssrn.com/abstract=2662660〉, 2015, (дата обращения: 02.03.2020).
5. Vukolic M. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. in: International Workshop on Open Problems in Network Security, Springer, 2015, с.. 112–125.
6. Buchko S. How long do Bitcoin transactions take? 〈https://coincentral.com/how-long-do-bitcoin-transfers-take/〉, 2017, (дата обращения: 02.03.2020).
7. Kramer K, Hartnett S. When it comes to throughput, transactions per second is the wrong blockchain metric. 〈https://energyweb.org/2018/05/10/when-it-comes-to-throughput-transactions-per-second-is-the-wrong-blockchain-metric/〉, 2018, (дата обращения: 02.03.2020).

8. Back A, Corallo M, Dashjr L, Friedenbach M, Maxwell G, Miller A, et al. Enabling blockchain innovations with pegged sidechains , 2014, (дата обращения: 02.03.2020).
9. Ethereum Wiki. Proof of stake FAQ. 〈https://github.com/ethereum/wiki/wiki/ Proof-of-Stake-FAQ〉, 2017, (дата обращения: 02.03.2020).
10. Deetman S. Bitcoin could consume as much electricity as Denmark by 2020. 〈 https://motherboard.vice.com/en_us/article/aek3za/bitcoin-could-consume-as- much-electricity-as-denmark-by-2020〉, 2016, (дата обращения: 02.03.2020).
11. Martin W. The electricity required for a single bitcoin trade could power a house for a whole month. 〈http://uk.businessinsider.com/electricity-required-for-single-bitcoin-trade-could-power-a-house-for-a-month-2017-10?R=US&IR=T〉, 2017, (дата обращения: 02.03.2020).
12. Conti M., Kumar S., Lal C., Ruj S. A Survey on Security and Privacy Issues of Bitcoin // arXiv:1706.00916v3 [cs.CR], 2017, URL: https://arxiv.org/pdf/1706.00916.pdf, (дата обращения: 02.03.2020).
13. Poon J, Dryja T. The Bitcoin Lightning Network: Scalable off-chain instant payments. 〈 https://lightning.network/lightning-network-paper.pdf〉, 2016, (дата обращения: 02.03.2020).

**ГУРЬЕВА Елизавета Александровна** – студентка кафедры безопасности информационных систем и технологий, факультет компьютерных наук, Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 4, Харьков, Харьковская область, 61000; e-mail: mariia.popova26@gmail.com; ORCID: 0000-0002-9751-1717.

Научные интересы:

- *защита информации, технология блокчейн, теория передачи информации и кодирование*.

**ПОПОВА Мария Валерьевна** – студентка кафедры безопасности информационных систем и технологий, факультет компьютерных наук, Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 4, Харьков, Харьковская область, 61000; e-mail: mariia.popova26@gmail.com; ORCID: 0000-0002-9751-1717.

Научные интересы:

- *защита информации, технология блокчейн, программная инженерия*.

**ЕСИНА Марина Витальевна** - к.т.н., старший преподаватель кафедры безопасности информационных систем и технологий, факультет компьютерных наук, Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 4, Харьков, Харьковская область, 61000; e-mail: m.v.yesina@karazin.ua; ORCID: 0000-0002-1252-7606.

Научные интересы:

- *защита информации, постквантовая криптография, технология блокчейн.*

**УДК 004.738.5:621.395**

ДЕМ`ЯНЕЦЬ А. О.

# МОДЕЛЬ УПРАВЛІННЯ КОМП'ЮТЕРНОЮ МЕРЕЖЕЮ НА ОСНОВІ НЕЙРОМЕРЕЖЕВОЇ ТЕХНОЛОГІЇ

**Вступ.**

Постійне зростання кількості користувачів Інтернету висуває нові вимоги до пропускної здатності сучасних мереж зв'язку. Всесвітня павутина привела до появи різних видів трафіку. Графічна інформація, голосові дані, а також різні відеододатки пред'являють свої особливі вимоги до таких мереж.

Для задоволення всіх запитів одного збільшення ємності мережі недостатньо. Так як кількість користувачів Інтернету і різних мережевих додатків збільшується з кожним днем, мережа потребує коштів управління, які б забезпечили підтримку як існуючих, так і нових додатків, служб і послуг.

**Мета.**

Метою є використання штучної нейронної мережі яка дозволяє не тільки виконувати заздалегідь запрограмовану послідовність дії на заздалегідь визначеному наборі даних, а й аналізувати знову надходячю информацію, знаходити в ній закономірності, адаптуватися і проводити прогнозування.

**Спосіб управління мережею передачі даних на основі штучних нейронних мереж.**

Для управління сучасною мережею передачі даних необхідно застосовувати ефективні методи маршрутизації, управління трафіком і контролю завантаженості мережі, які грунтувалися б на даних, наданих інструментом прогнозування трафіку на основі попередніх значень. Найбільш підходящим інструментом для прогнозування є штучні нейронні мережі (ШНМ). Можна було б використовувати статистичні методи, проте в даний час структура мереж дуже швидко змінюється, і такі методи в деяких випадках можуть не впоратися з цим завданням. На відміну від цих методів, використання ШНМ дозволяє не тільки виконувати заздалегідь запрограмовану послідовність дії на заздалегідь визначеному наборі даних, а й аналізувати знову надходить, знаходити в ній закономірності, адаптуватися і проводити прогнозування. Таким чином, штучні нейронні мережі безперервно навчаються на основі попередніх значень. Розглянемо формулювання і основні принципи організації обчислень при вирішенні подібного роду завдань. Нехай для деякої групи вузлів мережі, з відомими відстанями між ними, потрібно знайти найкоротший маршрут. Позначимо вузли буквами A, B, C ...,а відстані між ними – dAB, dAC, …dBC … . Рішенням є впорядкована множина з n вузлів. Послідовність, в якій перебираються вузли, зручно представляти матрицею n × n, рядки якої відповідають вузлам, а стовпці - номерам вузлів в послідовності. Наприклад, для п'яти вузлів A, B, C, D, E, порядок пересування між цих вузлів може бути задана матрицею виду.

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| A | 0 | 1 | 0 | 0 | 0 |
| B | 0 | 0 | 0 | 1 | 0 |
| C | 0 | 0 | 0 | 1 | 0 |
| D | 1 | 0 | 0 | 0 | |
| E | 0 | 0 | 1 | 0 | 0 |

*Таблиця №1 Матриця виду.*

Таким чином, вузол C підключається першим, вузол A - другим і т. д. Довжина маршруту дорівнює dCA + dAE + ... + dDC .... У кожному стовпці і в кожному рядку цієї матриці може бути

тільки одна одиниця, т. К. В кожен момент підключається тільки один вузол і кожен вузол підключається тільки один раз. Матрицю можна сприймати як стан нейронної мережі з N = n2 нейронів. Завдання полягає в тому, щоб з n n 2!/2n маршрутів вибрати один з найменшою довжиною. Стан кожного нейрона описується двома індексами, які відповідають вузлу і порядковому номеру його підключення в маршруті. Наприклад, Yxj = 1 показує, що вузол x був j-м по порядку вузлом маршруту. Запишемо функцію обчислювальної енергії для мережі, призначеної для вирішення завдання маршрутизації, в якій стан з найменшою енергією відповідає найкоротшим маршрутом. У загальному вигляді така функція для даної мережі має такий вигляд:

$$E=-\frac{1}{2}\sum_{i}\sum_{j} w_{ij}Y_iY_j-\sum_{j} I_iY_j+\sum_{j} T_jY_j \tag{1}$$

де E - штучна енергія мережі; wij - вага від виходу нейрона i до входу нейрона j; Yj - вихід нейрона j; Ij - зовнішній вхід нейрона j; Tj - поріг нейрона j. Зміна енергії, викликане зміною стану j-го нейрона, можна обчислити таким чином:

$$\partial E=(\sum_{i} w_{ij}Y_i+I_j-T_j) \tag{2}$$

де $\delta Y_j$ - зміна виходу j-го нейрона. Кожному стану системи відповідає конкретна величина обчислювальної енергії. Сталий стан має меншу енергію, ніж нестійке. Еволюція системи в часі - це рух в просторі станів в пошуках мінімуму енергії і зупинка в цій точці.

Для даної системи функція енергії повинна відповідати таким вимогам [1]. По-перше, вона повинна підтримувати стійкі стану в формі матриці; по-друге, з усіх можливих рішень функція енергії повинна підтримувати ті рішення, які відповідають коротким маршрутам.

Цим вимогам задовольняє функція енергії виду:

$$E=-\frac{A}{2}\sum_{x}\sum_{i}\sum_{j} Y_{xi}Y_{xj}+\sum_{x}\sum_{i}\sum_{k} Y_{xi}Y_{kj}+\frac{C}{2}\sum_{x}\sum_{i} Y_{xi}-n \tag{3}$$

При цьому $Y_{xj}$ = 0, 1. Перші три члена вирази (3) підтримують перша вимога, четвертий член - друга; A, B, C, D - позитивні множники. Перший член дорівнює нулю, якщо кожен рядок x містить не більше однієї одиниці. Другий член дорівнює нулю, якщо кожен стовпець містить не більше однієї одиниці. Третій член дорівнює нулю,

якщо в матриці виду n одиниць. Таким чином, без урахування четвертого члена функція енергії має мінімуми (E = 0) у всіх станах, представлених матрицею з однією одиницею в кожному стовпці і кожному рядку. Всі інші стани мають більш високу енергію. Короткі маршрути підтримує четвертий член. У ньому індекси і беруться за mod n,

для того щоб показати, що i-й вузол сусідить в маршруті з (n - 1) -м і першим, т. е. $Y_{k,\,n+j} = Y_{kj}$. Четвертий член чисельно дорівнює довжині маршруту. Розкриваючи дужки в (3) і прирівнюючи коефіцієнти при квадратичних і лінійних членах в отриманому виразі і загальній формулі енергії, визначаємо матрицю зв'язків і зовнішні взаємодії:

$$W_{xi,kj}=A\partial_{xk}(B\partial_{xk})-C-DW_{xk}(\partial_{j,i+1}+\partial_{j,i-1}) \tag{4}$$

де $\delta_{ij}$ = 1, якщо i = j, в іншому випадку $\delta_{ij}$ = 0. Крім того, кожен нейрон має смещающий вага $I_{xi}$ = Сп. Перший член в (4) задає зв'язку нейронів в кожному рядку, другий - всередині кожного стовпця, третій і четвертий - глобальні зв'язки. I в (3), і в (4) три перших члена відповідають за загальні обмеження для будь-якого завдання маршрутизації і призводять мережу до фінального

станом у вигляді. Четвертий член управляє тим, яке з n!/2n можливих різних фінальних станів відповідає найкоротшим маршрутом.

Вибір маршрутів, максимізує ступінь вузла в мережі, дозволяє спланувати роботу так, щоб час її виконання було мінімальним. Ступінь вузла для цього випадку визначається як сума всіх потоків, що надходять у вузол і виходять від вузла. Наприклад, лінія, яка повинна активуватися, три рази додає потік з трьох одиниць до обох вузлів,які вона з'єднує. При цьому критерій якості роботи, який обирається для завдання маршрутизації, повинен відображати цілі, пов'язані з відповідною завданням складання плану роботи ліній зв'язку. Показник якості роботи повинен узгоджуватися зі структурою ШНМ Хопфилда. Вихідні напруги нейронів, які і визначають їх стану, такий ШНМ наближають до двійковим значенням у міру переходу мережі до стану стійкої рівноваги з мінімальною «енергією». З'єднання між нейронами і та j описуються вагою $T_{ij}$, який позитивний, якщо з'єднання збудливу, і негативний, якщо з'єднання гальмує.

Іноді ШНМ не може провести розпізнавання образів (зразків) і видає на виході неіснуючий образ. Це пов'язано з проблемою обмеженості можливостей штучної нейронної мережі. Для ШНМ Хопфилда число образів, що запам'ятовуються т не повинно перевищувати значення рівного $0{,}15 \cdot n$. Крім того, якщо два образи А і Б сильно схожі, вони можуть викликати у мережі перехресні асоціації, т. е. пред'явлення на вході мережі вектора А призведе до появи на її виходах вектори Б, і навпаки. Ще одним недоліком мереж Хопфілда є їх тенденція стабілізуватися в локальному, а не в глобальному мінімумі. У тих випадках, коли не потрібно, щоб ШНМ в явному вигляді видавала образ (зразок), т. е. досить отримати номер зразка, завдання може бути успішно вирішена за допомогою ШНМ Хеммінга (рис. 1). Моделі на основі ШНМ Хеммінга, в порівнянні з ШНМ Хопфилда, мають менші витрати на пам'ять і обсяг обчислень.

Штучна нейронна мережа Хеммінга складається з двох шарів: перший і другий шари мають m нейронів, де m - число зразків. Нейрони першого шару мають по n синапсів, з'єднаних з входами мережі.

Нейрони другого шару пов'язані між собою негативними зворотними синаптическими зв'язками. Роль першого шару умовна: скориставшись один раз на 1-му кроці значеннями його вагових коефіцієнтів, мережа більше не звертається до нього. Тому перший шар може бути виключений з мережі і замінений на матрицю вагових коефіцієнтів.

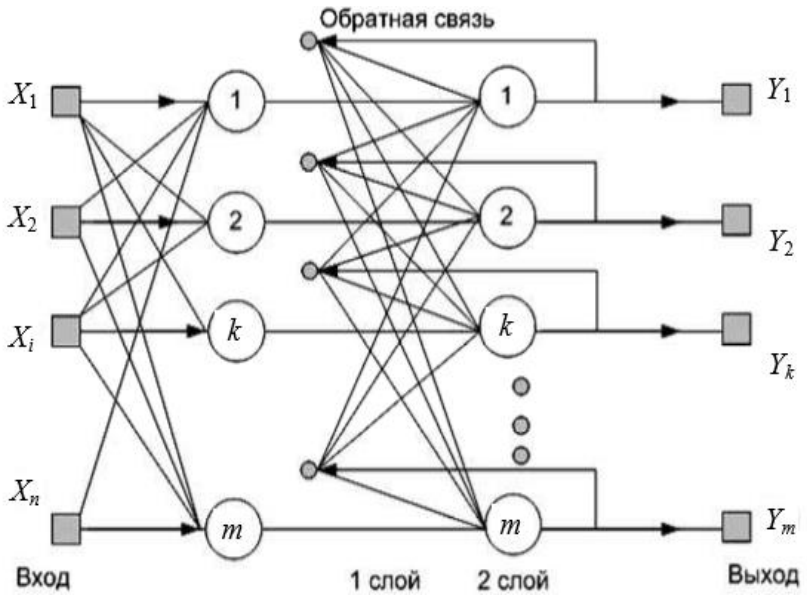



*Рисунок №1. Структурна схема ШНМ Хеммінга.*

Ідея роботи ШНМ Хеммінга - визначення відстані Хеммінга від тестованого зразка до всіх зразків. Відстанню Хеммінга називається число відрізняються бітів в двох бінарних

векторах. Мережа повинна вибрати зразок з мінімальною відстанню Хеммінга до невідомого вхідного сигналу, в результаті активізується тільки один вихід мережі,

який відповідає цьому зразку.

Найбільш широко застосовуваний протокол в Internet-мережах - це протокол TCP / IP. Для прискорення і оптимізації процесу передачі великих обсягів даних протокол TCP визначає метод управління потоком, званий методом ковзного вікна, який дозволяє відправнику посилати черговий сегмент, не чекаючи підтвердження про отримання в пункті призначення попереднього сегмента [2]. Якщо мати можливість заздалегідь отримувати дані про переповнення буфера обладнання або зростаючих затримках мережі, то можна управляти мережею для запобігання втрати даних і збільшення схоронності переданої інформації.

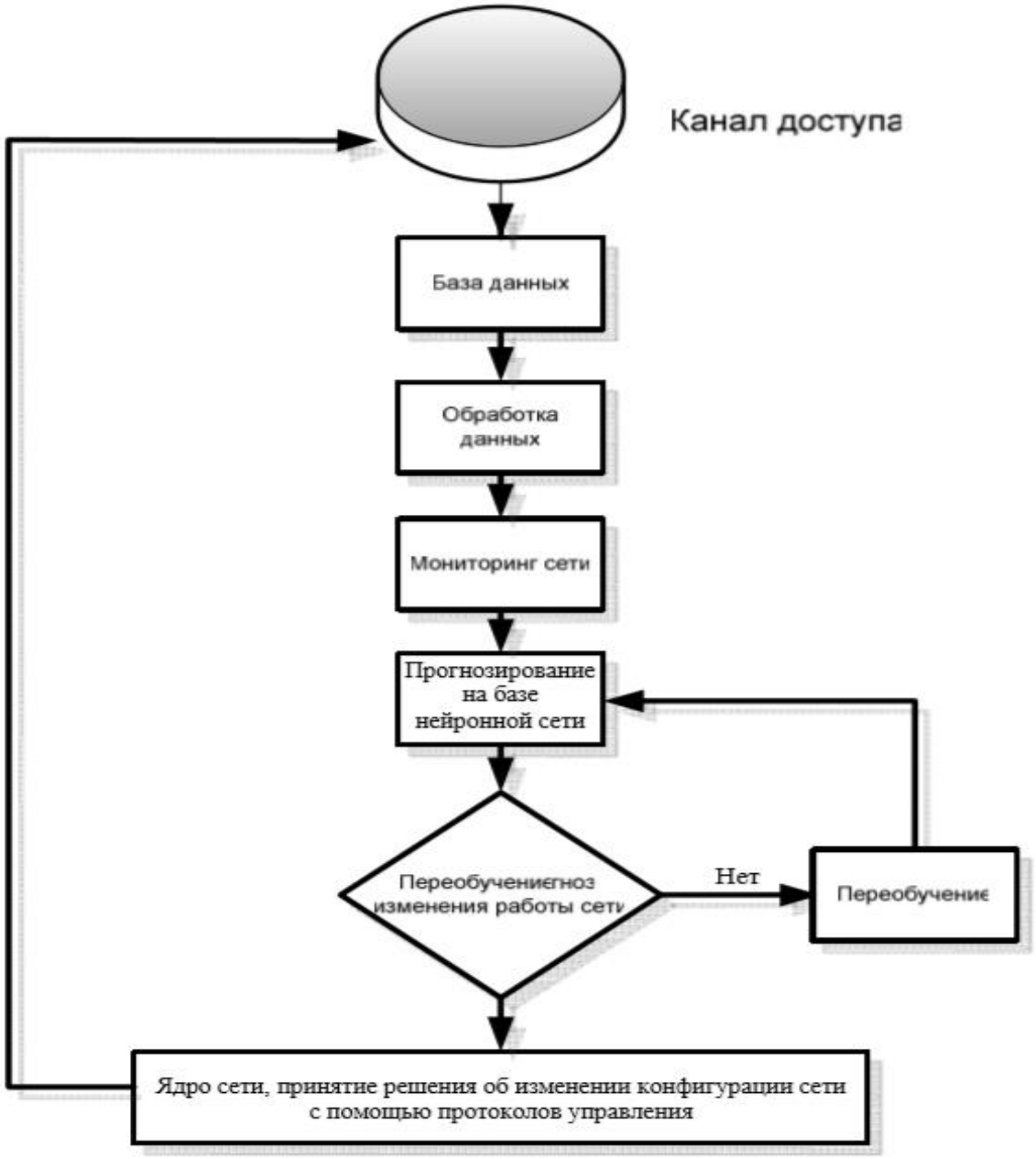



*Рисунок №2. Алгоритм управління мережею передачі даних на базі ШНМ.*

Це завдання може взяти на себе ШНМ Хеммінга (рис. 1) [3], на вхід якої подаються дані про обсяг буфера або затримки в мережі. Дана ШНМ може спрогнозувати майбутню поведінку трафіку мережі на основі відомих даних, зібраних заздалегідь.

Нижче представлений алгоритм, який реалізує прогноз і прийняття рішення для управління мережею передачі даних (рис. 2). В якості вхідних даних для процесу прогнозування нейронної мережі використовуються заздалегідь зібрані значення трафіку мережі в блоці бази даних. Поступив трафік N аналізується на наявність бажаних елементів: широкосмуговий трафік, надмірність, затримка мережі, смуга пропускання, надійність і завантаженість мережі. На основі вибраних даних відбувається прогнозування повторного появи обраного об'єкта дослідження (N + 1), крок прогнозування можна збільшити, але достовірність прогнозу значно зменшується.

На основі прогнозованих даних відбувається вплив на мережу передачі даних. Під впливом розуміється активація мережевих команд для запобігання появи прогнозованого об'єкта (керуючий вплив на пропускну здатність каналу, очищення буферів активних мережевих елементів, активація фільтрів і т. д.).

**Отримані результати.**

Очікується отримати результати:

• Запропонований алгоритм на управління мережі передачі даних на базі штучних нейронних мереж забезпечує можливість прогнозування трафіку, більш стабільної роботи мережі передачі даних зі збереженням самих даних, а також скорочення часу простою мережі в випадках виявлення небажаного трафіку;

• Описаний алгоритм дозволяє побудувати короткострокову модель прогнозу переміщення об'єктів в мережі і рівнів їх сигналів для зміни таблиць маршрутизації.

**Висновок.**

В даний час завдання прогнозування та управління трафіком мереж дуже важлива і вимагає найпильнішої уваги. Запропонований варіант алгоритму забезпечує можливість більш стабільної роботи мережі передачі даних зі збереженням самих даних, а також скорочення часу простою мережі в випадках виявлення небажаного трафіку.

ЛИТЕРАТУРА

1. Комашинский В. И., Смирнов Д. А. Нейронные сети и их применение в системах управления и связи. – М.: Горячая линия – Телеком, 2003. – 94 с.
2. Уоссерман Ф. Нейрокомпьютерная техника: Теория и практика. – М.: Мир, 1992. – 192 с.
3. Мамаев М. Телекоммуникационные технологии (Сети TCP/IP): учеб. пособие. – Владивосток, 2001.
4. Семейкин В. Д., Скупченко А. В. Применение модели на основе ИНС Хемминга для построения оптимальной системы маршрутизации в телекоммуникационных сетях / Международный форум информатизации (МФИ-2009). Тр. конф. «Телекоммуникации и вычислительные системы». – М.: МТУСИ, 2009.

**ДЕМ`ЯНЕЦЬ Артур Олексійович** – студент; студент кафедри теоретичної та прикладної системотехніки; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: ddemianetsart@gmail.com; ORCID 0000-0002-2461-2349.

Наукові інтереси:

– *штучні нейромережеві технології.*

**УДК 004.931**

ДМІТРІЕВ А.Г.

# ЗАСТОСУВАННЯ НЕЙРОМЕРЕЖЕВИХ ТЕХНОЛОГІЙ В РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ АВТОМОБІЛЕЙ НА ЗОБРАЖЕННЯХ ЗІ СКЛАДНИМ ФОНОМ

**Вступ**

Процес розпізнавання символів на зображенні зі складним фоном можна розділити на 3 етапи:

1. Виділення області розташування символів на зображенні.
2. Виділення окремих символів.
3. Розпізнавання символів

В даний час такі технології реалізуються трьома традиційними методами - структурним, признаковым і шаблонним. Кожен з цих методів орієнтований на свої умови застосування. Проте всі ці методи мають певні недоліки. Так, при обробці зображень найбільші спотворення, що впливають на результат розпізнавання, вносять зміни кута реєстрації, зміни масштабу, погодні умови. Також наявність сторонніх об'єктів на зображеннях зі складним фоном істотно знижують ефективність розпізнавання переліченими методами.

Аналіз методів виділення і розпізнавання об'єктів на зображеннях показав, що для вирішення даної проблеми ефективно використовувати штучні нейронні мережі в зв'язку з тим, що вони є слабо чутливими до спотворень вхідного сигналу, забезпечують можливість отримання класифікатора та добре моделюють складну функцію розподілу зображень символів.

Мета даної роботи полягає в розробці штучної нейронної мережі для покращення ефективності існуючих систем розпізнавання номерних знаків.

**Застосування згорткової нейронної мережі для виділення області розташування символів на зображенні**

Для вирішення завдання виділення області розташування символів на зображенні були обрані згорткові нейронні мережі, оскільки вони є стійкими до змін масштабу, зсувів, поворотів, змін ракурсу та інших спотворень.

Кожен шар згорткової нейронної мережі являє собою безліч площин, що складаються з нейронів. Нейрони однієї і тієї ж площині мають однакові вагові коефіцієнти синапсів, що ведуть до сусіднім областям попереднього шару. Кожен нейрон шару отримує вхідні дані з певної області попереднього шару (локальне рецептивне поле), тобто вхідне зображення попереднього шару сканується невеликим вікном і пропускається через набір синаптичних коефіцієнтів, після чого результат відображається на відповідному потоці нейронів. Таким чином, набір площин є картою характеристик, і кожна площина знаходить «свої» ділянки зображення в будь-якому місці попереднього шару. Розмір локального рецептивного поля вибирається самостійно в процесі розробки нейронної мережі [1, 2].

**Виділення окремих символів за допомогою гістограми середньої інтенсивності**

Після визначення місця розташування символів на зображенні, необхідно вибрати окремі символи для їх подальшого розпізнавання. Для цього пропонується використовувати метод, заснований на побудові гістограм середньої інтенсивності.

Область розташування символів, вибраних на попередньому етапі, сканується попіксельно зліва направо, зверху вниз, і обчислюється середня інтенсивність пікселів в кожному стовпці. У тих місцях, де немає символу, середня інтенсивність буде значно відрізнятися від інтенсивності в тих місцях, де присутні символи [3].

Щоб відокремити рядок символів від всього зображення, пропонується розрахувати горизонтальні гістограми. Оскільки найяскравіша область на зображенні - це фон номерної пластина, то два найбільших максимуми будуть відповідати областям 1 і 2 (рис. 1) [4, 5].


*Рис. 1. Побудова горизонтальних гістограм. Лініям 1 та 2 відповідають два найбільші максимуми. x – номер лінії зображення, y - середня інтенсивність лінії зображення*

Потім будуються вертикальні гістограми під взаємно перпендикулярним кутом n, і виділяються вже близько 10 максимумів в інтервалах між символами. Таким чином, виділяються області розташування окремих символів на номерній пластині (рис. 2).

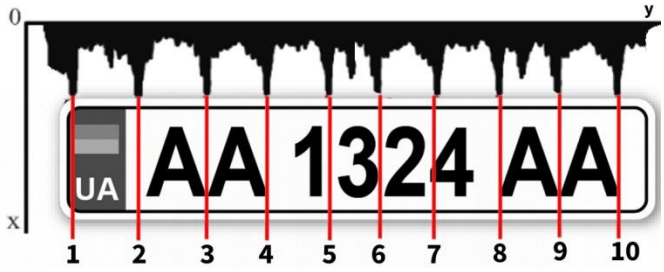



*Рис. 2. Побудова вертикальних гістограм. Лініям від 1 до 10 відповідають максимальні максимуми. y - номер лінії зображення, x - середня інтенсивність лінії зображення*

Під час реєстрації зображення номерних знаків піддається різним змішуванням і спотворень, отже, лінії, які відповідають областям 1 і 2, будуть розташовані не горизонтально, а під невідомим кутом. У зв'язку з цим пропонується побудувати не одну, а n гістограм середньої інтенсивності, кожна з яких побудована не горизонтально, а під заданим кутом [6].

Необхідна кількість гістограм середньої інтенсивності визначається з технічних умов реєстрації зображення, відповідно до цих умов кут повороту зображення не перевищує 20 ° по горизонталі як вправо, так і вліво, отже, n = 41. З n побудованих гістограм вибирається та, яка містить найбільше значення по y, оскільки найбільше значення буде відповідати областям 1 або 2 (рис. 1) [7, 8].

**Розробка згорткової нейронної мережі для розпізнавання символів на зображенні**

Для розпізнавання обраних символів була розроблена згорткова нейронна мережа з 4 прихованими шарами (рис. 3).

Перший шар є вхідним і складається з 28 × 28 = 841 нейронів.

Другий шар є згортковим і складається з шести площин розміром 24 × 24 = 578 нейронів.

Третій шар являє собою підвибірку і також складається з 5 площин розміром 12 × 12 = 144 нейронів.

Четвертий шар являє собою згортковий шар і складається з 50 площин розміром 8 × 8 = 64 нейронів.

П'ятий шар складається з 126 простих сигмоїдальних нейронів. Роль цього шару полягає в забезпеченні класифікації після того, як виконано витяг ознак і зменшення розмірів вхідних даних.

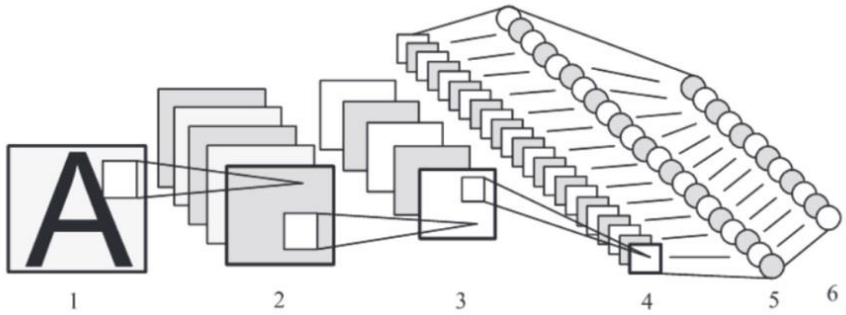



*Рис. 3. Архітектура згорткової нейронної мережі для розпізнавання символів: 1) введення; 2, 4) згорткові шари; 3) підвибірковий шар; 5, 6) шари звичайних нейронів*

Останній, шостий, шар є вихідним шаром і складається з 21 нейрона.

Згідно ГОСТ Р 50577-93, реєстраційні знаки транспортних засобів можуть містити такі символи: А, В, Е, К, М, Н, О, Р, С, Т, Х, всі цифри від 0 до 9. Тому вихідний шар виділяється з 21 нейрона, оскільки розпізнається 21 символ.

В якості активаційної функції був обраний гіперболічний тангенс, оскільки ця функція має ряд переваг для вирішення завдання, а саме:

- функція має безперервну першу похідну;
- функція має просту похідну, яка може бути обчислена через її значення, що дає економію обчислень.

Для навчання мережі використовувалася база з 60000 зображень рукописних чисел (MNIST) і була створена база з 20000 зображень букв. Розмір тестового зразка становить 10000 символів. Загалом навчання мережі займає близько 10-20 хвилин.

На основі представлених алгоритмів була розроблена програмна система, що забезпечує ймовірність розпізнавання автомобільних номерів на зображеннях зі складним фоном не менше 98% за таких умов реєстрації:

- час обробки: 35 мілісекунд;
- висота символу не менше 12 пікселей;
- освітлення номерного знака в діапазоні від 50 до 1000 люкс;
- кут відхилення номерного знака по горизонталі щодо реєстраційного пристрою до ± 60 °;
- кут відхилення номерного знака щодо записувального пристрою до ± 65 °;

**Порівняння технічних характеристик систем розпізнавання автомобільних номерних знаків**

В табл. 1 наведено порівняння технічних характеристик розробленої програмної системи з існуючими на ринку системам розпізнавання автомобільних номерів.

*Табл.1 Порівняння технічних характеристик систем розпізнавання автомобільних номерів*

| Назва системи | Ймовірність Розпізнавання % | Час розпізнавання | Оснащеність, лк | Мінімальна висота номера у кадрі |
|---|---|---|---|---|
| «Авто-інспектор» | 95 | не зазначено | не менше 50 | не зазначено |
| «Авто-інтелект» | 90 | не зазначено | | не зазначено |
| «SL-Traffic» | 90 | не зазначено | | 25 пікселей |
| «Дигнум-авто» | 90 | не зазначено | | не зазначено |
| «CarFlow II» | 93…98 | 60 мс | | не зазначено |
| Розроблене ПЗ | 98 | 35 мс | від 50 до 1000 | 12 пікселей |

Як видно з табл. 1, розроблена програмна система по всім параметрам не поступається існуючим на ринку системам, а за деякими параметрами перевершує.

**Висновок**

Було запропоновано використовувати згорткову нейронну мережу для забезпечення знаходження та виділення області розташування символів на зображеннях зі складним фоном. Для виділення окремих символів запропоновано використовувати алгоритм, заснований на побудові гістограм середньої інтенсивності пікселів. Для розпізнавання окремих символів розроблена згорткова нейронна мережа, що працює як класифікатор символів.

Отримані результати є дуже значущими, оскільки вони доводять ефективність використання згорткових нейронних мереж для виділення і розпізнавання автомобільних номерних знаків на зображеннях зі складним фоном, оскільки вони є стійкими до зсувів, поворотів, змін ракурсу та інших спотворень вхідних даних.

ЛИТЕРАТУРА

1. Макаренко А.А., Калайда В.Т. Методика локализации изображения лица для систем видеонаблюдения на основе нейронной сети. *Известия Томского Политехнического Университета.* 2006. Вып. 8. С. 113–118.
2. Елизаров А.И., Афонасенко А.В. Методика построения систем распознавания автомобильного номера. *Известия Томского Политехнического Университета.* 2006. Вып. 8. С. 118–121.
3. Болотова, Ю.А., Спицын В.Г. Распознавание символов на цветном фоне на основе иерархической временной модели с предварительной обработкой фильтрами Габора. *Электромагнитные волны и электронные системы*. 2012. Вып. 16. С. 14–19.
4. Буй Тхи Тху Чанг, Фан Нгок Хоанг. Логика и программное обеспечение для классификации цифровых изображений с использованием вейвлет-преобразования Хаара нейронных сетей. *Известия Томского Политехнического Университета.* 2011. Вып. 5. С. 103–106.
5. Bundzel M., Hashimoto S. Identification of objects in dynamic images based on the theory of predicting the memory of brain functions. *Journal of Intelligent Learning Systems and Applications*. 2010. Issue 4. P. 212–220.
6. Le Cun Y., Bengio Y. Convolutional networks for images, speechand time series. *The handbook of brain theory and neural net works*. 1998. Issue 7. P. 225–258.
7. Hansen D.W., Hansen J.P. Eye typing using Markovand active appearance models. *Applications of computer vi sion*. 2002. Issue 12. P. 132–136.
8. Feraud R., Bernier O. A fast and accura te face detector based on neural networks. *IEEE Transactions onpattern analysis and machine intelligence*. Issue 23. P. 42–53.

**ДМІТРІЄВ Артем Григорович** – студент; студент кафедри теоретичної та прикладної системотехніки; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: artdmitriev77@gmail.com; ORCID 0000-0003-2577-3934.

Наукові інтереси:

*– штучні нейромережеві технології.*

**УДК 004.056.55**

ДРОЗДОВА О.С., ГОРБЕНКО Ю.І.

# АНАЛІЗ ПОСТКВАНТОВОГО ЕЛЕКТРОННОГО ПІДПИСУ НА РЕШІТКАХ FALCON

**Вступ**

Сьогодні для розвитку квантових обчислень прикладаються значні зусилля. Цей факт мотивує криптографічну спільноту шукати альтернативи сучасній криптографії, яка виявиться нестійкою в разі створення квантового комп'ютера. Дана робота зосереджена на перспективному варіанті пост-квантового електронного підпису (ЕП) – Falcon. Нашою метою є аналіз Falcon на відповідність вимогам до пост-квантового ЕП та порівняння його показників з іншими 8-ми алгоритмами, які претендують на пост-квантовий ЕП. Результати даної роботи показують дослідникам орієнтир для розробки ефективних схем пост-квантового ЕП.

**1 Базові відомості про Falcon**

Falcon – це пост квантова схема електронного підпису заснована на математиці алгебраїчних решіток [1]. Дана схема є однією з дев'ятьох схеп ЕП, які пройшли до другого раунду процесу стандартизації пост-квантових алгоритмів інституту NIST. Основними компонентами Falcon є схема ЕП GPV(схема типу гешування-та-підпис) [2], решітки NTRU та швидка вибірка Фур'є. Дана схема є найбільш компактним ЕП при показниках швидкодії та стійкості на достатньому рівні.

**2 Аналіз рішень розробки Falcon з урахуванням вимог NIST**

При виборі компонентів для побудови схеми Falcon, автори опиралися на вимоги NIST [3]. До них належать: швидкодія, стійкість, компактність, простота конструкції, портативність.

**2.1 Компактність**

Головною позитивною рисою Falcon є його компактність, на відміну від інших схем ЕП. Це має значення при застосуванні алгоритму на малоресурсних пристроях. На першому рівні стійкості сума довжин відкритого ключа та підпису дорівнює 1587 байт, на третьому 2518 байт, на п'ятому – 3123 байт, ці показники є порівняні з аналогічними параметрами схеми RSA [4]. Використання алгоритмів стиснення впливає на розмір підпису. Також варто враховувати, що Falcon використовує арифметику з плаваючою точкою у процедурі підпису, хоча це не створює проблем для програмної реалізації, це може виявитися головним обмеженням, при реалізації на обмежених пристроях, які не мають модулів для арифметики з плаваючою точкою.

**2.2 Стійкість**

Безпека є найбільш значним критерієм при виборі пост-квантового алгоритму. Щодо стійкості Falcon в ROM та QROM, структура GPV поставляється з доказом безпеки у випадковому оракулі, а доказ стійкості у квантовій моделі випадкового оракула згодом був наданий у [5]. На відміну від схем, що використовують евристику Fiat-Shamir, яким важче забезпечити стійкість у QROM [6]. Також під час підпису повідомлення застосовується одноразове випадкове значення (нонс). Його розмір повинен обиратися таким чином, щоб ризики повторного використання не перевищували $2^{-192}$. 40 байт - це безпечна довжина для випадкової генерації без збереження будь-якого змінного стану. Falcon може використовувати набагато коротші нонси, якщо існує якийсь механізм, що запобігає повторному використанню. Програмна реалізація характеризується константним часом виконання, що забезпечує стійкість до часових атак, які можуть викрити інформацію про особистий ключ.

Також у [7] була запропонована покращена реалізація Falcon. Зокрема, для забезпечення константного часу виконання були оптимізовані операції з плаваючою точкою, застосовується нова Гаусова вибірка [8], та весь доступ до пам'яті здійснюється за несекретними адресами, тобто доступ не залежить від якої-небудь секретної інформації. Нова реалізація швидка та RAM-

ефективна. У ній підтримується система команд AVX2, яку можна застосовувати з Falcon-512 на Intel Core i7-6567U на частоті 3,3 ГГц. За секунду генерується приблизно 7700 підписів, використовуючи одне ядро.

Щодо захищеності Falcon додамо, що дана схема у значній мірі покладається на дискретну Гаусову вибірку над цілими числами. А способи її безпечної реалізації з урахуванням часових атак та атак по стороннім каналам залишаються майже не вивченими, за винятком кількох випадків [9,10].

### 2.3 Швидкодія

Процедури генерації підписів та перевірки проходять дуже швидко. Особливо це стосується алгоритму перевірки, але навіть алгоритм підпису може виконувати більше 1000 підписів в секунду на комп'ютері з середньою потужністю. Зокрема генерацію ключів можна прискорити, якщо попередньо обчислити дерево Falcon. Швидкість при формуванні підпису забезпечується завдяки спеціальній структурі модуля, щоб реалізувати пошук коротких векторів (підпису) як варіант алгоритму швидкого перетворення Фур'є. Щодо перевірки підпису: теоретико-числове перетворенню (NTT) прискорює дану процедуру, а також зменшує необхідний для перевірки розмір оперативної пам'яті.

Показники еталонної реалізації Falcon на процесорі Intel® Core® i7-6567U (тактова частота 3,3 ГГц) наведені у таблиці 1.

*Табл.1 Основні показники еталонної реалізації Falcon*

| степінь полінома | генерація ключа (мс) | генерація ключа (RAM) | підписів/с | перевірок/с | довжина відкритого ключа | довжина підпису |
|---|---|---|---|---|---|---|
| 512 | 6.98 | 14336 | 6081.9 | 37175.3 | 897 | 617.38 |
| 768 | 12.69 | 27648 | 3547.9 | 20637.7 | 1441 | 993.91 |
| 1024 | 19.64 | 28672 | 3072.5 | 17697.4 | 1793 | 1233.29 |

Як показано у новій реалізації Falcon [7], коли присутній вбудований модуль арифметики з плаваючою точкою, то швидкодія Falcon має такий же порядок як і найкращі схеми електронного підпису на еліптичній кривій. Наприклад Falcon-512 значно швидший, ніж RSA-2048 на тому ж апаратному забезпеченні, та при цьому забезпечує постквантову стійкість та відносно компактні підписи та відкриті ключі.

На невеликих вбудованих мікроконтролерах продуктивність Falcon погіршується через складність операцій з плаваючою точкою, але все одно становить допустимі значення. Наприклад, час генерації підпису - менше секунди навіть при середній робочій частоті (наприклад, 64 МГц). Більше того, перевірка підписів Falcon дуже швидка, що робить її придатною для звичайної невеликої вбудованої системи, яка перевіряє підпис на своєму программно-апаратному забезпеченні.

### 2.4 Портативність

Схема Falcon була успішно реалізована на ряді тестових систем, які мають апаратну реалізацію арифметики з плаваючою точкою [7]. У системах x86 тести виконувались як з AVX2 так і без.

### 2.5 Простота

Процедура підпису дуже проста: по суті, потрібно просто обчислити кілька операцій NTT і виконати гешування. Однак процедура генерації ключів, і швидка вибірка Фур'є нетривіальні для розуміння та делікатної реалізації, що є головним недоліком Falcon.

Цікавою особливістю Falcon є те, що він може бути перетворений безпосередньо на схему шифрування на основі ідентифікаторів [11].

## 3 Порівняння параметрів Falcon з іншими схемами пост квантового ЕП на решітках

*Табл.2 Порівняння стійкості схем підпису 2го раунду*


*Рисунок 1 – Порівняння швидкості підпису для претендентів на пост-квантовий ЕП*


*Рисунок 2 – Порівняння швидкості перевірки підпису для претендентів на пост-квантовий ЕП*


*Рисунок 3 – Порівняння розміру підпису претендентів на пост-квантовий ЕП*

**Висновки**

Схема Falcon є перспективним претендентом на пост-квантовий алгоритм електронного підпису. Адже Falcon на достатньому рівні задовольняє вимоги NIST, такі як: компактність, стійкість, швидкодія, портативність. Серед слабких місць даної схеми - складність реалізації операцій з плаваючою точкою, що уповільнює час виконання. Також стійкість Falcon у значній мірі покладається на дискретну Гаусову вибірку над цілими числами, стійкість якої до часових атак та атак по стороннім каналам залишається майже не вивченою. Ергономічні показники (швидкодія, простір для зберігання ключів та підпису) мають переваги серед інших алгоритмів ЕП, що можна бачити з рисунків 1-3.

ЛИТЕРАТУРА

1. Pierre-Alain Fouque et. al. Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU. Specifications v1.0. Режим доступу https://falcon-sign.info/falcon.pdf
2. Craig Gentry et. al. Trapdoors for Hard Lattices and New Cryptographic Constructions. Режим доступу https://eprint.iacr.org/2007/432
3. Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process. Режим доступу https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf
4. Режим доступу https://cyberleninka.ru/article/n/analiz-osnovnyh-suschestvuyuschih-post-kvantovyh-podhodov-i-shem-elektronnoy-podpisi/viewer
5. Dan Bone hey.al. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun.
6. Wang, editors, ASIACRYPT 2011, volume 7073 of LNCS, pages 41–69, Springer, Heidelberg, Germany.
7. Eike Kiltz et.al. A concrete treatment of fiat-shamir signatures in the quantum random-oracle model. Cryptology ePrint Archive, Report 2017/916, 2017. Режим доступу http://eprint.iacr.org/2017/916. 20
8. Thomas Pornin. New Efficient, Constant-Time Implementations of Falcon. Режим доступу https://eprint.iacr.org/2019/893
9. Daniele Micciancio. Gaussian sampling over the integers: Efficient, generic, constant-time. In Jonathan Katz and Hovav Shacham, editors, CRYPTO 2017, Part II, volume 10402 of LNCS, pages 455–485, 2017. Springer, Heidelberg, Germany.
10. Sujoy Sinha Roy et.al. Compact and side channel secure discrete Gaussian sampling. Cryptology ePrint Archive, Report 2014/591, 2014. Режим доступу http://eprint.iacr.org/2014/591. 21
11. Léo Ducas et. al. Efficient identity-based encryption over NTRU lattices. In Palash Sarkar and Tetsu Iwata, editors, ASIACRYPT 2014, Part II, volume 8874 of LNCS, pages 22–41, 2014. Springer, Heidelberg, Germany.

**ДРОЗДОВА Ольга Сергіївна** – магістр, молодш. наук. співроб; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: 4akolzinaolga@gmail.com; ORCID: 0000-0003-0073-9107.

Наукові інтереси:

*– електронний підпис, пост-квантова криптографія.*

**ГОРБЕНКО Юрій Іванович** – канд. тех. наук; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: gorbenkou@iit.kharkov.ua; ORCID: 0000-0002-8635-2327.

Наукові інтереси:

*– інфраструктура відкритого ключа, криптографічні засоби захисту інформації.*

**УДК 004.657**

ДУБИНКА А.Н., ЛАЗУРИК В.М.

# ОПТИМИЗАЦИЯ ДИЗАЙНА ЗАПРОСОВ НА ВЫБОРКУ

**Введение**

Информация – это неотъемлемая часть любого бизнеса, самый ценный его актив. Логически связанные данные (и их описание), характеризующие актуальное состояние некоторой предметной области, представляют собой базу данных (БД) [1, 2]. В настоящее время данные, хранящиеся в БД, стали более объемными, разнообразными и передаются с большей скоростью, объем их в самых разных аспектах жизни растет, и одновременно растут требования к системам управления базами данных (СУБД). Для удовлетворения возрастающих потребностей в хранении данных разрабатываются новые и совершенствуются уже существующие технологии, например, NoSQL, MapReduce, Hadoop, которые позволяют обеспечить хорошую горизонтальную масштабируемость. При этом не отказываются и от технологии Business Intelligence и реляционных систем управления базами данных с поддержкой языка SQL [3]. Очень популярны облачные хранилища, которые могут поддерживать многие СУБД. Сейчас разработаны новые решения, которые позволяют совместить все достоинства как NoSQL, так и реляционных баз данных. Это, так называемые, NewSQL базы данных. Наиболее ярким представителем этого решения является GOOGLE CLOUD SPANNER [4]. SPANNER поддерживает распределенные транзакции и позволяет системе масштабироваться на миллионы вычислительных узлов, работая с триллионами строк данных.

MySQL – свободная реляционная СУБД [5]. Первый выпуск MySQL был осуществлен в 1995 году компанией MySQL AB, сейчас разработку и поддержку осуществляет корпорация Oracle. Создана эта СУБД была как решение для малых и средних приложений, но она не только дожила до нынешних лет, но и повсеместно используется. MySQL – проверенная и очень мощная технология, на ней работает большое количество уже созданного программного обеспечения, в том числе и системы с большой нагрузкой. Многие современные высоконагруженные системы построены на основе MySQL, как одной из используемых в таком проекте СУБД. Часто рассматриваются решения совместного использования MySQL и NoSQL представителей БД.

При современном использовании MySQL важным моментом является время отклика сервера на запросы выборки и модификации данных. Для уменьшения этого времени, применяют различные методы оптимизации запросов. Поскольку MySQL используется не только в старом, уже разработанном программном обеспечении, но и при современной Веб разработке, важно учитывать некоторые моменты создания Веб приложений. Большинство программистов тестируют свои программы при малом количестве записей в таблицах, а проблемы у владельца сайта начинаются поздней, когда он наполнит свои таблицы реальными данными. Поэтому на первый план выходят вопросы оптимизации. Для этих целей используются разные приемы и методы.

В работе уделено внимание некоторым приемам оптимизации запросов на выборку, которые, большей частью касаются корректного дизайна запроса. Для рабочих серверов под большой нагрузкой время выполнения запроса может оказывать существенное влияние на поведение и надежность сервисов. Поэтому важным представляется реализация запросов на выборку таким образом, чтобы обеспечить минимальное время выполнения запроса.

**1. Общая постановка задачи и её актуальность**

Оптимизацию работы с БД можно разделить на 3 типа [6]: оптимизация запросов; оптимизация структуры; оптимизация сервера**.** Существуют факторы, оказывающие непосредственное влияние на скорость выполнения запроса [7]:

- индексы таблиц;
- условие WHERE (и использования функций MySQL, например, таких как IF или DATE);
- сортировка по ORDER BY;
- частое повторение одинаковых запросов;

- тип механизма хранения данных (InnoDB, MyISAM, Memory, Blackhole);
- конфигурации сервера (my.cnf / my.ini );
- большие выдачи данных (более 1000 строк);
- нестойкое соединение;
- распределенная или кластерная конфигурация;
- плохое проектирование таблиц.

Как правило, определение медленных запросов проводят уже на стадии, когда проект сдан в эксплуатацию и находится под рабочей нагрузкой. Это связано с тем, что разработчики, прежде всего, решают задачу поскорее вывести свой проект на рынок. На стадии разработки не уделяется внимания дизайну запроса, основной принцип − «работает, не трогай, пусть работает». Но, тем не менее, многие из таких запросов на рабочей нагрузке оказываются среди медленных. Часто для решения этой проблемы запросы приходится переписывать, вполне возможно, уже силами других профессионалов, тех, кто сопровождает проект.

В этой статье сделана попытка уделить должное внимание качественной разработке запросов на выборку, которые во многих проектах составляют основную часть обращений к базе данных. Структурированный язык запросов SQL дает возможность по-разному создать запрос с тем, чтобы получить правильные результаты. **Актуальной** представляется задача исследования зависимости времени выполнении запросов на выборку от их дизайна еще на этапе разработки.

Авторы провели эксперимент на учебной базе данных, сформулировав требования к запросу и предоставив возможность студентам, изучающим язык SQL, сформировать запрос. Оценивалась скорость выполнения созданных запросов в одинаковых условиях на ненагруженной системе. Другой эксперимент, проведенный авторами, коснулся не учебной, а типичной базы данных, в которой предусматривалась как запись в БД определенной информации о партии объектов с некоторыми свойствами, так и удаление (списание/продажа) отдельных объектов в разные моменты времени. Оценивалась скорость выполнения нескольких реализованных запросов на выборку, дающих правильные результаты. Условия выполнения для всех запросов устанавливались одинаковыми.

## 2. Учебная база данных «Устройства»

### 2.1. Проектирование

На рис. 1 изображен фрагмент схемы БД «Устройства», созданная в MySQL Workbench.

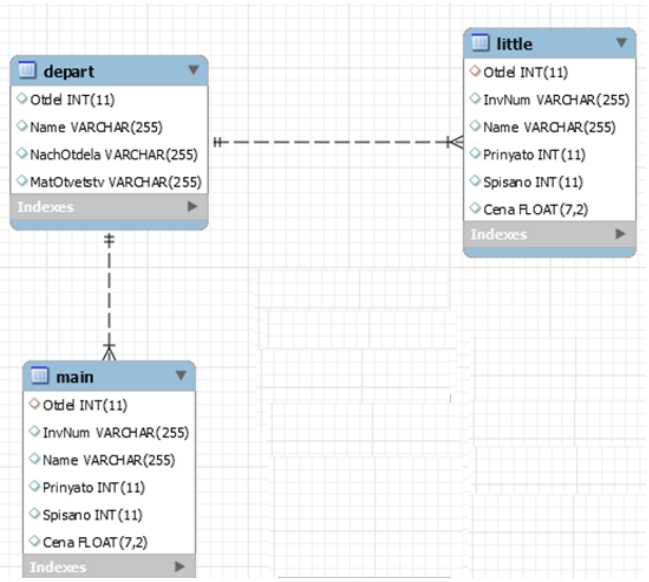



*Рис. 1 Структура учебной базы данных «Устройства»*

Таблица **depart** содержит данные о подразделениях организации. Основные характеристики {**Otdel** (номер отдела), **Name** (название отдела), **NachOtdela** (начальник отдела), **MatOtvetstv** (материально ответственный)}. Таблицы **little** и **main** имеют одинаковую

структуру. Данные, хранящиеся в таблице **little**, относятся к недорогим устройствам, в таблице **main** – к дорогостоящим. Атрибуты {**Otdel** (отдел, в который устройство принято), **InvNomer** (номер партии приборов), **Name** (название прибора), **Prinyato** (количество приборов в партии, поступивших на баланс предприятия), **Spisano** (количество приборов из партии, списанных с баланса предприятия), **Cena** (цена каждого прибора из партии)}.

### 2.2. Сводка по подразделениям

Задача формулируется так: предоставить сводку по отделам. Ожидаемый результат выполнения запроса представлен на рис. 2.

| Otdel | PrinMain | SpisMain | NalichMain | PrinLittle | SpisLittle | NalichLittle |
|---|---|---|---|---|---|---|
| 11 | 414976.00 | 142481.00 | 272495.00 | 70626.00 | 26091.50 | 44534.50 |
| 12 | 80379.44 | 39003.22 | 41376.22 | 0.00 | 0.00 | 0.00 |
| 13 | 1638148.00 | 562703.00 | 1075445.00 | 0.00 | 0.00 | 0.00 |
| 14 | 167858.40 | 91783.60 | 76074.80 | 0.00 | 0.00 | 0.00 |
| 15 | 0.00 | 0.00 | 0.00 | 49680.00 | 17897.00 | 31783.00 |
| 16 | 0.00 | 0.00 | 0.00 | 41600.00 | 20876.00 | 20724.00 |
| 17 | 0.00 | 0.00 | 0.00 | 50568.00 | 22794.20 | 27773.80 |
| 31 | 0.00 | 0.00 | 0.00 | 49904.00 | 20113.60 | 29790.40 |
| 37 | 785026.00 | 351532.50 | 433493.50 | 0.00 | 0.00 | 0.00 |

*Рис. 2 Результат выполнения запроса для создания сводки по отделам*

В итоговой таблице показаны для каждого отдела суммы принятых на баланс приборов, списанных, и имеющихся в наличии, как дорогостоящих, так и недорогих.

### 2.3. Инструменты для оценки производительности

Для оценки производительности запросов использовался инструмент бенчмарков (англ. *benchmark*). Бенчмарки используются в качестве тестов производительности для сравнения характеристик компьютерной системы [8]. В MySQL существует функция **BENCHMARK (loop_count, expr)**. Ее можно использовать для оценки скорости выполнения запроса.

Выполнение оператора **SELECT BENCHMARK (N, expr)** отличается от выполнения оператора **SELECT expr** с точки зрения количества включенных издержек. Эти операторы имеют разные профили выполнения и занимают разное время выполнения. Выполнение оператора **SELECT expr** включает в себя синтаксический анализатор, оптимизатор, табличную блокировку и т.д. Если этот оператор повторить **N** раз, каждый раз при выполнении его будут сопутствовать накладные расходы. Выполнение **SELECT BENCHMARK (N, expr)** включает все компоненты только первый раз, структуры памяти, уже выделенные однажды, снова используются, поэтому есть возможность таким образом определить эксплуатационные качества запроса за счет чистого времени его выполнения, удаляя "шум", представленный сетью, синтаксическим анализатором, оптимизатором, и т.д. Увеличение количества повторений **N** дает возможность в оценке времени минимизировать время на издержки.

### 2.4. Создание запросов на получение сводки по подразделениям

При формировании синтаксиса запроса для указания псевдонима использовано ключевое слово **AS.** Все запросы дают правильные результаты.

Вариант 1. Реализация запроса через подзапросы в разделе **SELECT** (листинг 1).

*Листинг 1.*

```
SELECT dep.Otdel,
 (SELECT Sum(main1.Prinyato*main1.Cena) FROM main AS main1
   WHERE dep.Otdel=main1.Otdel) AS PrinMain,
 (SELECT Sum(ifnull (main2.Spisano,0)*main2.Cena) FROM  main AS
main2
   WHERE dep.Otdel=main2.Otdel) AS SpisMain,
 (SELECT (Sum(main3.Prinyato*main3.Cena)-
  Sum(ifnull(main3.Spisano,0)*main3.Cena)) FROM  main AS main3
```

```
   WHERE dep.Otdel=main3.Otdel) AS NalichMain,
 (SELECT Sum(lit1.Prinyato*lit1.Cena) FROM little AS lit1
   WHERE dep.Otdel=lit1.Otdel) AS PrinLittle,
 (SELECT Sum(ifnull (lit2.Spisano,0)*lit2.Cena) FROM little AS lit2
   WHERE dep.Otdel=lit2.Otdel) AS SpisLittle,
 (SELECT (Sum(lit3.Prinyato*lit3.Cena)-
  Sum(ifnull(lit3.Spisano,0)*lit3.Cena)) FROM  little AS lit3
   WHERE dep.Otdel=lit3.Otdel) AS NalichLittle
FROM
 (SELECT main.Otdel FROM main
   UNION SELECT little.Otdel FROM little) AS dep;
```

Вариант 2. Реализация запроса через подзапрос в разделе **FROM** (листинг 2).

*Листинг 2.*

```
SELECT Result.Otdel, Result.PrinMain, Result.SpisMain,
 Result.NalichMain, Result.PrinLittle,Result.SpisLittle,
 Result.NalichLittle
FROM
 (SELECT main.Otdel, SUM(main.Prinyato * main.Cena) AS PrinMain,
   SUM(ifnull(main.Spisano,0) * main.Cena) AS SpisMain,
   SUM(main.Prinyato * main.Cena) –
   SUM(ifnull(main.Spisano,0) * main.Cena)) AS NalichMain,
   0 AS PrinLittle, 0 AS SpisLittle, 0 AS NalichLittle
  FROM main GROUP BY main.Otdel
 UNION
  SELECT little.Otdel, 0 AS PrinMain, 0 AS SpisMain, 0 AS
NalichMain,
   SUM(little.Prinyato * little.Cena) AS PrinLittle,
   SUM(ifnull(little.Spisano,0) * little.Cena) AS SpisLittle,
   SUM(little.Prinyato * little.Cena) –
   SUM(ifnull(little.Spisano,0) * little.Cena)) AS NalichLittle
  FROM little GROUP BY little.Otdel) AS Result
GROUP BY Result.Otdel;
```

Вариант 3 (листинг 3). Реализация запроса через два представления (**View**), связанные через **INNER JOIN**. Поскольку дорогостоящее и недорогое оборудование может находиться на балансе в разных отделах, при реализации каждого представления использовалось внешнее объединение с таблицей **depart,** для того чтобы получить одинаковый список отделов, как для представления **main1** (дорогостоящие приборы), так и для **little1** (недорогие приборы).

*Листинг 3.*

```
CREATE VIEW main1 AS
 (SELECT depart.Otdel AS Otdel,
   SUM(main.Prinyato * main.Cena) AS PrinMain,
   SUM(ifnull(main.Spisano,0) * main.Cena) AS SpisMain,
   SUM(main.Prinyato * main.Cena) –
   SUM(ifnull(main.Spisano,0) * main.Cena)) AS NalichMain
  FROM depart LEFT JOIN main ON depart.Otdel= main. Otdel
  GROUP BY depart.Otdel);
CREATE VIEW little1 AS
 (SELECT depart.Otdel AS Otdel,
   SUM(little.Prinyato * little.Cena) AS PrinLittle,
   SUM(ifnull(little.Spisano,0) * little.Cena) AS SpisLittle,
   SUM(little.Prinyato * little.Cena) –
   SUM(ifnull(little.Spisano,0) * little.Cena)) AS NalichLittle
  FROM depart LEFT JOIN little ON depart.Otdel= little. Otdel
  GROUP BY depart.Otdel);
SELECT main1.Otdel, main1.PrinMain, main1.SpisMain,
main1.NalichMain,
```

```
 little1.PrinLittle, little1.SpisLittle, little1. NalichLittle
FROM main1 INNER JOIN little1 ON main1.Otdel = little1.Otdel;
```

Вариант 4 (листинг 4). Реализация запроса через таблицу **depart** и два представления (**View**), связанные через **LEFT JOIN**. Представления **main2** и **little2** позволяют получить суммы полученного, списанного, и имеющегося в наличии дорогостоящего и недорогого оборудования соответственно. Представление **main2** содержит сводку по тем отделам, где имеется дорогостоящее оборудование, **little2** – по тем отделам, где имеется недорогое оборудование. Поскольку разное оборудование может находиться на балансе в разных отделах, в конечном запросе используется внешнее соединение таблицы **depart** с представлением **main2** и внешнее объединение с представлением **little2**.

*Листинг 4.*

```
CREATE VIEW main2 AS
 (SELECT main.Otdel AS Otdel,
   SUM(main.Prinyato * main.Cena) AS PrinMain,
   SUM(ifnull(main.Spisano,0) * main.Cena) AS SpisMain,
   SUM(main.Prinyato * main.Cena) –
   SUM(ifnull(main.Spisano,0) * main.Cena)) AS NalichMain
  FROM main GROUP BY main.Otdel);
CREATE VIEW little2 AS
 (SELECT little.Otdel AS Otdel,
   SUM(little.Prinyato * little.Cena) AS PrinLittle,
   SUM(ifnull(little.Spisano,0) * little.Cena) AS SpisLittle,
   SUM(little.Prinyato * little.Cena) –
   SUM(ifnull(little.Spisano,0) * little.Cena)) AS NalichLittle
  FROM little  GROUP BY little.Otdel);
SELECT depart.Otdel, main2.PrinMain, main2.SpisMain,
main2.NalichMain,
 little2.PrinLittle, little2.SpisLittle, little2.NalichLittle
FROM (depart LEFT JOIN main2 ON depart.Otdel = main2.Otdel)
             LEFT JOIN little2 ON depart.Otdel = little2.Otdel;
```

### 2.5. Результаты исследования

Время выполнения каждого из приведенных четырёх вариантов запросов для разного количества повторений в наших условиях проведения эксперимента приведено в Табл. 1. Эти значения могут быть иными при проведении вычислений в другой компьютерной системе. Но, можно утверждать, что зависимость скорости выполнения запросов от их написания сохранится в разных условиях проведения эксперимента.

*Табл. 1 Время выполнения четырёх вариантов запросов для разного количества повторений*

| *Количество повторений* | *Вариант 1, с* | *Вариант 2, с* | *Вариант 3, с* | *Вариант 4, с* |
|---|---|---|---|---|
| 100 | 0,001 | 0,002 | 0,015 | 0,002 |
| 1000 | 0,001 | 0,002 | 0,015 | 0,002 |
| 10000 | 0,001 | 0,002 | 0,015 | 0,002 |
| 100000 | 0,001 | 0,002 | 0,015 | 0,002 |
| 1000000 | 0,001 | 0,002 | 0,015 | 0,002 |

Результат зависимости времени выполнения четырех запросов от количества повторений запроса представлен на рис. 3.

Самым дорогостоящим оказался вариант 3 запроса на выборку. Он содержал два представления, у каждого из которых использовалось внешнее объединение таблиц с использование LEFT JOIN. Сами представления связаны через INNER JOIN. Все остальные запросы с той точностью, которую обеспечивает функция **BENCHMARK,** имеют приблизительно одинаковое время выполнения при мощности таблиц около 1500 записей.

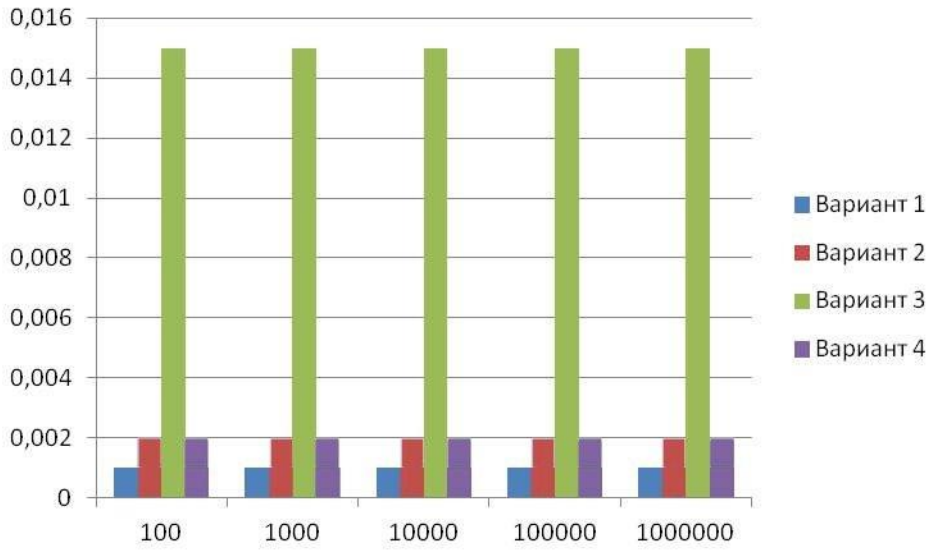



*Рис. 3. Результат зависимости времени выполнения запросов от количества повторений*

### 3. Задача учета наличия товаров на определенный момент времени

### 3.1. Проектирование

Рассмотрена модельная ситуация, представляющая собой обобщенный подход для целого класса IT решений, когда необходимо вести учет наличия товаров на указанный момент времени. У объектов выделены основные характеристики, остальные могут быть любыми. Каждый объект представляет собой набор элементов, подразумевается разовое поступление объектов на баланс подразделений предприятия. Анализируется изменение характеристик объекта с течением времени. Объект характеризуется названием элементов объекта, номером партии, принадлежностью подразделению, количеством элементов в партии, датой поступления, ценой элемента объекта. Базовыми операциями являются запись объекта в БД и удаление (списание/продажа) элементов объекта. БД должна предоставлять возможность получения сведений о состоянии объектов на любой указанный момент времени. В качестве имплементации рассматриваемой модели разработана БД «Devices», схема БД, созданная в MySQL Workbench, представлена на рис. 4.

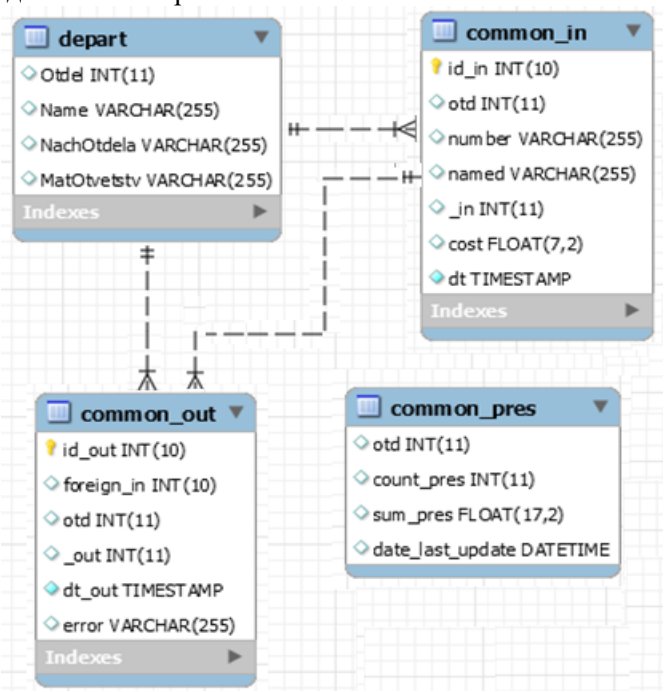



*Рис. 4. Структура базы данных «Devices»*

Таблица **depart** содержит данные о подразделениях организации. Основные характеристики {**Otdel** (номер отдела), **Name** (название отдела), **NachOtdela** (начальник отдела), **MatOtvetstv** (материально ответственный)}. Таблица **common_in** содержит данные о принятых на баланс предприятия устройствах, как и дорогостоящих, так и недорогих. Атрибуты {**id_in** (первичный ключ, однозначно определяющий характеристики партии приборов, поступивших на баланс предприятия), **otd** (номер отдела, на баланс которого поступили приборы), **number** (инвентарный номер партии приборов), **named** (название прибора), **_in** (количество приборов, принятых на баланс предприятия), **cost** (цена каждого прибора из партии), **dt** (дата принятия на баланс)}. Таблица **common_out** содержит данные о списанных с баланса предприятия устройствах (дорогостоящих и недорогих). Атрибуты {**id_out** (первичный ключ, однозначно определяющий характеристики партии приборов, списанных с баланса предприятия), **foreign_in** (внешний ключ, ссылка на партию приборов, поступивших на баланс предприятия), **otd** (номер отдела, с баланса которого были списаны приборы), **_out** (количество приборов, списанных с баланса предприятия), **dt_out** (дата списания с баланса), **error** (сведения о попытке списать с баланса предприятия количество приборов большее, чем есть на балансе предприятия в данный момент)}. Таблица **common_pres** содержит сведения об устройствах, находящихся на балансе предприятия в какой-то момент времени по отделам. Основные характеристики {**otd** (номер отдела, на балансе которого находятся приборы), **count_pres** (количество приборов, находящихся на балансе отдела), **sum_pres** (общая стоимость приборов, находящихся на балансе отдела), **date_last_update** (дата и время последнего обновления информации о наличии приборов в отделе)}.

### 3.2. Оптимизация условия в разделе WHERE

Существует несколько методов оптимизации оператора **WHERE**:

а) Устранение избыточных скобок.

Есть запрос на выборку принятых товаров, количество которых варьируется между 1 и 5, и цена – между 100 и 2000.

*Листинг 5.*

```
Select * from common_in where ((common_in._in>1 and common_in._in<5)
and common_in.cost<2000 or (((common_in._in>1 and common_in._in<5)
and (common_in.cost<2000 and common_in.cost>100))));
```

Этот запрос выполняется за 0,016 с. Если же устранить избыточные скобки таким образом, что это не повлияет на смысл запроса (листинг 6), то он будет выполняться за 0,001 с.

*Листинг 6.*

```
Select * from common_in where (common_in._in>1 and common_in._in<5
and common_in.cost<2000) or (common_in._in>1 and common_in._in<5 and
common_in.cost<2000 and common_in.cost>100);
```

б) Устранение постоянных условий.

Есть запрос для выборки принятых товаров, количество которых 1 либо 3. Помимо необходимых условий в этот запрос были добавлены условия, результат выполнения которых всегда одинаков.

*Листинг 7.*

```
Select * from common_in where (common_in._in>=1 and common_in._in=1)
or (common_in._in=3 and 1=1) or (common_in._in=5 and 1=3);
```

Время выполнения этого запроса составляет 0,0013 с. Если убрать из этого запроса постоянные условия (листинг 8), то время выполнения этого запроса составит 0,001 с.

*Листинг 8.*

```
Select * from common_in where (common_in._in=1 or common_in._in=3);
```

### 3.3. Присоединение условия из раздела HAVING

Если в запросе нет оператора **GROUP BY**, но есть операторы **WHERE** и **HAVING** (листинг 9), нужно условие из оператора **HAVING** присоединять к условию в операторе **WHERE** (листинг 10). Таким образом, запрос на выборку принятых товаров, количество которых варьируется между 2 и 7, а цена – между 100 и 1000, в форме, представленной в листинге 5, выполняется за 0,0014 с, а в форме, представленной в листинге 6, за 0,001 с.

*Листинг 9.*

```
Select * from common_in where (common_in.cost>100 and
common_in.cost<1000) having (common_in._in>=2 and common_in._in<7);
```

*Листинг 10.*

```
Select * from common_in where (common_in.cost>100 and
common_in.cost<1000) and (common_in._in>=2 and common_in._in<7);
```

### 3.4. Сохраненная процедура вместо запроса

Для возможности учета наличия товаров на указанный момент времени создана таблица **common_pres.** Заполнение таблицы можно осуществлять на какой-то определенный момент времени, а после использовать заполненную таблицу в качестве источника записей для реализации запросов устройств, которые есть сейчас в наличии – таблица **common_pres**, путём вызова сохраненной процедуры. Сохранённая процедура представлена в листинге 11, где **sum_in_by_dep** − представление, которое определяет по каждому подразделению количество принятого товара и сумму, а **sum_out_by_dep** − количество списанного товара и сумму соответственно.

*Листинг 11.*

```
CREATE VIEW sum_in_by_dep AS
(SELECT common_in.otd AS otd, SUM(common_in._in) AS count_in,
SUM((common_in._in * common_in.cost)) AS sum_in
FROM common_in GROUP BY common_in.otd);
CREATE VIEW sum_out_by_dep AS
(SELECT common_in.otd AS otd, SUM(common_out._out) AS count_out,
SUM((common_out._out * common_in.cost)) AS sum_out FROM (common_in
LEFT JOIN common_out ON ((common_in.id_in = common_out.foreign_in)))
GROUP BY common_in.otd);
CREATE PROCEDURE Sum_in_otd()
BEGIN
DELETE FROM common_pres;
INSERT INTO common_pres (otd, count_pres, sum_pres,
date_last_update)
SELECT sum_in_by_dep.otd,
(sum_in_by_dep.count_in-IFNULL(sum_out_by_dep.count_out,0)),
(sum_in_by_dep.sum_in-IFNULL(sum_out_by_dep.sum_out,0)), NOW()
FROM sum_in_by_dep LEFT JOIN sum_out_by_dep
ON sum_in_by_dep.otd = sum_out_by_dep.otd;
END
```

В результате вызова этой сохранённой процедуры, получаем итоговую таблицу, вид которой представлен на рис. 5.

| | otd | count_pres | sum_pres | date_last_update |
|---|---|---|---|---|
| ☐ | 11 | 639 | 176187.00 | 2019-03-16 12:24:42 |
| ☐ | 12 | 168 | 23685.55 | 2019-03-16 12:24:42 |
| ☐ | 13 | 324 | 979012.50 | 2019-03-16 12:24:42 |
| ☐ | 14 | 554 | 25807.90 | 2019-03-16 12:24:42 |
| ☐ | 15 | 170 | 8817.50 | 2019-03-16 12:24:42 |
| ☐ | 16 | 148 | 8132.00 | 2019-03-16 12:24:42 |
| ☐ | 17 | 172 | 9665.70 | 2019-03-16 12:24:42 |
| ☐ | 31 | 184 | 7132.10 | 2019-03-16 12:24:42 |
| ☐ | 37 | 398 | 402503.75 | 2019-03-16 12:24:42 |
| ☐ | 101 | 518 | 128120.00 | 2019-03-16 12:24:42 |
| ☐ | 102 | 68 | 412350.00 | 2019-03-16 12:24:42 |
| ☐ | 103 | 71 | 5230.00 | 2019-03-16 12:24:42 |

*Рис. 5. Итоговая таблица наличия товара*

Для оценки целесообразности разработки сохраненной процедуры и итоговой таблицы для нахождения стоимости всего имеющегося в наличии товара был выполнен запрос-выборка из исходных таблиц БД (листинг 12) и из итоговой таблицы (листинг 13). Время выполнения первого запроса при мощности исходных таблиц около 2500 записей составила 0,016 с, второго − 0,001 с. Если приложение к БД предполагает много запросов, источником данных которых может служить итоговая таблица, создание такой таблицы и сохраненной процедуры для ее заполнения полностью оправдано.

*Листинг 12.*

```
SELECT SUM((sum_in_by_dep.sum_in – IFNULL(sum_out_by_dep.sum_out,
0))) AS sum FROM (sum_in_by_dep LEFT JOIN sum_out_by_dep ON
sum_in_by_dep.otd =sum_out_by_dep.otd));
```

*Листинг 13.*

```
SELECT SUM (common_pres.sum_pres) AS sum FROM common_pres;
```

**4. Результаты и направление дальнейших исследований**

В статье уделено внимание исследованию зависимости времени выполнения запросов на выборку от их дизайна. Показано, что такой оценкой на этапе разработки нельзя пренебрегать. Проведенные эксперименты показали, что время выполнения запросов разного дизайна может отличаться более чем в 10-15 раз. Особенно это касается запросов с множественным внешним объединением таблиц, или с небрежно сформированными условиями выборки. В перспективе предполагается оценить скорость выполнения рассмотренных запросов при нагрузочном тестировании и сравнить с результатами, полученными на «спокойной» системе.

ЛИТЕРАТУРА

1. Когаловский М.Р. Энциклопедия технологий баз данных. М.: Финансы и статистика, 2002. 800 с. ISBN 5-279-02276-4.
2. Дейт К. Дж. Введение в системы баз данных = Introduction to Database Systems. 8-е изд. М.: Вильямс, 2005. 1328 с. ISBN 5-8459-0788-8 (рус.) 0-321-19784-4 (англ.).
3. James Manyika et al. Big data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute, June, 2011.
4. James C. Corbett, Jeffrey Dean, Michael Epstein, Andrew Fikes, Christopher Frost, JJ Furman, et al. Spanner: Google’s Globally-Distributed Database. Proceedings of OSDI, 2012.
5. MySQL 5.7 Reference Manual. Available at: https://dev.mysql.com/doc/refman/5.7/en/ (Last accessed: 02.02.2020).
6. Соколов Сергей. Оптимизация запросов в MySQL. [Электронный ресурс] Режим доступа: http://www.php.su/articles/?cat=phpdb&page=005 (последнее обращение: 12.02.2020).
7. Как оптимизировать MySQL запросы. [Электронный ресурс] Режим доступа: https://upread.ru/art.php?id=199 (последнее обращение: 10.02.2020).
8. Тест производительности. Материал из Википедии — свободной энциклопедии. [Электронный ресурс] Режим доступа: https://ru.wikipedia.org/wiki/Тест_производительности (последнее обращение: 11.02.2020).

**ДУБИНКА Анастасия Николаевна** – магистрант; магистрант факультета компьютерных наук, Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 4, Харьков-22, Украина, 61022; e-mail: dubinka.anastasya@gmail.com; ORCID: 0000-0001-6272-3018.

Научные интересы:

*– проектирование баз данных и SQL запросов.*

**ЛАЗУРИК Валентина Михайловна** – старший преподаватель; старший преподаватель кафедры искусственного интеллекта и программного обеспечения, Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 4, Харьков-22, Украина, 61022; e-mail: lazurik@hotmail.com; ORCID: 0000-0020-3340-9780.

Научные интересы:

*– разработка программного обеспечения в области радиационных технологий;*

*– базы данных; компьютерное моделирование педагогических измерений.*

**УДК 004.056.55**

Д'ЯЧЕНКО А.С., КАНДІЙ С.О. ОСТРЯНСЬКА Є.В.

# ПОРІВНЯЛЬНИЙ АНАЛІЗ СУЧАСНИХ СХЕМ ПОВНІСТЮ ГОМОМОРФНОГО ШИФРУВАННЯ

**Вступ**

Повністю гомоморфне шифрування це відносно новий вид криптоперетворень, який дозволяє виконувати певні математичні дії з зашифрованим текстом і отримувати зашифрований результат, який відповідає результату аналогічної операції, що проводиться з відкритим текстом [1]. Потенціал гомоморфного шифрування для прикладних задач досить великий.

У даній роботі розглянуто сучасні схеми повністю гомоморфного шифрування та проведено їх порівняльний аналіз за допомогою методу метод аналізу ієрархій.

**Порівняння схем гомоморфного шифрування**

Для порівняння було обрано чотири state-of-the-art схеми повністю гомоморфного шифрування: BGV, BFV, LWE, LTV. Тестування відбувалося на базі бібліотек SEAL, HeLib, cuHe. У якості параметрів було обрано $n = 4096$ та $\log(q) = 109$. Для порівняння використовувався метод аналізу ієрархій. Характеристики обраних схем наведено в таблиці 1. Шум та стійкість було оцінено за 10 бальною шкалою у порівнянні один з одним.

*Табл. 1. Характеристики схем гомоморфного шифрування*

| | Показник | Схема | | | |
|---|---|---|---|---|---|
| | | BGV | BFV | CKKS | LTV |
| 1 | Час розгортання ключів (ms) | 319658 | 305937 | 5754892 | 4258114 |
| 2 | Швидкість шифрування (ms) | 4538 | 8465 | 126751 | 3089 |
| 3 | Швидкість дешифрування (ms) | 916 | 1256 | 1573 | 341 |
| 4 | Шум | 6 | 5 | 5 | 7 |
| 5 | Стійкість | 6 | 7 | 7 | 5 |

Розглянемо детальніше сутність методу ієрархій. Вибір перспективного шифру – це ціль $X_0$. Здійснимо для головної цілі процедуру декомпозиції та побудуємо дерево цілей, де технічні показники $X_1^1$, показники цільового призначення $X_2^1$; час розгортання ключів $X_1^2$, швидкість шифрування $X_2^2$, швидкість дешифрування $X_3^2$, шум $X_4^2$, стійкість $X_5^2$. (Рис.1).

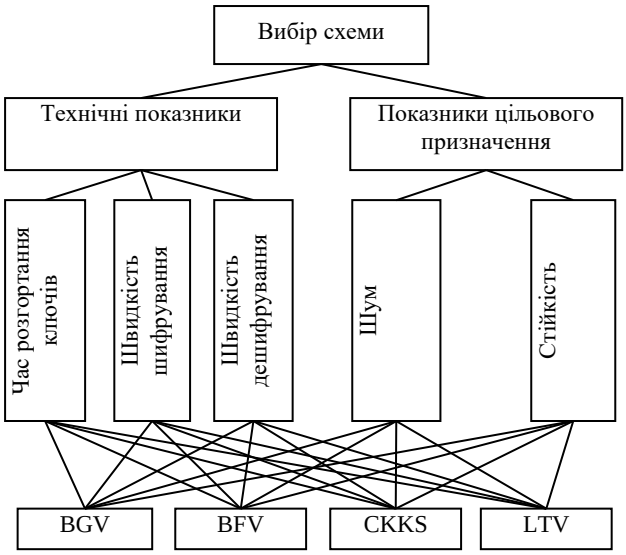



*Рис.1 Дерево цілей*

Оцінки значущості вкладу підцілей у досягнення цілі вищого рівня, згідно методу аналізу ієрархій здійснюються зверху вниз парним порівнянням. Сутність парного порівняння, наприклад $X_i^1$ та $X_j^1$ відносно $X^0$ до цілі полягає у оцінці (суджень) про те, у якій мірі $X_i^1$ більш важлива (більш вагома) для досягнення цілі $X^0$ ніж підціль $X_j^1$. Позначимо цю оцінку через $a_{ij}^{(1)}$. Подібні оцінки надаються експертами та носять суб'єктивний характер. При нашому порівнянні було вживано наступну шкалу оцінок (Таблиця 3).

*Таблиця 3. Шкала оцінок*

| Перевага $X_i^1$ над $X_j^1$ | Відсутня | Помірна | Значна | Велика | Дуже велика | Проміжні оцінки |
|---|---|---|---|---|---|---|
| $a_{ij}^{(1)}$ | 1 | 3 | 5 | 7 | 9 | 2, 4, 6, 8 |

Отримані експертні оцінки підлягають обробці наступним чином:

- обчислюється середнє геометричне для кожного рядка: $q_j^{(r-1)} = \sqrt[t_r]{a_{j1}^{(r)} ... \times a_{jj}^{(r)} \times a_{jt_r}^{(r)}}$;

- обчислюються нормовані значення: $\gamma_j^{(r-1)} = \dfrac{q_j^{(r-1)}}{\sum_{i=1}^{t_r} q_i^{(r-1)}}$ (1)

,де $\gamma_j^{(r-1)}$ характеризує значущість цілі $X_j^{(r)}$ для цілі $X^{(r-1)}$.

Використаємо приведений вище алгоритм послідовно для всіх дерев.

*Таблиця 4. Шкала оцінок $X^0$*

| $X^0$ | $X_1^1$ | $X_2^1$ | $q_j^{(0)}$ | $\gamma_j^{(0)}$ |
|---|---|---|---|---|
| $X_1^1$ | 1 | 1/2 | 0.7 | 0.33 |
| $X_2^1$ | 2 | 1 | 1.4 | 0.66 |

$$\left\| Y_1^{10} \right\| = \begin{pmatrix} 0.33 \\ 0.66 \end{pmatrix}$$

*Таблиця 5. Шкала оцінок $X_1^1$*

| $X_1^1$ | $X_1^2$ | $X_2^2$ | $X_3^2$ | $q_j^{(1)}$ | $\gamma_j^{(1)}$ |
|---|---|---|---|---|---|
| $X_1^2$ | 1 | 3 | 1/7 | 0.754199 | 0.247 |
| $X_2^2$ | 1/3 | 1 | 5 | 1.185615 | 0.388 |
| $X_3^2$ | 7 | 1/5 | 1 | 1.118689 | 0.366 |

$$\left\| Y_1^{21} \right\| = \begin{pmatrix} 0.247 \\ 0.388 \\ 0.366 \end{pmatrix}$$

*Таблиця 6. Шкала оцінок $X_2^1$*

| $X_2^1$ | $X_4^2$ | $X_5^2$ | $q_j^{(1)}$ | $\gamma_j^{(1)}$ |
|---|---|---|---|---|
| $X_4^2$ | 1 | 1/6 | 0.408248 | 0.143 |
| $X_5^2$ | 6 | 1 | 2.449489 | 0.857 |

$$\left\| Y_2^{21} \right\| = \begin{pmatrix} 0.143 \\ 0.857 \end{pmatrix}$$

*Таблиця 7. Шкала оцінок $X_1^2$*

| $X_1^2$ | $X_1^3$ | $X_2^3$ | $X_3^3$ | $X_4^3$ | $q_j^{(2)}$ | $\gamma_j^{(2)}$ |
|---|---|---|---|---|---|---|
| $X_1^3$ | 1 | 1/3 | 8 | 1/5 | 0.85444 | 0.152 |
| $X_2^3$ | 3 | 1 | 9 | 6 | 3.567621 | 0.633 |
| $X_3^3$ | 1/8 | 1/9 | 1 | 2 | 0.407962 | 0.072 |
| $X_4^3$ | 5 | 1/6 | 1/2 | 1 | 0.803107 | 0.143 |

*Таблиця 8. Шкала оцінок* $X_2^2$

| $X_2^2$ | $X_1^3$ | $X_2^3$ | $X_3^3$ | $X_4^3$ | $q_j^{(2)}$ | $\gamma_j^{(2)}$ |
|---|---|---|---|---|---|---|
| $X_1^3$ | 1 | 5 | 8 | 1/4 | 1.77828 | 0.273 |
| $X_2^3$ | 1/5 | 1 | 1/2 | 1/7 | 0.34569 | 0.053 |
| $X_3^3$ | 1/8 | 2 | 1 | 1/9 | 0.40821 | 0.063 |
| $X_4^3$ | 4 | 7 | 9 | 1 | 3.98428 | 0.611 |

*Таблиця 9. Шкала оцінок* $X_3^2$

| $X_3^2$ | $X_1^3$ | $X_2^3$ | $X_3^3$ | $X_4^3$ | $q_j^{(2)}$ | $\gamma_j^{(2)}$ |
|---|---|---|---|---|---|---|
| $X_1^3$ | 1 | 6 | 8 | 1/3 | 1.99688 | 0.298 |
| $X_2^3$ | 1/6 | 1 | 1/2 | 1/7 | 0.58738 | 0.087 |
| $X_3^3$ | 1/8 | 2 | 1 | 1/9 | 0.40796 | 0.061 |
| $X_4^3$ | 3 | 7 | 9 | 1 | 3.70779 | 0.553 |

$$\left\|Y_{1-3}^{32}\right\| = \begin{pmatrix} 0.273 & 0.152 & 0.298 \\ 0.053 & 0.633 & 0.087 \\ 0.063 & 0.072 & 0.061 \\ 0.611 & 0.143 & 0.553 \end{pmatrix}$$

*Таблиця 10. Шкала оцінок* $X_4^2$

| $X_4^2$ | $X_1^3$ | $X_2^3$ | $X_3^3$ | $X_4^3$ | $q_j^{(2)}$ | $\gamma_j^{(2)}$ |
|---|---|---|---|---|---|---|
| $X_1^3$ | 1 | 1/2 | 1/2 | 3 | 0.9306 | 0.193 |
| $X_2^3$ | 2 | 1 | 1 | 5 | 1.77828 | 0.368 |
| $X_3^3$ | 2 | 1 | 1 | 5 | 1.77828 | 0.368 |
| $X_4^3$ | 1/3 | 1/5 | 1/5 | 1 | 0.33981 | 0.070 |

*Таблиця 11. Шкала оцінок* $X_5^2$

| $X_5^2$ | $X_1^3$ | $X_2^3$ | $X_3^3$ | $X_4^3$ | $q_j^{(2)}$ | $\gamma_j^{(2)}$ |
|---|---|---|---|---|---|---|
| $X_1^3$ | 1 | 1/2 | 1/2 | 2 | 0.84089 | 0.182 |
| $X_2^3$ | 2 | 1 | 1 | 4 | 1.68179 | 0.364 |
| $X_3^3$ | 2 | 1 | 1 | 4 | 1.68179 | 0.364 |
| $X_4^3$ | 1/2 | 1/4 | 1/4 | 1 | 0.42045 | 0.091 |

$$\left\|Y_{4-5}^{32}\right\| = \begin{pmatrix} 0.193 & 0.182 \\ 0.368 & 0.364 \\ 0.368 & 0.364 \\ 0.070 & 0.091 \end{pmatrix}$$

Розраховуємо вклад цілей третього рівня для піддерев:

$$\left\|Y_1^{31}\right\| = \left\|Y_{1-3}^{32}\right\| \times \left\|Y_1^{21}\right\| = \begin{pmatrix} 0.273 & 0.152 & 0.298 \\ 0.053 & 0.633 & 0.087 \\ 0.063 & 0.072 & 0.061 \\ 0.611 & 0.143 & 0.553 \end{pmatrix} \times \begin{pmatrix} 0.247 \\ 0.388 \\ 0.366 \end{pmatrix} = \begin{pmatrix} 0.235475 \\ 0.290537 \\ 0.065823 \\ 0.408799 \end{pmatrix} \quad (2)$$

Розрахуємо вклад цілей третього рівня в досягнення головної цілі:

$$\left\|Y_1^{30}\right\| = \left\|Y_{1-2}^{11}\right\| \times \left\|Y_1^{10}\right\| = \begin{pmatrix} 0.235475 & 0.053625 \\ 0.290537 & 0.104676 \\ 0.065823 & 0.104676 \\ 0.408799 & 0.023023 \end{pmatrix} \times \begin{pmatrix} 0.33 \\ 0.66 \end{pmatrix} = \begin{pmatrix} 0.11309925 \\ 0.16496337 \\ 0.09080775 \\ 0.15009885 \end{pmatrix} \quad (3)$$

Таким чином, для досягнення головної цілі вибору кращої схеми перевагу має BFV (0.165), LTV (0.15), BGV (0.113), CKKS (0.091).

**Висновки**

1. Схеми повністю гомоморфного шифрування є одним з великих напрямків в криптології, проте досі не існує детального порівняння state-fo-the-art схем. У роботі було проведено первинний аналіз існуючих схем. За результатами аналізу можна сказати, що схема BFV має незначну перевагу над іншими схемами.
2. Схема LTV дещо поступається BFV, проте має великий потенціал для досліджень. Головною проблемою є вразливість в алгебраїчній структурі циклотомічного кільця, що дозволяє будувати ефективні атаки у багатьох випадках. Проблема може бути вирішена, якщо будувати поле на базі поліномів з великою групою Галуа, як це робиться в стандарті ДСТУ 8961:2019. Синтез повністю гомоморфної схеми на базі цього стандарта є задачею подальших досліджень.
3. Аналіз проводився з точки зору ефективності вирішення загальних задач. У більш специфічних випадках інші схеми можуть показати себе кращим чином. При виборі схеми повністю гомоморфного шифрування є важливий контекст задачі.
4. Всі сучасні схеми повністю гомоморфного шифрування грунтуються на складності задач в теорії граток, що додатково робить схеми гомоморфного шифрування захищеними від квантових атак при правильному виборі параметрів.

ЛІТЕРАТУРА

1. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, ITCS 2012, pages 309–325. ACM, 2012.
2. Homomorphic Encryption based on Hidden Subspace Membership Uddipana Dowerah and Srinivasan Krishnaswamy Indian Institute of Technology Guwahati.
3. Craig Gentry. A fully homomorphic encryption scheme. PhD thesis, Stanford University, 2009.
4. Craig Gentry Amit Sahai Brent Waters Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based.
5. Adriana L´opez-Alt On-the-Fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption.
6. Jung Hee Cheon1, Andrey Kim1, Miran Kim2, and Yongsoo Song1 Homomorphic Encryption for Arithmetic of Approximate Numbers.
7. Homomorphic Encryption Standardization, [Online]. Available: https://homomorphicencryption.org/.

**Д'ЯЧЕНКО Андрій Сергійович**, студент, Харківський національний університет імені В. Н. Каразіна, майдан Свободи 4, Харків, Україна, 61022; e-mail: andrey.090220@gmail.com; ORCID: 0000-0002-2342-4231.

Наукові інтереси:

*– Пост квантова криптографія, гомоморфне шифрування, криптографія на алгебраїчних решітках, криптографія на кодах, що виправляють помилки.*

**КАНДІЙ Сергій Олегович**, студент, Харківський національний університет імені В. Н. Каразіна, майдан Свободи 4, Харків, Україна, 61022; e-mail: kandy.sergey@yandex.ua; ORCID: 0000-0003-0552-8341.

Наукові інтереси:

*– Пост квантова криптографія, гомоморфне шифрування, криптографія на алгебраїчних решітках*

**ОСТРЯНСЬКА Єлизавета Вадимівна**, студент, Харківський національний університет імені В. Н. Каразіна, майдан Свободи 4, Харків, Україна, 61022; e-mail: antelizza@gmail.com; ORCID: 0000-0003-1412-8470.

Наукові інтереси: *– Пост квантова криптографія, гомоморфне шифрування, криптографія на алгебраїчних решітках.*

**УДК 003.26:004.056.55**

ЕЛИСЕЕВ Р.Ю., ОЛЕЙНИКОВ Р.В., РОДИНКО М.Ю.

# ФОРМИРОВАНИЕ БЛОКА ПОДСТАНОВКИ НА ОСНОВЕ ARX-ПРЕОБРАЗОВАНИЙ ДЛЯ МАЛОРЕСУРСНЫХ ШИФРОВ

## 1 Введение

Современные постквантовые блочные симметричные шифры вынуждены иметь размеры блока и ключа шифрования большие или равные 256 битам, но в то же время обладать высокой производительностью на максимально широком спектре устройств и платформ (как аппаратных, так и программных).

В работе представлены результаты исследования устойчивости к дифференциальному криптоанализу, производительности и статистической безопасности 32-битного блока подстановки на основе примитивных операций, доступных в широком спектре процессоров общего назначения и микроконтроллеров.

## 2 Блок подстановки на основе ARX-преобразований

Схема предложенного блока подстановки приведена на рис. 1.

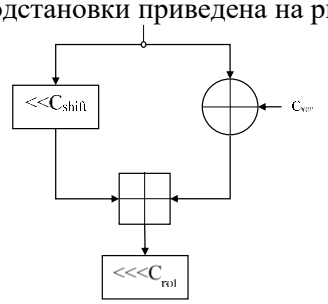



*Рис. 1 Схема блока подстановки на основе ARX-преобразований*

Один «раунд» преобразования состоит из 4 операций:

- побитового нециклического сдвига в сторону старших разрядов (<<);
- побитового исключающего ИЛИ (XOR, $\oplus$) с константой;
- сложения по модулю $2^{32}$ ($\boxplus$) результатов двух предыдущих операций;
- побитового циклического вращения в сторону старших разрядов (<<<).

Эта конструкция близка к умножению на константу с двумя значащими битами по модулю $2^{32}$ с последующим циклическим сдвигом и отличается от него только наличием дополнительной операции XOR с константой. Сдвиги подбираются таким обрпзом, чтобы минимизировать вероятности дифференциальных путей, проходящих через функцию.

## 3 Дифференциальный криптоанализ

В данной работе рассматривается в первую очередь простейший вариант с $C_{shift}=C_{rot}=3$, т.к. перебор всех возможных $C_{shift}=C_{rot}$ показал, что данный сдвиг является оптимальным с точки зрения снижения вероятности дифференциальных путей. Дифференциальный криптоанализ выполнялся на основе частичной таблицы дифференциальных разностей (ТДР), построенной в соответствии с подходом, предложенным в [1]. Частичная ТДР для предложенного блока подстановки строилась по XOR разностям на основе частичной ТДР для 32-битного сложения с граничной вероятностью 0.05. Константа в XOR операции не принималась во внимание. Поиск путей осуществлялся с помощью модифицированного алгоритма Мацуи [2], переориентированного с цепи Фейстеля на последовательное применение раундовой функции.

Лучшая найденная с помощью алгоритма Мацуи характеристика (таблица 1) имеет вероятность $2^{-74}$ для 13 раундов. Вероятности путей через отдельные раунды, полученные на основе частичной ТДР, отображены в колонке A. Экспериментальные оценки производились для 4 констант:

- $\pi$ – константа построенная на основе числа $\pi$ как $C_{xor}= (\pi-3) * (2^{32}-1)$;

- e – константа построенная на основе числа e как $C_{xor}= (e-2) * (2^{32}-1)$;
- 0 – $C_{xor}=0$;
- FFFFFFFF – $C_{xor}=2^{32}-1$.

*Табл. 1 Дифференциальный путь*

| № раунда | Входная разность | Выходная разность | A | π | e | 0 | FFFFFFFF |
|---|---|---|---|---|---|---|---|
| 1 | 80000000 | 00000004 | $2^{0}$ | $2^{0}$ | $2^{0}$ | $2^{0}$ | $2^{0}$ |
| 2 | 00000004 | 00000120 | $2^{-2}$ | $2^{-0,9850734557618972}$ | $2^{-1,0107156402523292}$ | $2^{-0,989363363891263}$ | $2^{-0,9925174236687938}$ |
| 3 | 00000120 | 00004100 | $2^{-3}$ | $2^{-3,0709665213541437}$ | $2^{-2,9919434451289417}$ | $2^{-2,9839316313723465}$ | $2^{-3,009262921328968}$ |
| 4 | 00004100 | 00124800 | $2^{-4}$ | $2^{-4,857259827883918}$ | $2^{-3,5872726614083574}$ | $2^{-3,1844245711374275}$ | $2^{-4,0516981876493645}$ |
| 5 | 00124800 | 04004000 | $2^{-5}$ | $2^{-4,717856771218502}$ | $2^{-5,351074440546879}$ | $2^{-4,895394956770689}$ | $2^{-5,237863830098888}$ |
| 6 | 04004000 | 20120001 | $2^{-4}$ | $2^{-4,006941609418847}$ | $2^{-3,8804446153047176}$ | $2^{-3,656535323845471}$ | $2^{-4,516222910048851}$ |
| 7 | 20120001 | 04100049 | $2^{-6}$ | $2^{-5,205563338195578}$ | $2^{-5,405069330187608}$ | $2^{-4,772012541265407}$ | $2^{-5,316168825598678}$ |
| 8 | 04100049 | 24801009 | $2^{-8}$ | $2^{-6,546245393148302}$ | $2^{-6,965784284662087}$ | $2^{-6,28771237954945}$ | $2^{-6,930160374931366}$ |
| 9 | 24801009 | 04048208 | $2^{-8}$ | $2^{-7,310432456049533}$ | $2^{-6,55979192498625}$ | $2^{-6,947862376664824}$ | $2^{-7,356975041986563}$ |
| 10 | 04048208 | 21049241 | $2^{-9}$ | $2^{-7,673002535434241}$ | $2^{-10,287712379549449}$ | $2^{-7,861447624847352}$ | $2^{-8,828280760912152}$ |
| 11 | 21049241 | 49000249 | $2^{-11}$ | $2^{-10,965784284662087}$ | $2^{-10,702749878828293}$ | $2^{-8,380821783940931}$ | $2^{-11,702749878828293}$ |
| 12 | 49000249 | 08008008 | $2^{-8}$ | $2^{-7,55979192498625}$ | $2^{-7,243318260190996}$ | $2^{-6,895394956770689}$ | $2^{-7,356975041986563}$ |
| 13 | 08008008 | 40240242 | $2^{-6}$ | $2^{-5,7027498788282935}$ | $2^{-6,34519787421021}$ | $2^{-5,710283551513701}$ | $2^{-6,519528054772523}$ |
| | | | $2^{-74}$ | $2^{-68,60166799694159}$ | $2^{-70,33107473525612}$ | $2^{-62,56518506156955}$ | $2^{-71,818403251811}$ |

Эксперименты для всех 4х констант проводились с использованием генератора псевдослучайных чисел с фиксированным начальным значением. В каждом случае исследовалось 10000 (одних и тех же) пар входных блоков. Оценивались однораундовые дифференциалы, вероятность полной характеристики рассчитывалась как произведение вероятностей однораундовых.

## 4 Производительность

Оценка производительности осуществлялась для 2 алгоритмов:

1. AES* – 32 битный блок подстановки на основе 4 последовательных применений SubBytes, ShiftRows, MixColumns преобразований AES [3] над 32 битным словом (для оптимизации производительности операции были объединены в одну табличную операцию);
2. $ARX_{p\_mul}$ – 13 раундов ARX преобразования, изображенного на рис. 1 с ненулевой константой. Количество раундов подобрано таким образом, чтобы количество операций было близко к количеству операций в AES*.

*Табл. 2 Результаты измерения производительности*

| Процессор | AES* | $ARX_{p\_mul}$ |
|---|---|---|
| i5-8265u +power (Windows 10 Education) | **589.9** | 563.05 |
| i5-8265u -power (Windows 10 Education) | **592.254** | 560.49 |
| i7-7700HQ +power (Windows 10 Education) | **563.554** | 525.769 |
| i7-7700HQ -power (Windows 10 Education) | **449.675** | 428.538 |
| i7-7700HQ +power (VirtualBox, Xubuntu 19.04) | **299.026** | **299.02** |
| i5-4300u +power (Windows 10 Education) | 334.537 | **394.923** |
| i5-4300u -power (Windows 10 Education) | 329.464 | **396.682** |
| i5-6400 (Windows 10 Professional) | **496.69** | 471.142 |
| i5-4460 (Windows 10 Professional) | 368.716 | **449.676** |
| i5-7400 (Windows 10 Professional) | **564.373** | 524.876 |
| E5-2643 (Hyper-V, Windows Server 2016) | **289.174** | 282.244 |
| Snapdragon 410 (ARM x32, Android 7.1.1) | 46.8565 | **89.5131** |
| MTK6577 (ARM x32, Android 4.0.4) | 43.7591 | **62.7212** |
| Helio P23 (ARM x64, Android 8.1.0) | 26.3378 | **27.8848** |
| MT6750 (ARM x32, Android 6.0) | **40.7978** | 21.5153 |

Для оценки производительности было собрано 4 бинарных файла:

- x86_x32 оптимизированная сборка для запуска на Windows системах. VS 2019;
- x86_x64 оптимизированная сборка для запуска на Linux системах, gcc 8.3.0;
- armeabi-v7a оптимизированная сборка для Android x32. Android NDK 21.0.6113669;
- arm64-v8a оптимизированная сборка для Android x64. Android NDK 21.0.6113669.

Все приложения собирались из одних исходников. Android приложения запускались из терминальной оболочки.

Алгоритм измерения производительности основан на шифровании 40 мегабайтного блока памяти. Выполнялось 10 групп измерений производительности по 5 замеров. В каждой из групп была выбрана максимальная производительность. На основе максимальных производительностей групп была рассчитана средняя, которая отображена в таблице. Жирным шрифтом выделены лучшие производительности на каждой платформе.

**5 Статистическая безопасность**

Оценка статистической безопасности выполнялась с использованием пакета NIST STS. Исследовались следующие алгоритмы:

1. $ARX_{p\_mul}$ – реализация 13 раундового алгоритма, схема которого изображена на рис. 1. $C_{shift}=C_{rot}=3$ для всех раундов $C_{xor}=(\pi-3) * (2^{32}-1)$;
2. $ARX_{p\_mul}$* – реализация 13 раундового алгоритма, схема которого изображена на рис. 1. $C_{shift}$ и $C_{rot}$ имеют псевдослучайные значения на каждом из раундов, $C_{xor}=(\pi-3) * (2^{32}-1)$;
3. AES* – 32 битный блок подстановки на основе 4 применений группы SubBytes, ShiftRows, MixColumns, AddRoundKey преобразований AES над 32 битным словом.

Для каждого из алгоритмов было сгенерировано по 200 последовательностей из 1000000 бит. Для генерации последовательностей блок подстановки работал в CTR режиме [4]. На графиках (рис. 2) изображено соотношение последовательностей, которые успешно прошли тест, к их общему количеству.

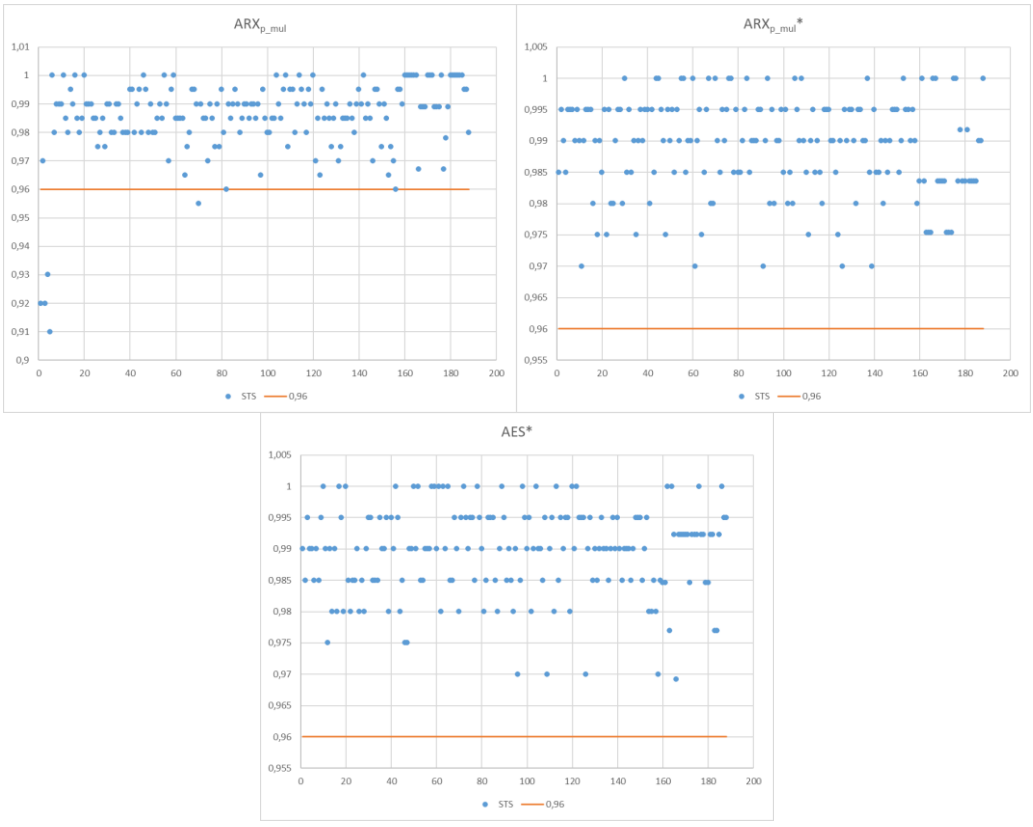



*Рис. 2 Результаты оценки статистической безопасности*

**6 Выводы**

Предложен компонент симметричного блочного шифра с простой структурой, производительность которого на ряде платформ выше, чем у традиционного подхода на основе блоков подстановки.

Предложенный блок подстановки обладает более высокой устойчивостью к дифференциальному криптоанализу ($2^{-74}$) при одинаковом количестве операций, чем блок на основе традиционных компонентов ($2^{-50}$), однако эти оценки не имеют строгого доказательства, в отличии от оценок шифров, основанных на стратегии широкого следа [5].

Предложенный блок подстановки обладает статистической безопасностью, аналогичной блокам на основе традиционных операций при условии использования различных констант сдвигов в отдельных раундах.

Экспериментальные оценки дифференциальных характеристик предложенного алгоритма зависят от константы, которая используется в XOR операции. Наиболее близкий к аналитической оценке результат получен для константы равной $2^{32}$-1.

ЛИТЕРАТУРА

1. Biryukov and V. Velichkov, “Automatic search for differential trails in arx ciphers,” in Topics in Cryptology (CT-RSA’14), pp. 227–250, Springer, 2014.
2. Matsui M. (1995) On correlation between the order of S-boxes and the strength of DES. In: De Santis A. (eds) Advances in Cryptology — EUROCRYPT'94. EUROCRYPT 1994. Lecture Notes in Computer Science, vol 950. Springer, Berlin, Heidelberg
3. Daemen, Joan; Rijmen, Vincent (March 9, 2003). "AES Proposal: Rijndael". National Institute of Standards and Technology. p. 1.
4. Diffie, Whitfield & Hellman, Martin. (1979). Privacy and authentication: An introduction to cryptography. Proceedings of the IEEE. 67. 397 - 427. 10.1109/PROC.1979.11256.
5. Daemen J., Rijmen V. The wide trail design strategy //IMA International Conference on Cryptography and Coding. – Springer, Berlin, Heidelberg, 2001. – С. 222-238.

**ЕЛИСЕЕВ Роман Юрьевич –** аспирант кафедры безопасности информационных технологий; Харьковский национальный университет радиоэлектроники; пр. Науки, 14, г. Харьков, 61166, Украина; email: r.yelysieiev@gmail.com; ORCID: 0000-0002-8847-490X.

Научные интересы:

– *анализ и синтез симметричных криптографических преобразований.*

**ОЛЕЙНИКОВ Роман Васильевич -** д. т. н., доцент; профессор кафедры безопасности информационных систем и технологий; Харьковский национальный университет имени В.Н. Каразина; пл. Свободы 4, 61022, Харьков; email: roliynykov@gmail.com; ORCID: 0000-0002-3494-0493.

Научные интересы:

– *анализ и синтез симметричных криптографических преобразований;*
– *сетевая безопасность.*

**РОДИНКО Мария Юрьевна –** аспирантка кафедры безопасности информационных систем и технологий; Харьковский национальный университет имени В.Н. Каразина; пл. Свободы 4, 61022, Харьков; email: m.rodinko@gmail.com; ORCID: 0000-0003-4692-9811.

Научные интересы:

– *методы построения и анализа симметричных криптографических преобразований.*

**УДК 004.056.55**

ЄСІНА М.В, ПОНОМАР В.А.

# ДОСЛІДЖЕННЯ ТА ПОПЕРЕДНІЙ АНАЛІЗ АЛГОРИТМІВ ЕЛЕКТРОННОГО ПІДПИСУ ДЛЯ ПОСТКВАНТОВОГО ПЕРІОДУ

**Вступ**

Нині все більш актуальною стає проблема захисту інформації та ресурсів від існуючих та потенційних криптоаналітичних атак з використанням квантового комп'ютера та квантової математики. Це обумовлено розробкою математичних основ для квантового комп'ютера та безпосередньо стрімким розвитком теорії та практики квантових комп'ютерів. На сьогодні вже створені та застосовуються 20, 53 та 72-кубітні квантові комп'ютери [6, 7]. І тому виникає необхідність розгляду та аналізу існуючих на сьогодні криптографічних алгоритмів, заміни їх параметрів або збільшення їх розміру, а також необхідність створення нових стандартизованих криптографічних алгоритмів з огляду на можливості квантового комп'ютера та його математичного забезпечення у постквантовий період. Вирішення цієї проблеми здійснюється на світовому рівні в процесі проведення NIST США міжнародного конкурсу [1].

NIST США, розуміючи необхідність пошуку нових стандартизованих криптопримітивів, особливу увагу звертає на створення електронного підпису (ЕП). На конкурс PQC NIST США було подано 22 кандидати на постквантовий стандарт ЕП, що базуються на використанні різних математичних основ. У ході 2-го етапу конкурсу проведено проміжний семінар, за рішенням якого рекомендовано 9 криптопримітивів типу ЕП до подальших досліджень.

Як показують попередні дослідження, надійною математичною основою, на якій можуть бути створені постквантові ЕП, нині вважаються алгебраїчні решітки [13]. На конкурс NIST було подано наступні механізми ЕП, що базуються на алгебраїчних решітках: CRYSTALS-Dilithium, FALCON та qTESLA. Вони зараз аналізуються на другому етапі конкурсу NIST [1].

Отже, актуальною є проблема аналізу, дослідження, оцінки та порівняння кандидатів на постквантові стандарти ЕП з використанням математики алгебраїчних решіток, а також розробки рекомендацій щодо вибору із них найбільш перспективних для майбутнього застосування. Метою роботи є оцінка, порівняння, обґрунтування та вибір перспективного ЕП для постквантового періоду, а також його більш детальне дослідження.

**Попередній аналіз та вибір перспективного кандидата на постквантовий електронний підпис**

При порівняльному аналізі використовувалася сукупність безумовних і умовних оцінок згідно [8, 9]. Умовними оцінками виступали наступні характеристики алгоритмів: $I_{ст.}$ – рівень криптографічної стійкості; $l_{в.к.}$ – довжина відкритого ключа; $l_{о.к.}$ – довжина особистого ключа; $l_{рез.}$ – довжина результату криптоперетворення; $T_{пр.}$ – швидкість прямого криптоперетворення; $T_{зв.}$ – швидкість зворотного криптоперетворення.

*Табл. 1 Експертні оцінки характеристик криптоалгоритмів*

| Показники / Експерти | $I_{ст.}$ | $l_{в.к.}$ | $l_{о.к.}$ | $l_{рез.}$ | $T_{пр.}$ | $T_{зв.}$ |
|---|---|---|---|---|---|---|
| 1 | 0,316 | 0,179 | 0,101 | 0,045 | 0,179 | 0,179 |
| 2 | 0,209 | 0,342 | 0,038 | 0,078 | 0,124 | 0,209 |
| 3 | 0,111 | 0,242 | 0,027 | 0,051 | 0,111 | 0,457 |
| 4 | 0,107 | 0,239 | 0,045 | 0,107 | 0,045 | 0,458 |
| 5 | 0,206 | 0,125 | 0,041 | 0,140 | 0,171 | 0,317 |
| W | 0,190 | 0,225 | 0,050 | 0,084 | 0,126 | 0,324 |

Експертні оцінки використовувалися для оцінки важливості кожної з наведених характеристик, а безпосередньо при порівнянні алгоритмів використовувалися об'єктивні числові значення, шкала оцінки, та вагові коефіцієнти важливості характеристик, що були отримані при експертному оцінюванні (таблиця 1) [9].

В даному дослідженні до алгоритмів висувалося додаткові безумовні вимоги [1]:

1) алгоритм повинен гарантувати, що найменше 3 рівень безпеки за класифікацією NIST;

2) якщо існує декілька варіантів наборів параметрів для одного алгоритму, то в порівнянні бере участь варіант, що гарантує найбільшу безпеку.

У табл. 2 наведені характеристики обраних для порівняння алгоритмів ЕП, що засновані на використані перетворень в алгебраїчних решітках [9].

У табл. 3 наведено результати оцінювання вибраних механізмів ЕП.

*Табл. 2 Характеристики алгоритмів ЕП в алгебраїчних решітках*

| Алгоритми | Тип | $I_{ст.}$ | $l_{в.к.}$ | $l_{о.к.}$ | $l_{рез.}$ | $T_{пр.}$ | $T_{зв.}$ |
|---|---|---|---|---|---|---|---|
| Dilithium_very_high | Lattices | 3 | 1760 | 3856 | 3366 | 2293141 | 611325 |
| falcon1024 | Lattices | 5 | 1793 | 8193 | 1330 | 19884364 | 1384574 |
| falcon768 | Lattices | 3 | 1441 | 6145 | 1077 | 13058641 | 1117624 |
| qTesla_256 | Lattices | 5 | 8224 | 8256 | 6176 | 8143869 | 1436949 |

*Табл. 3 Результати оцінювання механізмів ЕП в алгебраїчних решітках*

| Назва алгоритму | Значення оцінки | Назва алгоритму | Значення оцінки |
|---|---|---|---|
| Dilithium_very_high | 0,0964 | qTesla_256 | 0,0660 |
| falcon1024 | 0,0715 | falcon768 | 0,0712 |

На рис. 1 наведено гістограму відносної переваги алгоритмів, причому для більш узагальненого порівняння ЕП на алгебраїчних решітках наведено також оцінки щодо механізмів ЕП на основі MQ-криптоперетворень та функцій гешування. Як видно з рисунка, перше місце займає алгоритм Dilithium_very_high, друге місці займають алгоритми picnicl5fs, falcon1024, falcon768. Розглядаючи алгоритми, що зайняли третє місце, можна відмітити наступне: алгоритм falcon768 відрізняється від falcon1024 більш низькою стійкістю, falcon1024 має більш високий рівень стійкості, а falcon768 має більш високі інші основні характеристики.

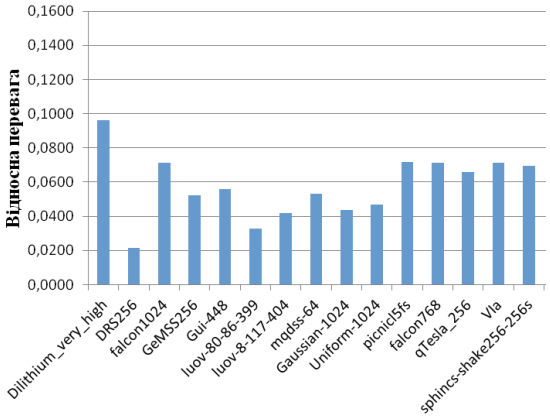



*Рис. 1 Відносна перевага алгоритмів ЕП*

**Модель порушника щодо постквантових механізмів електронного підпису**

Побудова моделі порушника необхідна для того, щоб розробити комплекс заходів із забезпечення захищеності алгоритму. Така модель може бути побудована з урахуванням різних критеріїв. Звичайно модель порушника розробляється з метою отримання відповідей на наступні питання: від кого необхідно захищати інформацію; якою є мета порушника; якими знаннями володіє порушник; які повноваження в системі має потенційний порушник; які методи і засоби використовує порушник.

По суті модель порушника – це опис можливих дій порушника, який формується на основі аналізу типу зловмисника, рівня його повноважень, знань, теоретичних та практичних можливостей. У якості порушника розглядається особа, що може отримати доступ до роботи з включеними до складу відповідної комп'ютерної системи (КС) засобами.

Порушники класифікуються за рівнем можливостей, що надаються їм штатними засобами КС. Кожний наступний рівень включає в себе функціональні можливості попереднього. Виділяються чотири рівні таких можливостей:

- 1-й рівень визначає найнижчий рівень можливостей проведення діалогу з КС – можливість запуску фіксованого набору завдань (програм), які реалізують заздалегідь передбачені функції обробки інформації;
- 2-й рівень визначається можливістю створення і запуску власних програм з новими функціями обробки інформації;
- 3-й рівень визначається можливістю управління функціонуванням КС, тобто впливом на базове програмне забезпечення системи, на склад та конфігурацію її устаткування;
- 4-й рівень визначається всім обсягом можливостей осіб, що здійснюють проектування, реалізацію і ремонт апаратних компонентів КС, аж до включення до складу КС власних засобів з новими функціями обробки інформації [8].

Припускається, що в своєму рівні порушник – фахівець вищої кваліфікації, який має повну інформацію про КС і КЗЗ.

Така класифікація порушників є корисною для використання в процесі оцінки ризиків, аналізу вразливості системи, ефективності існуючих і планових заходів захисту [8].

При розробці моделі порушника необхідно визначити, що і як має відображати модель. Для цього необхідно визначитись з необхідним ступенем її деталізації – можливі наступні:

- змістовна модель порушників – відображає причини та мотивацію дій порушників, переслідувані ними цілі і загальний характер дій у процесі підготовки і здійснення порушення цілісності алгоритму;
- сценарії впливу порушників – визначають класифіковані типи порушень з конкретизацією алгоритмів і етапів, а також способи дії на кожному етапі.

Під час побудови моделі порушника необхідно спочатку проаналізувати усіх користувачів системи, розподілити їх за категоріями та визначити найбільш критичні. Користувачі таких категорій будуть прийняті як можливі внутрішні порушники системи. Потім необхідно визначити, які категорії відвідувачів можуть бути зовнішніми порушниками.

Усіх можливих порушників необхідно класифікувати за різними показниками для того, щоб далі скласти модель порушника. Нижче наведені можливі види класифікацій порушників:

- за метою порушення. Проводиться для визначення мотивів порушника.
- за рівнем знань про алгоритм. Кожен порушник має певний рівень кваліфікації та поінформованості щодо алгоритму та його «секретів».
- за місцем дії. Класифікація проводиться для визначення розташування порушника відносно організації під час здійснення спроби НСД до інформаційного ресурсу.
- за методами і способами, якими вони користуються. Порушник може отримати конфіденційну інформацію та інформацію з обмеженим доступом, користуючись при цьому різними методами та засобами.

З урахуванням вказаного вище приймемо наступну модель порушника:

- захищати інформацію необхідно від зловмисника з квантовим комп’ютером, що має як мінімум декілька сотень кубітів потужності, або від добре укомплектованої хакерської групи, потрібно намагатися забезпечити захист навіть проти атаки великих професійних груп.
- порушник має за мету отримати інформацію будь-яким чином, використовуючи квантовий комп’ютер.
- у найкращому випадку порушник не має жодних знань ні про систему, ні про алгоритм. Але частіше за все порушник може знати тільки загальнодоступні дані або може перехватити частину інформації, або знати внутрішні дані (інсайдер).
- також у найкращому варіанті порушник не має повноважень в системі. Але, якщо це інсайдер, або людина, що отримала адміністративний доступ, то він зможе підміняти процеси в системі і перестроїти її «під себе», при умові, що у адміністратора є відповідні права.

Вважатимемо, що порушник використовує усі доступні йому ресурси – найпотужніші комп’ютери і необмежний час. Таким чином, у найгіршому випадку – порушник знає все про об’єкт та про всі засоби безпеки встановлені на ньому. У найкращому – порушник не знає нічого про об’єкт. У нашому випадку це рівноімовірні варіанти.

**Висновки**

1. Наразі створені та застосовуються 20, 53 та 72-кубітні квантові комп'ютери. Тому актуальною стає проблема захисту інформації та ресурсів від існуючих та потенційних криптоаналітичних атак.

2. У ході семінару 2-го етапу NIST США рекомендував до подальших досліджень 9 криптопримітивів ЕП. Дослідження можливостей створення ЕП, що можуть бути стійкими у постквантовий період, підтверджують, що надійною математичною основою для цього можуть бути алгебраїчні решітки.

3. На конкурс NIST США щодо стандарту ЕП було подано механізми ЕП, що ґрунтуються на алгебраїчних решітках Crystals-Dilithium, FALCON та qTESLA. Наразі вони досліджуються на 2-му етапі конкурсу NIST США.

ЛІТЕРАТУРА

1. Post-Quantum Cryptography. Round 2 Submissions. URL: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions.

2. Vadim Lyubashevsky Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. ASIACRYPT, 2009. – pp. 598–616.

3. Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann Practical lattice-based cryptography: A signature scheme for embedded systems. In CHES, pages 530–547, 2012.

4. Shi Bai, Steven D. Galbraith An improved compression technique for signatures based on learning with errors. In CT-RSA, pages 28–47, 2014.

5. Quantum Computing: Breaking Through the 49 Qubit Simulation Barrier. URL: https://www.ibm.com/blogs/research/2017/10/quantum-computing-barrier/.

6. IBM claims 'quantum supremacy' over Google with 50-qubit processor. URL: https://thenextweb.com/google/2017/11/14/ibm-claims-quantum-supremacy-over-google-with-50-qubit-processor/.

7. Google reclaims quantum computer crown with 72 qubit processor. URL: https://thenextweb.com/artificial-intelligence/2018/03/06/google-reclaims-quantum-computer-crown-with-72-qubit-processor/.

8. Горбенко Ю. І. Методи побудування та аналізу криптографічних систем : монографія. / Х. : Форт, 2015. 959 с.

9. Горбенко І. Д., Качко О. Г., Єсіна М. В., Пономар В. А. Методи, методика та результати порівняльного аналізу кандидатів на постквантовий стандарт електронного підпису // XX Ювілейна Міжнародна науково-практична конференція "Безпека інформації в інформаційно-телекомунікаційних системах", 22-24 травня, 2018, м. Буча. – С. 96-97.

**ЄСІНА Марина Віталіївна** – к.т.н., старший викладач кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків, Харківська область, 61000; e-mail: m.v.yesina@karazin.ua; ORCID: 0000-0002-1252-7606.

Наукові інтереси:

*– захист інформації, постквантова криптографія.*

**ПОНОМАР Володимир Андрійович** – к.т.н., науковий співробітник кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків, Харківська область, 61000; e-mail: Laedaa@gmail.com.

Наукові інтереси:

*– криптографічні перетворення, безпечне програмування, захист криптографічних засобів інформації.*

**УДК 004.4:004.5:62.51**

ЖИВАГА В.В, ШЕВЧЕНКО Д.О, МАЛАХОВА М.О.

# ІНТЕГРОВАНА INTERNET OF THINGS СИСТЕМА НА ОСНОВІ ОДНОПЛАТНОГО КОМП'ЮТЕРУ

*У роботі відображено інтегровану систему для Internet of Things (IoT), яка використовується для автоматизації процесів, аналізу продуктивності та сприяння людині, яку можна використовувати для великого спектру задач таких як «Smart House», вдосконалення роботи підприємств або лікування хворих. Запропонована система виконує всі сучасні вимоги до швидкодії та надійності, при доволі низькій ціні порівняно з аналогами, а також присутні широкі можливості для поліпшення та розширення цільових завдань.*

*The paper presents an integrated Internet of Things (IoT) system that is used to automate processes, performance analysis, and human assistance that can be used to a wide variety of tasks. Such as Smart house, enterprise improvement or patient care. The proposed system meets all modern requirements for performance and reliability at a relatively low price compared to adversary, and also has a many opportunities to improve and expand the tasks.*

Нині у всьому світі стрімко розвивається процес діджиталізації. Розвиваючи цей процес, *метою даної роботи* являється розробити доступну та інтуїтивно зрозумілу інтегровану систему для Internet of Things (IoT).

Зараз на промисловий сектор(як і на більшість інших різновидів діяльності) має вплив цифрова трансформація, скажімо, завдяки таким концепціям, як промисловий Інтернет речей (Industrial Internet of Things, IIoT – надалі II). За його допомогою стає доступною можливість використання гнучких ланцюжків постачання, які дозволяють підприємствам адаптувати власні промислові потужності для вдоволення мінливого попиту, зменшувати час виробництва або користуватися механізмами розумного обслуговування для зменшення простоїв промислових потужностей та ланцюжків постачання до мінімуму. До того ж, інтенсивний моніторинг у виробничих цехах значно полегшує встановлення складних етапів промислових процесів, зокрема, для їх покращення та обмеження пересування виробів. Завдяки міжмашинним комунікаціям (M2M) та промисловому Інтернету речей компанії мають змогу з'єднувати своє обладнання, за допомогою оснащення його комунікаційними властивостями та функціями, і вводити «розумні» рішення для трансформації зібраних даних в пристрій підвищення ефективності.

Проте для роботи в покращеному і автономному режимі промислові майданчики на основі IoT потребують правильно побудованої інфраструктури та використання штучного інтелекту. При правильній реалізації IoT є можливість знаходження та усунення зайвих економічних витрат як для підприємства, так і для приватної оселі.

Маючи потужні інструменти в галузі, недостатньо спеціалістів, які здатні їх використовувати. Компанії використовують моделі II, але не завжди мають у власному розпорядженні інструменти для підтримки власних систем. Складна технологія потребує глибоких знань для виявлення і рішення проблем. Але в найближчий час в області II може знову відбутися вибухове зростання.

Виконуючи аналіз подібних систем[1], були зроблені висновки, що нині не існує адаптивних систем автоматизації. Ринок висуває готові рішення окремих завдань, які не мають змоги модернізації системи та розширення їх можливостей. За мету цієї наукової роботи взята розробка інтегрованої системи для IoT, яку можна було б використовувати для великого спектру завдань. Наприклад «Smart House», вдосконалення роботи підприємств або лікування хворих. Завданням даної наукової роботи було розробка системи, під назвою Universal IoT System (UIoTS), яка спирається на системну архітектуру «Smart House» Інтернету речей *(IoT)* та

відображає прикладну систему, яка застосовує адаптивне управління з метою продуктивного керування навантажувальними приладами, щоб сприяти тепловому комфорту, забезпеченню контроля за вологістю, економічністю ресурсів та безпекою у приміщенні.

IoT – це система автоматизації, основою якої є певна кількість взаємопов'язаних між собою підсистем, які відповідають за окремі задачі та мають один центральний пристрій управління та/або проміжний пристрій, який також має можливість управління, якщо кількості даних достатньо для аналізу та прийняття рішення.

UIoTS розроблена згідно European Telecommunications Standards Institute (ETSI). Згідно до цього стандарту IoT має тришарову ієрархію нерозривно пов'язану зі своїми функціями. Ієрархія представлена трьома шарами:1) датчиком; 2) мережею; 3) додатком. Більш детально пояснено на рис.1 .

Широка інтегрованість, розповсюдженість та потужний обчислювальний потенціал реалізовані на одній платформі є основою для досягнення поставленої мети. Також суттєвою перевагою була б наявність бази знань що до вирішення нетипових завдань. Щоб побудувати таку платформу необхідно об'єднати одноплатний комп'ютер (наприклад Raspberry Pi 3 ), який забезпечить управління та розрахунки завдяки своїм обчислювальним можливостям, та Arduino, що відповідатиме збір та обробку аналогових даних, передачу цифрових даних на одноплатний комп'ютер та підключення периферійних пристроїв таких як датчики, мотори та інші.

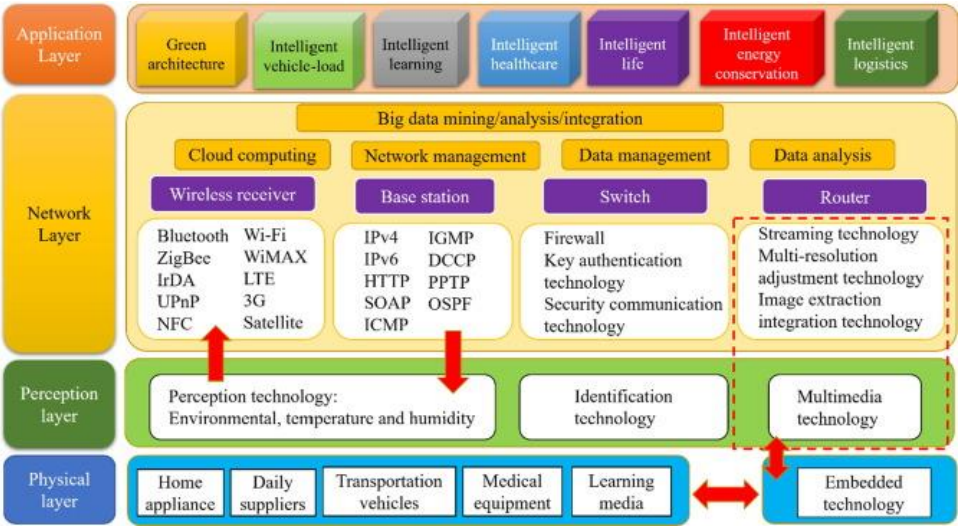



*Рис. 1 – IoT архітектура в нашій системі.*

Архітектура системи представлена чотирма частинами:

1. Зондуванням довкілля займається перша частина. Вона є частиною групи обладнання екологічного зондування , яке знаходиться в приміщенні. Це обладнання представлено датчиками, виконавчими пристроями та аналізуючим пристроєм Arduino[2]. Зондувальна частина відповідає за збір даних та виконавчі дії.

2. Бездротову передачу забезпечує друга частина за допомогою передачі ZigBee. Її основна мета полягає в основному в обміні обробленими Arduino сигналами, які аналізуються на одноплатному комп'ютері.

3. Третя частина відповідальна за аналіз та зберігання інформації. Аналіз базується в основному на реалізації індикаторів і стандартів, згаданих у частині II, які аналізуються, обробляються,зберігаються та відображаються програмним забезпеченням.

4. Четверта частина на основі отриманого аналізу здійснює контроль навантажень для якісного керування внутрішнім середовищем. Для цього використовуються навантажувальні пристрої, такі як зумер, електричний вентилятор, зволожувач повітря, кондиціонер і тд.

Автоматичне управління системою[3] може здійснюватись в залежності від становища навколишнього середовища або переваг користувача. Це реалізовано наступним чином. Одноплатний комп'ютер подає сигнал на Arduino, яка здійснить регулювання температури, керуючи кондиціонером, вентилятором та зволожувачем повітря або увімкне пожежну тривогу ( у випадку критичної ситуації) відповідно до сигналу, отриманого ZigBee[4]. Повна архітектура системи показана на рис. 2.

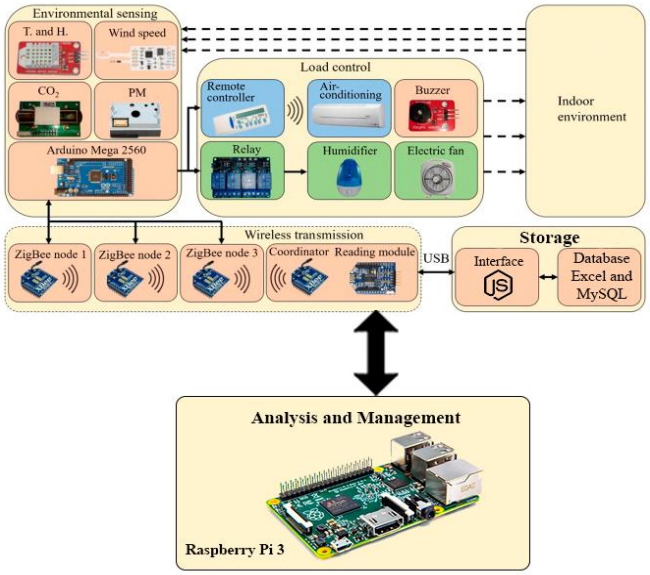



*Рис. 2 – Схема архітектури передачі зондування даних та керуючих сигналів.*

Архітектура програмного забезпечення представлена трьома частинами:

1. Кінцевий вузол. Він являє собою зондувальне обладнання, яке знаходиться в приміщенні та протягом певного часу приймає аналоговий сигнал. Arduino обробляє інтегровані сигнали та робить це за алгоритмом мінімізації ефекту шуму для підвищення точності. Після цього оброблені данні передаються через ZigBee на координатор. Процес обміну через ZigBee може проходити через кожен або декілька вузлів ZigBee. Перший вузол – головний вузол управління. Після того як одноплатний комп’ютер проаналізує отриманий сигнал, вузол програмного забезпечення визначить сценарій дій та вирішить чи запускати кінцевий пристрій або повідомити користувача.

2. Одноплатний комп'ютер. У другій частині, дані з кожного вузла інтегруються для обробки та аналізу. Отримані результати зберігаються до бази даних та стають базою знать для визначення норми показників. Якщо отриманий результат аналізу внутрішнього середовища ненормальний, то корегуючи команди надсилаються назад до кожного вузла через координатори ZigBee для відповідних дій. У цій частині вибір одноплатного комп’ютера є сутевою переваго в порівнянні з іншими приладами, бо в даному випадку забезпечує достатню економічність системи та велику швидкодію при великих об’ємах даних.

3. Кінцевий комп’ютер. У третій частині, після отримання корегуючої команди, кінцевий пристрій її обробляє та виконує безпосередня керування відповідними пристроями з метою досягнення норми показників для внутрішнього середовища.

Таким чином дана робота присвячена розробці IoT системи з можливістю інтеграції, яку можна було б застосувати для широкого спектру завдань. Та застосування IoT побудованих на використанні розумних та адаптивних алгоритмів обробки даних з різних пристроїв з метою формування комфорту та безпеки для користувача. Інтегрованість системи досягнута завдяки можливості легко розширити функціонал системи та адаптувати кожен пристрій під користувача. Для статистичного аналізу даних, їх подальшого зберігання та створення адаптивних алгоритмів створені бази даних та таблиці в них, для різних показників зовнішнього та навколишнього середовища. За рахунок підключення одноплатного комп’ютера, як головного пристрою, у системі достатньо велика швидкодія для вчасної обробки даних та управління, а також значна економічність. Взаємодія користувача з системою

відбувається за допомогою веб-сторінки на якій є можливість налаштувати систему, переглянути всю інформацію та виконати певні дії для забезпечення комфорту та безпеки.

Отже у *результаті даної роботи* були отримані:

- Розумний алгоритм управління системою, враховуючи потреби користувача;
- Бази даних та таблиці для зберігання даних;
- Веб-сторінка для керування роботою системи;
- Доступна та інтуїтивно зрозуміла інтегрована система для IoT.

Розроблена система призначена для полегшення моніторингу та управління показниками певного приміщення, забезпечення комфортних умов для користувача та підвищення безпеки за рахунок автоматичного контролю за критичними ситуаціями.

У майбутньому планується поліпшення системи за рахунок створення більш зручного інтерфейсу для користувача, покращення алгоритмів роботи та розширення функціоналу системи.

ЛІТЕРАТУРА

1. Рейтинг систем "Розумний будинок" по виробниках. Vencon: веб-сайт. URL: https://vencon.ua/ua/articles/reyting-sistem-umnyy-dom-po-proizvoditelyam (дата звернення: 19.08.2019).
2. Arduino:Бібліотеки/OneWire. Wikihandbk: веб-сайт. URL: http://wikihandbk.com/wiki/ (дата звернення: 22.08,2019)
3. G. Mois, S. Folea, T. Sanislav. Аналіз трьох бездротових датчиків на основі IoT для моніторингу навколишнього середовища. *IEEE Transactions on Instrumentation and Measurement.* 2017. URL:https://ieeexplore.ieee.org/abstract/document/7887698 (дата звернення: 08.08.2017).
4. F. Montori, L. Bedogni, L. Bononi. Інтернет спільної архітектури речей для розумних міст та моніторингу навколишнього середовища. *IEEE Internet of Things Journal*. 2017. URL: https://ieeexplore.ieee.org/abstract/document/7961139 (дата звернення: 02.04.2018).

**ЖИВАГА Владлен Володимирович** – студент, Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 6, Харків, Україна, 61022; e-mail: vladonstar@gmail.com; ORCID: 0000-0002-7117-4468.

Наукові інтереси:

– *Internet of Things системи;*
– *архітектура комп'ютерів.*

**ШЕВЧЕНКО Дмитро Олександрович** – студент, Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 6, Харків, Україна, 61022; e-mail: dimyich24@gmail.com; ORCID: 0000-0002-7897-250X.

Наукові інтереси:

– *системи штучного інтелекту;*
– *робототехніка;*
– *теорія прийняття рішень.*

**МАЛАХОВА Марина Олегівна** – к.т.н., старший викладач кафедри електроніки та управляючих систем; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 6, Харків, Україна, 61022; e-mail: maryna.malakhova88@gmail.com; ORCID: 0000-0001-5082-5279.

Наукові інтереси:

– *робототехніка;*
– *кіберфізичні системи.*

**УДК 65.0(075.8)**

ЖМЫРОВ Д.А., БЕРДНИКОВ А.Г.

# МОДЕЛИРОВАНИЕ РИСКОВ ПРИ РЕАЛИЗАЦИИ IT-ПРОЕКТОВ

**Введение**

В настоящее время требования рынка к качеству представленных товаров и услуг очень высокие, а время, которым располагают производители для выпуска на рынок новой продукции, стремится к минимуму.

При организации работы по проекту в сокращенные сроки необходимо учитывать все критичные факторы, влияющие на разработку. В связи с этим необходимо уметь работать с рисками предприятия и учитывать возможные последствия, которые они могут повлечь за собой.

Риск это не случайное явление, а конкретное событие, которое возникает на разных стадиях разработки проекта. Практически каждая IT-компания сталкивается в своей деятельности с определенными рисками, которые могут привести к непредвиденным финансовым, временным и качественным потерям. Правильная оценка рисков менеджером, организующим работу по IT-проекту, помогает минимизировать убытки и существенно снизить возможные затраты. Большинство управленческих решений принимаются в условиях неопределенности, что приводит к изменению намеченных планов и различного рода другим негативным последствиям. Этим подтверждается важность и актуальность темы данной работы.

Целью данной работы является разработка модели оценки рисков в IT-проектах, которую после апробации на практическом занятии в учебном процессе можно рекомендовать для анализа и идентификации рисков в соответствующей компании.

**Постановка задачи**

IT-компании требуется CRM-система, для их внутреннего продукта, чтобы качественно заниматься управлением взаимоотношений с клиентами.

Продукт компании – это система интернет-бронирования мест для аренды гостиниц, коттеджей, апартаментов и т.д. в разных странах мира. Разработанная CRM-система поможет автоматизировать коммуникацию с клиентами, что повлечет улучшение качества и повышение спроса на продукт.

**CRM** (customer relationship management, система управления взаимоотношениями с клиентами) — прикладное программное обеспечение, предназначенное для автоматизации взаимодействия с клиентами (заказчиками), в частности для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов. Достигается это за счет сохранения информации о клиентах, истории взаимоотношений с ними, улучшения соответствующих бизнес-процессов и последующего анализа результатов.

На этот проект выделяется определенные человеческие, временные и финансовые ресурсы, а также предъявляются определенные требования к качеству.

Перед руководством фирмы стоит вопрос, разработать CRM-систему силами своих разработчиков или использовать услуги сторонней организации.

Для принятия оптимального решения необходимо классифицировать риски, количественно и качественно оценить возможные последствия для компании при обоих вариантах, сформулировать предложения руководству по мониторингу ситуации и проведению мероприятий по управлению рисками.

**Классификация рисков в IT-проектах**

Неопределенность при принятии решений по организации работы над проектом характеризуется следующими аспектами: неполным знанием всех обстоятельств ситуации; невозможностью адекватного и точного учета всей информации; факторами, которые априори невозможно предусмотреть и спрогнозировать. Названные условия связаны между собой и

приводят к потенциальным рискам, которые являются источниками неблагоприятных ситуаций и связанных ними последствий в виде потерь и убытков.

В работе сделана классификация рисков IT-компании, которая позволила систематизировать финансовые и временные риски, а также риски снижения качества продукции. Данная классификация представлена на схеме как дерево рисков, образованное группами рисков различной природы.

Дерево рисков представляет собой иерархическую структуру, соответствующую следующим иерархическим моделям: разбиения работ (Work Breakdown Structure – WBS), организационной структуры управления проектом (Organization Breakdown Structure - OBS), структуры разбиения стоимости проекта (Cost Breakdown Structure – CBS), структуры ресурсов проекта (Resource Breakdown Structure – RBS). Оно может быть использовано для проведения качественного и количественного анализа рисков и обеспечивает осуществление процесса систематической идентификации рисков в зависимости от уровня детализации.

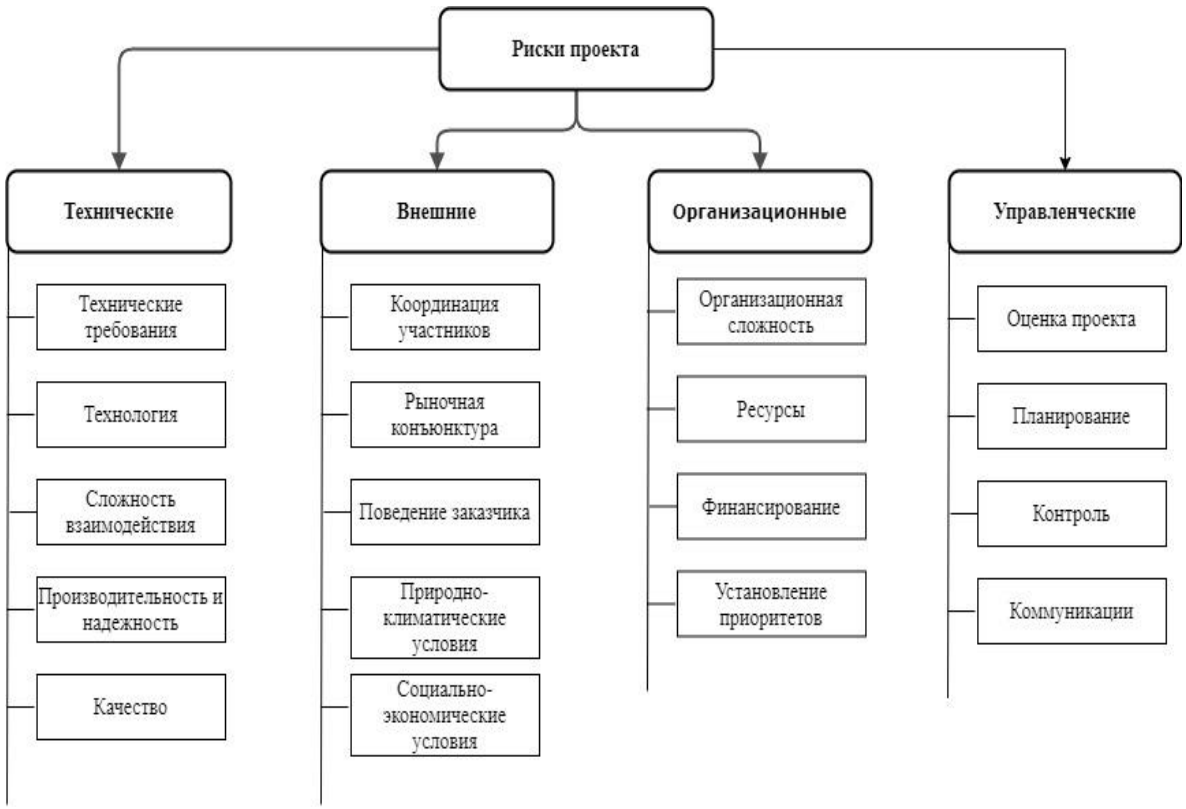



*Рис. 1 Дерево Рисков*

Принцип оценки рисков предполагает учет взаимосвязи количественных и качественных показателей.

Качественный анализ рисков позволяет менеджеру выявить причину появления риска и определить, на каких стадиях есть угроза его возникновения. Таким образом, устанавливаются возможные области риска и вычисляются возможные убытки при его возникновении.

При количественном анализе рисков проводится оценка рисков, которые фигурируют в соответствующей операции алгоритма принятия решения.

Менеджер, принимающий решение, получает числовые значения параметров по каждому риску и по объекту в целом, вычисляет потенциальный урон, вырабатывает его стоимостную оценку и определяет действия для нейтрализации последствий с финансовым расчетом.

Для эффективного выполнения этой задачи менеджеру необходим соответствующий инструментарий, которым и является разработанная модель оценки рисков.

**Суть разработанной модели оценки рисков IT-проекта**

Для оценки финансовых и временных рисков проекта разработана модель, которая включает в себя комплексное использование метода дерева решений и метода PERT (Project Evaluation and Review Technique), что по нашему мнению делает оценку более точной.

Дерево решений представляет собой модель, которая позволяет разбить сложную проблему принятия решений в условиях риска на совокупность меньших проблем, каждая из которых может быть рассмотрена отдельно, а затем в совокупности. Применение этого метода предполагает, что временные параметры процесса являются детерминированными.

Однако на практике временные показатели проекта могут характеризоваться случайными параметрами, которые подчиняются собственному закону распределения со своими числовыми характеристиками, а неопределенность сроков выполнения отдельных операций означает, что общая продолжительность работ также подвержена неопределенности. Поэтому с целью повышения точности определения временных рисков предлагается использовать метод PERT, который позволяет оценивать временные параметры проекта в условиях случайного характера продолжительности работ.

Качество и эффективность разработанной модели предлагается оценить с помощью диаграммы Парето.

Таким образом, предлагаемая модель оценки рисков при реализации IT-проектов позволяет проводить более точный мониторинг хода работ по проекту.

ЛІТЕРАТУРА

1. Управление проектом. Основы проектного управления: учебник / коллектив авторов под редакцией М.Л. Разу. – М.: изд КНОРУС, 2006 – 768.
2. Теория принятия решений: учебник / Черноморов Г. А; изд: Ред. Журнал «Изв. вузов, Электромеханика», 2002 - 276.

**ЖМЫРОВ Данил Андреевич –** студент группы КИ-41 факультету компьютерных наук; Харьковский национальный университет имени В.Н. Каразина, майдан Свободы, 4, Харків-22, Украина, 61022; e-mail: reus11fan@gmail.com; ORCID: 0000-0003-3661-2642.

Научные интересы:

- *Программирование*
- *Управление проектами.*
- *Алгоритмы и структуры данных.*

**БЕРДНИКОВ Анатолий Георгиевич** – к. т. н., доцент, доцент кафедры теоретической и прикладной системотехники; факультету компьютерных наук; Харьковский национальный университет имени В.Н. Каразина, майдан Свободы, 4, Харків-22, Украина, 61022; e-mail: tps@karazin.ua.

Научные интересы:

- *Программирование.*
- *Моделирование.*
- *Проектирование.*

**УДК 004.7**

ЗЕЛЕНСЬКА Н.В.

# АНАЛІЗ ЗАСОБІВ МОНІТОРИНГУ КОМП’ЮТЕРНОЇ МЕРЕЖІ

**Вступ**

На сьогоднішній день комп'ютерна мережа є невід'ємною частиною нашого життя. У повсякденні – це зв'язок з усім світом. З розвитком світових технологій та їх впровадженням в промислових сферах ускладнюються і їх принцип роботи, збільшується кількість показників, за якими потрібно слідкувати, а це в свою чергу призводить до збільшення навантажень на комп’ютерні мережі. Тож зараз вкрай необхідні фізичні засоби для вдосконалення систем, а також засоби, які допоможуть нам ефективно слідкувати за змінами стану мережі та її елементів, задля підтримання надійного, безперебійного зв’язку та швидкого усунення несправностей.

Рішення цього питання для пристроїв першого покоління не забезпечують нормального контролю, так як частіше за все не враховується динамічний характер мережевих топологій. Зростання рівня складності та адаптація атак до вжитим заходам захисту змушують організації встановлювати суворі правила безпеки, щоб отримувати повідомлення про будь-які потенційні порушення ще до їх виникнення. Засоби аналізу поведінки мережі або моніторинг мережевого трафіку стають обов’язковими складовими будь-якої мережі і разом з системами виявлення аномалій гарантують повну мережеву безпеку.

Вибудовуючи стратегію безпеки, фахівці рекомендують IT-службам поряд із засобами захисту встановлювати в своїх мережах також і системи аналізу поведінки часто можуть виявити загрози, що вислизають від уваги інших засобів. Вони зміцнюють безпеку мережі за рахунок моніторингу трафіку і відділення незвичайних дій, подій і тенденцій від нормальних операцій в мережевому трафіку.

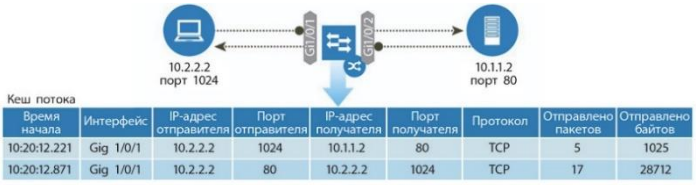
**Методи збору та моніторингу даних мережі**

Існують різні механізми збору інформації про мережевий трафік. Більшість необхідних компонентів вже присутні в інфраструктурі сучасної мережі. Для виявлення і відображення мережевої атаки організації можуть використовувати одну або кілька технологій для моніторингу додатків і / або моніторингу даних мережевих потоків, зібраних в мережі:

- захоплення мережевих пакетів;
- вибірковий експорт потоків (захоплення пакетів);
- повний експорт потоків.

Моніторинг комп'ютерних мереж в поєднанні з використанням орієнтованих на безпеку колекторів NetFlow сприяє швидкому або завчасному виявленню незвичайного і аномального поведінки. Повна реалізація NetFlow дозволяє фіксувати всю мережеву активність на інтерфейсі, через який проходить IP-потік, і корисна для кореляції подій і аналізу даних. Важливою властивістю є збір даних з усього потоку. Це допомагає зменшити число помилкових спрацьовувань (навідміну від вибіркового захоплення пакетів). Крім того, завдяки можливості переглядати весь потік даних вдається виявляти випадки повільного сканування мережі та атомарних атак, нерозпізнаних традиційними рішеннями безпеки.

NetFlow - це мережевий протокол, розроблений компанією Cisco для збору і моніторингу даних про рух мережевого трафіку, що генеруються маршрутизаторами і комутаторами з підтримкою NetFlow. Він створювався як технологія пакетної комутації для маршрутизаторів Cisco. Покладена в основу NetFlow ідея (рис. 1) полягає в тому, що перший пакет потоку генерує на комутаторі або маршрутизаторі запис комутації NetFlow. Надалі цей запис використовується при обробці пакетів з того ж потоку аж до його закінчення. Пошук конкретного маршруту в таблиці маршрутизації потрібно тільки для першого пакету потоку.




Кеш потока

| Время начала | Интерфейс | IP-адрес отправителя | Порт отправителя | IP-адрес получателя | Порт получателя | Протокол | Отправлено пакетов | Отправлено байтов |
|---|---|---|---|---|---|---|---|---|
| 10:20:12.221 | Gig 1/0/1 | 10.2.2.2 | 1024 | 10.1.1.2 | 80 | TCP | 5 | 1025 |
| 10:20:12.871 | Gig 1/0/1 | 10.2.2.2 | 80 | 10.2.2.2 | 1024 | TCP | 17 | 28712 |

*Рис. 1 – Приклад отриманої за допомогою NetFlow інформації*

Представлена на рис. 1 інформація про потік може бути експортована. Сьогодні вона використовується для аналізу продуктивності мережі і поведінки, які проводяться з метою безпеки. Потоки не містять реальних даних пакета, в них присутні тільки метадані про з'єднання. Це стандартна форма даних сеансу, в яких детально розписано все, що стосується мережевого трафіку: хто, що, коли і де (рис. 2). Кожна мережева транзакція зазвичай розбивається на два потоки – по одному в кожному напрямку.

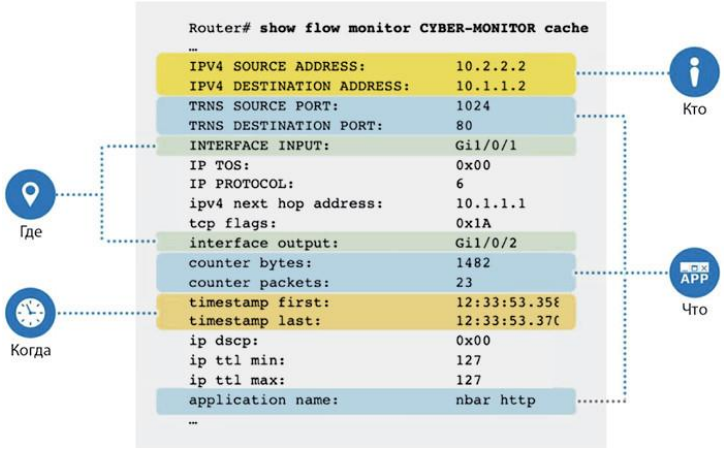



*Рис. 2 - Один запис NetFlow містить значний обсяг інформації*

**Переваги NetFlow**

NetFlow і sFlow мають принципові відмінності. У NetFlow облік трафіку, що проходить через мережу, може виконуватися як для всіх транзакцій, так і для вибіркового їх числа. sFlow є «виключно вибіркову» технологію, в якій комутатор або маршрутизатор випадковим чином відбирає зразки фіксованої довжини для одного з N пакетів. Хоча sFlow може бути корисний при плануванні пропускної здатності і забезпеченні видимості трафіку і додатків, як цілісного рішення безпеки він не підходить. І ось чому:

- потенційно високий рівень помилкових спрацьовувань.
- неможливість виявлення повільного сканування мережі.
- атомарні атаки.

Тому навіть цієї досить не значної, на перший погляд, різниці досить для того, щоб зробити вибір на користь NetFlow.

**Врахування навантаження при побудові комп'ютерної мережі та моделі управління даними**

Навантаження каналу комп'ютерної мережі є однією з найважливіших характеристик, які необхідно враховувати під час побудови системи моніторингу комп'ютерної мережі, так як це один с з перших ознак того, шо з мережею щось відбувається. Тож прогнозування роботи каналів або комп'ютерної мережі в цілому, визначення небезпечних наближень навантаження мережі до її граничних значень та визначення пропускної здатності каналів є необхідними для можливості прийняття рішень щодо модернізації обладнання комп'ютерної мережі або можливості програмного продукту змінювати її конфігурації. Такі характеристики як середні значення та діапазон мінливості навантаження каналу, варіабельність середніх навантажень за великий проміжок часу дозволяють визначити основні ймовірності, характеристику змін в навантаженні каналів, робити висновки про стабільність роботи мережі та відстежувати тренди поведінки

мережі в часі. Ці характеристики дозволяють зібрати статистичні дані про навантаження мережі. Ця інформація є основною в процесі визначення обміну інформації по каналам зв'язку за умови нормальної роботи мережі та за допомогою математичної моделі з трендом та сезонною складовою, яка описується формулою:

$$Y(t) = f(t) + q(t) + \psi(t) \quad (1)$$

де – $f(t)$ тренд; функція, що повільно змінюється у часі, описує зміни середніх добових (тижневих, інших) навантажень за інтервали часу; $q(t)$ – сезонна складова; $\psi(t)$ – випадкова послідовність, відносно якої робиться припущення про рівність нулю її математичного очікування $M[\psi(t)] = 0$ та дисперсією $\sigma^2 = \sigma^2(t)$.

$q(t)$ і $f(t)$ можуть бути оцінені за значеннями ряду, отриманих за допомогою процедури його декомпозиції на складові (Season Trend Decomposition). Для перевірки наявності та визначення сезонної складової може використовуватися аналіз періодограми (спектральної щільності).

Тренд моделюється на основі на основі параметричних моделей за допомогою методів регресійного аналізу. Методи аналізу періодограм та спектрального аналізу випадкових процесів слід використовувати для побудови ряду Фурьє.

Властивості та характеристики випадкової послідовності $\psi(t)$ вивчаються за допомогою класичних методів математичної статистики та методів аналізу випадкових послідовностей.

Випадкова послідовність $\psi(t)$в моделі (1) аналізується тільки після «очищення» даних від детермінованих складових – тренду $f(t)$ та періодичної функції сезонної складової $q(t)$.

Відмінною особливістю часової послідовності навантаження каналу $Y(t)$ є складна форма періодичного сигналу (форма сигналів ближча до трапецевидної), а отже ряд є нестаціонарним та неоднорідним.

**Висновки**

У роботі були розглянуті декілька основних моментів, що розглядаються та використовуються під час побудови програмної моделі моніторингу та управління мережевими даними. Однак чимало проблем може виникнути на момент реалізації програмного коду. Наприклад, налаштування обладнання для перевірки працездатності продукту, так як деяке обладнання доволі погано сприймає неофіційне програмне забезпечення, нестача пам'яті на жорсткому диску для збереження статистичних даних та логів, тощо.

ЛІТЕРАТУРА

1. Трухан А.В. Оптимизация сетей телекоммуникаций на основе требований к качеству обслуживания. – Минск: БГУИР, 2011.
2. Олифер В., Олифер Н. Компьютерные сети. Принципы, технологии, протоколы : учеб. для вузов. – СПб., 2016. – 992 с.
3. Лосев Ю. І., Руккас К. М., Шматков С. І. Комп'ютерні мережі: навч. посіб. / за редакцією Ю. І. Лосева. – Х. : ХНУ імені В. Н. Каразіна, 2013. – 248 с.
4. Уэнделл Одом. Официальное руководство Cisco по подготовке к сертификационным экзаменам CCNA ICND2 200-101: маршрутизация и коммутация, акад. изд. – 2015. – 736 с.
5. Гостев В.М., Хабибуллин Р.Ф. Технологии оптимизации проектирования сетей передачи данных территориальных компьютерных сетей. Исслед. по информ., 1999, выпуск 1, 157–174 с.

**ЗЕЛЕНСЬКА Ніка Вікторівна** – бакалавр, студентка кафедри теоретичної та прикладної системотехніки, Харківський національний університет імені В. Н. Каразіна, майдан Свободи 6, Україна, 61022; e-mail: zelenskaya.nika@gmail.com; ORCID: 0000-0002-9289-5061.

Наукові інтереси:

*– комп'ютерні мережі;*

*– методи моніторингу даних.*

**УДК 519.711**

ЗЕМЦОВА І.Р.

# КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ТА ВИЗНАЧЕННЯ ЗАКОНУ РОЗСІЮВАННЯ МУЛЬТИАГЕНТНОЇ СИСТЕМИ НА ПЕРЕШКОДІ

Мультиагентні системи реального часу є ефективним інструментом для моделювання складних процесів, в яких бере участь велика кількість активних автономних сутностей. До таких процесів відносяться потоки міського руху, логістичні системи, соціальні явища, епідемії. Методи мультиагентного моделювання також використовуються для пошуку і обробки інформації в інформаційних мережах, системах управління автономними роботами. Перспективним напрямком розвитку мультиагентних систем є розробка безпілотних автомобілів і літальних апаратів. Одним із напрямків, який активно розвивається є використання мультиагентного підходу в системах віртуальної реальності та відеоіграх. У цих системах агентне моделювання застосовується для збору статистики, розрахунку обчислювального навантаження серверів і реалізації ігрового штучного інтелекту [1, 2].

Мультиагентна система - це система, створена декількома інтелектуальними агентами, які взаємодіють між собою. Багатоагентні (мультиагентні) системі можуть бути використані для вирішення таких проблем, які важко, чи взагалі неможливо, розв'язати за допомогою одного агента чи монолітної системи [1].

Основна мета роботи полягає у визначенні закону розсіювання мультиагентної системи на перешкоді та розробці програмного забезпечення, за допомогою якого можна створити комп'ютерну модель поведінки мультиагентної системи при певних умовах.

Головний алгоритм, який полягає в основі вивчення закону розсіювання - це алгоритм Флокінга. Флокування - це форма колективної поведінки великої кількості взаємодіючих агентів із загальною груповою метою. Приклади цих агентів включають птахів, риб, пінгвінів, мурах, бджіл та натовпу. Флокування є прикладом узгодженої задачі, яка виконується динамічними агентами над (візуально невидимими) самоорганізючими мережами в природі. Самоорганізована особливість скупчень може забезпечити більш глибоке розуміння проектування сенсорних мереж. Флокування на основі частинок також є одним з елементів технології тривимірної анімації, яка зробила революцію в індустрії [3, 4].

Флокування вважається емерджентною - або такою, що виявляється несподівано, - поведінкою, що випливає із простих правил, яким слідують особи, і яка не включає в себе ніякої центральної координації [5]. Основні моделі поведінки флокування контролюються трьома простими правилами [6, 7]:

1. відокремлення
2. вирівнювання
3. згуртованість

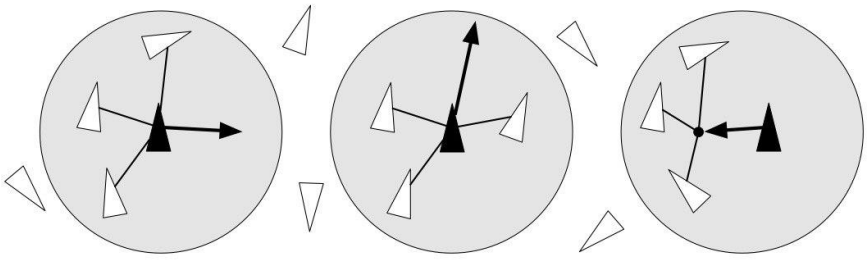


*Рис.1 Візуалізація правил поведінки*

Об'єктом вивчення та спостереження є 3D модель мультиагентної системи (зграї), яка має наступні характеристики: швидкість та напрямок руху, кількість елементів (птахів) у зграї.

Модель мультиагентної системи, має дотримуватися вищезазначених трьох правил Рейнольдса та виконувати наступні дії: спочатку обчислюється поточний стан агентів, далі

візуалізуємо стан та повторюємо попередні кроки. На рис.2 показана структура реалізації алгоритму. Перший крок для внутрішніх циклів - це обчислення. Три правила алгоритму змінюють лише швидкість та напрямок агента, вони можуть застосовуватися в будь-якому порядку. Після того як швидкість та напрямок агентів було оновлено трьома правилами, можемо оновлювати й їх позиції. Оновлення позицій відбувається за допомогою простого обчислення, де буде агент, після того як він пролетить прямо вперед невеликий проміжок часу (вже з новою швидкистю та курсом). Після того, як дані усіх агентів будуть оновлені, візуалізуємо поточний стан.

```
Data: A group of boids.
Result: Simulates flocking behaviour with an animation.
foreach Frame do
    foreach boid do
        separation(boid);
        cohesion(boid);
        alignment(boid);
    end
    foreach boid do
        boid.x ← cos(boid.course) * b.velocity * dTime;
        boid.y ← sin(boid.course) * b.velocity * dTime;
        draw(boid);
    end
end
```

*Рис. 2 Структура реалізації алгоритму*

Вирівнювання - це поведінка, яка дозволяє окремому агенту рухатись в одну й туж сторону, як і інші агенти. Кожен агент намагається наслідувати один одному напрямок і швидкість. Структура алгоритму реалізації цього правила (рис. 3):

```
Data: A boid.
Result: The course and velocity of the boid is updated.
dCourse ← 0;
dVelocity ← 0;
neighbours ← getNeighbours(boid);
foreach nBoid in neighbours do
    dCourse ← dCourse + getCourse(nBoid) - getCourse(boid);
    dVelocity ← dVelocity + getVelocity(nBoid) - getVelocity(boid);
end
dCourse ← dCourse / neighbours.size();
dVelocity ← dVelocity / neighbours.size();
boid.addCourse(dCourse);
boid.addVelocity(dVelocity);
```

*Рис. 3 Структура алгоритму реалізації першого правила*

Згуртованість - це поведінка, яка змушує агентів спрямовуватись до центру маси (до середнього положення у певному радіусі). Необхідно знайти середнє положення агентів та перемістити їх до цієї області. Алгоритм реалізації вказаного правила (рис. 4):

```
Data: A boid.
Result: The course of the boid is updated.
goal ← (0,0);
neighbours ← getNeighbours(boid);
foreach nBoid in neighbours do
    goal ← goal + positionOf(nBoid);
end
goal ← goal / neighbours.size();
steerThoward(goal, boid);
```

*Рис.4 Структура алгоритму реалізації другого правила*

Відокремлення - це поведінка, яка дозволяє повертати в сторону, щоб запобігти зіткнення з сусідами [8, 9]. Агенти намагаються відхилитися один від одного, з метою уникнення зіткнення. Відстань, з якої вони починають уникати один одного, має бути меншою, ніж відстань, з якою вони притягуються (дотримуючись правила згуртованості). Структура алгоритму даного правила (рис. 5):

```
Data: A boid.
Result: The course of the boid is updated.
goal ← (0,0);
neighbours ← getNeighbours(boid);
foreach nBoid in neighbours do
    goal ← goal + positionOf(boid) - positionOf(nBoid);
end
goal ← goal / neighbours.size();
steerThoward(goal, boid);
```

*Рис. 5 Структура алгоритму реалізації третього правила*

Зграя взаємодіє з перешкодою - ще один необхідний об'єкт дослідження. Вона також представлена у вигляді 3D модель циліндра.

Характер зіткнення визначається двома параметрами: швидкість руху зграї (v) (її напрямок та значення) та прицільний параметр($\rho$) (рис. 6). Прицільний параметр показує відстань між центром зграї та паралельною лінією до напрямку руху зграї, яка проходить через центр перешкоди.

Початкове положення центрів (зграї та перешкоди) та напрямок руху мультиагентної системи задаються. Змінюючи центр ваги мультиагентної системи та перешкоди а також напрямок руху зграї, змінюється значення прицільного параметра.

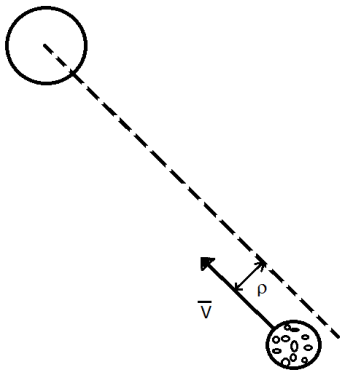



*Рис. 6 Схема розміщення та взаємодії зграї і перешкоди*

Після зіткнення з перешкодою змінюється швидкість руху зграї, кількість елементів та напрямок їх руху. Напрямок руху елементів мультиагентної системи відхиляється на певний кут, в залежності від того, якого характеру було зіткнення.

Досліджуючи розсіювання мультиагентної системи на перешкоді, ми розглядаємо всі елементи зграї як окремі частинки та обчислюємо значення характеристик кожної із них. Внаслідок чого ми формуємо матрицю розсіювання мультиагентної системи. Аналогічним чином відбувається дослідження закону розсіювання частинок у фізиці. Адже розсіювання частинок - це зміна напрямку руху частинок в результаті зіткнення з іншими частинками. В ході обчислень формується матриця розсіювання, яка характеризує систему, в якій проводяться спостереження [10, 11,12].

Отже, характеристики розсіювання мультиагентної системи подаються у вигляді матриці розсіювання зграї. Після кожної взаємодії зграї з перешкодою знаходимо матрицю розсіювання. Змінюючи прицільний параметр, та швидкість руху та кількість агентів визначаємо залежність

матриці розсіювання від цих параметрів. Додаткову специфіку розсіювання зграї визначає зміна кількості агентів у зграї після розсіювання.

За допомогою програмної реалізації, було проведено значну кількість експериментів. В ході виконання ряду експериментів встановлено закономірності розсіювання мультиагентної системи на перешкоді. Додатково виявлено залежність кількості агентів в зграї після розсіювання на перешкоді у залежності від прицільного параметру та швидкості зграї.

ЛІТЕРАТУРА

1. Multi-agent system: веб-сайт. URL: https://en.wikipedia.org/wiki/Multi-agent_system (дата звернення 05.12.2019).
2. Carl-Oscar Erneholm Simulation of the Flocking Behavior of Birds with the Boids Algorithm: Bachelor of Science Thesis Stockholm, Sweden 2011.
3. Reza Olfati-Saber Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory: Technical Report CIT-CDS 2004-005, June 22, 2004. *Submitted to the IEEE Transactions on Automatic Control.*
4. Flocking (behaviour): веб-сайт. URL: https://en.wikipedia.org/wiki/Flocking_(behavior) (дата звернення 25.12.2019).
5. Флокування (поведінка) веб-сайт. URL: https://uk.wikipedia.org/wiki/Флокування_(поведінка) (дата звернення 20.12.2019).
6. 3 Simlpe Rules of Flocking Behaviour: Alignment, Cohesion and Separation: веб-сайт. URL: *https://gamedevelopment.tutsplus.com/tutorials/3-simple-rules-of-flocking-behaviors-alignment-cohesion-and-separation--gamedev-3444* (дата звернення: 11. 02. 2020).
7. Boids (Flocks, Herds, and Schools: a Distributed Behaviour Model): веб-сайт. URL: https://www.red3d.com/cwr/boids/ (дата звернення 06.02.2020).
8. Daniel Sinkovits Flocking Behavior: May 5, 2006
9. Bird Flocking Behavior Algotithms: веб-сайт. URL: https://www.lalena.com/AI/Flock/ (дата звернення 28.02.2020).
10. Рассеяние частиц: веб-сайт. URL: https://ru.wikipedia.org/wiki/Рассеяние_частиц (дата звернення 01.03.2020).
11. Эффективное сечение: веб-сайт. URL: https://ru.wikipedia.org/wiki/Эффективное_сечение (дата звернення 20.03.2020).
12. Flocking: веб-сайт. URL: https://www.processing.org/examples/flocking.html (дата звернення 06.02.2020).

**ЗЕМЦОВА Інна Романівна** - студентка факультету комп'ютерних наук, кафедри штучного інтелекту Харківського національного університету імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: zemtsova22inna@gmail.com; ORCID:0000-0002-9017-4092.

Наукові інтереси:

*– дослідження та вивчення штучного інтелекту, мультиагентних систем.*

**УДК 003.26**

КАПТЬОЛ Є.Ю., ГОРБЕНКО І.Д.

# АНАЛІЗ МОЖЛИВОСТЕЙ ПРОГРАМУВАННЯ ЗАДАЧ КРИПТОЛОГІЇ НА КВАНТОВОМУ КОМП'ЮТЕРІ

**Вступ**

Питання програмування задач криптології на квантовому комп'ютері стає все більш актуальним. Особливої актуальності воно набуло через успіхи у справі створення робочого екземпляру квантового комп'ютера.

Необхідність вирішення цієї задачі базується на тому, що квантові комп'ютери мають значну перевагу над класичними у швидкодії. Ця перевага здебільшого базується на використанні квантовими комп'ютерами квантових властивостей, що є недоступними для класичних комп'ютерів. Так, наприклад, завдяки квантовим властивостям, ми маємо можливість розглядати одразу весь регістр, замість одного елемента, як в класичному комп'ютері, що дає суттєву перевагу в швидкодії.

**Квантовий комп'ютер**

На основі квантових властивостей (властивостей квантової фізики) розроблено особливу квантову математику, що покладено в основу квантових алгоритмів, таких як: алгоритми Шора для факторизації, для вирішення дискретного логарифму в скінченному полі та в групі точок ЕК, алгоритм Гровера та алгоритм для криптоаналізу перетворень в фактор-кільці. Вони забезпечують вирішення задач, що є або неможливими для вирішення за допомогою класичних комп'ютерів, або нерентабельно складними з точки зору часу, необхідного для отримання результату. Однією з важливих особливостей квантових алгоритмів є те, що вони мають ймовірнісну природу, тобто результат буде отримано з певною ймовірністю. Таким чином, задачею алгоритмів є підвищення ймовірності отримання правильного результату при вимірюванні стану квантового регістру після завершення роботи алгоритму.

Застосування ж квантових алгоритмів на класичному комп'ютері не тільки не дає переваги, а є дуже невигідним з точки зору швидкодії. Так, наприклад, реалізація методу Гровера на класичному комп'ютері є невигідною через те, що сам алгоритм передбачає багаторазове повторення ітерації Гровера для підвищення ймовірності отримання потрібного результату при вимірюванні стану квантового регістру. Тоді як на квантовому комп'ютері повтори покращують результат, а квантові властивості нівелюють затрати на повтори, на класичному комп'ютері ці повтори не є потрібними, а лише зайвими. Щодо властивостей, що нівелюють для квантового комп'ютера проведення повторів, то, наприклад, в той час, як квантовий комп'ютер має змогу переглядати весь регістр однією операцією, в той час, як на класичному комп'ютері регістр переглядається по одному елементу за операцію.

Завдяки сучасним успіхам в створенні квантового комп'ютера існує можливість доступу до програмування квантового комп'ютера за допомогою хмарних сервісів.

Наразі для загального доступу з використанням хмарних сервісів доступними є квантові комп'ютери компанії IBM на 1, 5 та 15 кубітів. Вони виконують квантові властивості та можуть реалізовувати квантові гейти. Також можна скористатися квантовим симулятором (до 32 кубітів).

Також слід помітити, що доступні для використання квантові гейти однокубітні та двохкубітні, та один з них трикубітний. Робоча реалізація методу Гровера в свою чергу потребує наявності квантових гейтів, що діяли б на більшу кількість кубітів одночасно. Тому багатокубітні гейти доводиться розкладати на послідовності гейтів, що є в наявності(універсальних).

Серед доступних для загального доступу квантових комп'ютерів наявні: ibmq_16_melbourne (15 кубітів), ibmq_london (5 кубітів), ibmq_burlington (5 кубітів), ibmq_essex (5 кубітів), ibmq_ourense (5 кубітів), ibmq_vigo (5 кубітів), ibmq_5_yorktown-ibmqx2 (5 кубітів),

ibmq_armonk (1 кубіт). Квантовий симулятор ibmq_qasm_simulator може розраховувати схеми, що передбачають використання до 32 кубітів.

**Метод Гровера**

Одним з основних квантових методів, що є необхідними для вирішення задач криптології є метод Гровера. Алгоритм Гровера є квантовим алгоритмом, що призначений для проведення вичерпного пошуку унікального елементу в несортованій базі даних, що містить $N = 2^n$ елементів, де $n$ позначає довжину задіяного для представлення пошукового простору квантового регістру (кількість кубітів в ньому), а $N$ є розміром пошукового простору[1].

Особливість його полягає в тому, що, завдяки квантовим властивостям та використанню функції «чорної скриньки» (у вигляді квантового оракула), він потребує лише $O(\sqrt{N})$ групових операцій замість $O(N)$ у класичних алгоритмів. Квадратичне прискорення у порівнянні з класичними алгоритмами досягається завдяки використанню квантових властивостей, таких як квантова суперпозиція станів.

Хоча інші квантові алгоритми при порівнянні з класичними аналогами можуть забезпечити експоненційне прискорення, а алгоритм Гровера може забезпечити лише квадратичне прискорення, слід зауважити, що навіть таке прискорення є дуже значним та його значущість збільшується зі зростанням $N$. Для прикладу, методом Гровера 128-бітний криптографічний ключ можна зламати приблизно за $2^{64}$ звернень до функції «чорної скриньки», що можна визначити як $2^{64}$ звернень до ітерації Гровера, а отже $2^{64}$ ітерацій методу, в той час як 256-бітний криптографічний ключ можна зламати за, приблизно, $2^{128}$ ітерацій. Виходячи саме з цього твердження для збільшення стійкості проти квантових атак іноді пропонують збільшувати довжину криптографічних ключів в 2 рази [2].

Метод Гровера, як і більшість квантових методів, є ймовірнісним, тобто правильна відповідь може бути виміряна з квантового регістру з певною ймовірністю, що не перевищує 1. Також слід зауважити, що існує можливість значно знизити ймовірність виміру правильного результату виконавши на 1 ітерацію більше, ніж необхідно [3].

Метод Гровера має безліч можливостей для застосування, одним з котрих є реалізація його як алгоритму криптоаналізу симетричних перетворень, функцій гешування, асиметричного шифру в кільці поліномів та ін. у зв’язку з його узагальненим змістом [1,2]. Для криптоаналізу симетричних блокових перетворень метод можна звести до алгоритму пошуку сеансового чи довгострокового ключа тощо. У випадку функцій гешування метод можна застосувати для пошуку колізій тощо. Метод Гровера має значні потенційні можливості, котрі беруться в розрахунок з огляду на сучасний стан розробки квантового комп’ютера.

Для розуміння методу Гровера необхідно визначити квантову суперпозицію станів. Нехай $|\psi\rangle$ - суперпозиція всіх станів (згідно нотації Дірака):

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle. \qquad (1)$$

З урахуванням цього, алгоритм приймає такий вигляд:

1) Встановлення системи в стан суперпозиції $|\psi\rangle$.
2) Виконання «ітерації Гровера» (або ж $G$) $\frac{\pi}{4}\sqrt{N}$ разів, де $N = 2^n$ та $N$ становить розмір пошукового простору, а $n$ - розмір квантового регістру, що використовується для представлення пошукового простору. При цьому $G$ включає в себе 2 етапи:
   2.1) Застосування квантового оракула ($O$).
   2.2) Застосування оператора дифузії, що здійснює «інверсію щодо середнього» та має вигляд $2|0\rangle\langle 0| - I$.
3) Виконання класичних вимірювань регістру для отримання результату роботи алгоритму, що з ймовірністю близькою до 1 буде вірним.

**Метод Гровера на квантовому комп'ютері**

Враховуючи все наведене вище було вироблено три схеми застосування методу Гровера на квантовому комп'ютері для квантового регістру з чотирьох кубітів для отримання результату «0», що може бути представлено рядком бітів $|0000\rangle$, та три схеми для отримання результату 4, що може бути представлено рядком бітів $|0100\rangle$. Схеми відрізняються кількістю застосування ітерацій Гровера. Так в перших схемах застосовується одна ітерація Гровера, в других – дві ітерації, в третіх – три ітерації. Ці схеми було випробувано на доступних для загального доступу тестових квантових комп'ютерах компанії IBM та доступному через той же сервіс квантовому симуляторі. Серед квантових комп'ютерів, використаних в досліджені, були: $ibmq\_16\_melbourne$, $ibmqx2$ (він же $ibmq\_5\_yorktown - ibmqx2$), $ibmq\_burlington$. Також було використано квантовий симулятор $ibmq\_qasm\_simulator$.

Як вже було зазначено, особливості реалізації метода Гровера для квантового регістру, що складається з чотирьох кубітів на квантовому комп'ютері включають в себе необхідність застосування 3-кубітних квантових гейтів. В той же час інструментарій для взаємодії з доступними квантовими комп'ютерами не включає в себе квантові гейти, що оперують над потрібною кількістю кубітів. Потрібні квантові гейти можна замінити сукупностями наявних в інструментарії квантових гейтів, що дають той самий ефект.

Результати виконаних випробувань можуть вказувати як на недосконаліть методів, використаних для представлення багатокубітних гейтів у вигляді сукупності одно- та двокубітних гейтів, так і на недосконаліть розроблених квантових комп'ютерів, що підлягає перевірці в майбутніх дослідженнях.

Результати застосування методу Гровера з однією ітерацією для отримання $|0000\rangle$ показали значні розбіжності між реальними результатами та отриманими за допомогою квантового симулятора. Так, симулятор вказував на те, що ймовірність отримання $|0000\rangle$ становить 46,875%, в той час як при проведенні реальних вимірювань цей результат було отримано лише 6,152% разів на $ibmq\_burlington$, 7,52% - на $ibmqx2$ та 8,789% - на $ibmq\_16\_melbourne$ квантових комп'ютерах.

Результати застосування методу Гровера з двома ітераціями для отримання $|0000\rangle$ показали ще більші розбіжності між очікуваними та реальними результатами. Так, на симуляторі ймовірність отримання $|0000\rangle$ становила 91,211%, в той час як насправді було отримано лише 6,543% на $ibmq\_burlington$, 8,398% на $ibmqx2$ та 10,156% на $ibmq\_16\_melbourne$ квантових комп'ютерах.

Результати застосування методу Гровера з трьома ітераціями (що є остаточною кількістю ітерацій для регістру цього розміру) для отримання $|0000\rangle$ підтвердили розбіжності, отримані на попередньому кроці. Так, Симулятор спрогнозував ймовірність отримання $|0000\rangle$ в 96,387%, в той час як на реальних квантових комп'ютерах було отримано потрібний результат значно меншу кількість разів: 7,227% - на $ibmq\_burlington$, 8,301% - на $ibmqx2$, 9,961% на $ibmq\_16\_melbourne$ квантовому комп'ютері.

Результати застосування методу Гровера з однією ітерацією для отримання $|0100\rangle$ показали значні розбіжності між реальними результатами та отриманими за допомогою квантового симулятора. Так, симулятор вказував на те, що ймовірність отримання $|0100\rangle$ становить 46,387%, в той час як при проведенні реальних вимірювань цей результат було отримано лише 5,469% разів на $ibmq\_burlington$, 10,547% - на $ibmqx2$ та 6,506% - на $ibmq\_16\_melbourne$ квантових комп'ютерах.

Результати застосування методу Гровера з двома ітераціями для отримання $|0100\rangle$ показали ще більші розбіжності між очікуваними та реальними результатами. Так, на

симуляторі ймовірність отримання $|0100\rangle$ становила 90,259%, в той час як насправді було отримано лише 6,006% на $ibmq\_burlington$, 6,982% на $ibmqx2$ та 6,897% на $ibmq\_16\_melbourne$ квантових комп'ютерах.

Результати застосування методу Гровера з трьома ітераціями для отримання $|0100\rangle$ підтвердили розбіжності, отримані на попередньому кроці. Так, Симулятор спрогнозував ймовірність отримання $|0100\rangle$ в 96,436%, в той час як на реальних квантових комп'ютерах було отримано потрібний результат значно меншу кількість разів: 6,299% - на Бурлінгтонському, 6,152% - на Йорктаунському, 7,837% на Мельбурнському квантовому комп'ютері.

З отриманих результатів можна зробити висновок, що нинішні квантові комп'ютери ще не є здатними на повноцінне контрольоване відтворення всіх квантових властивостей. Хоча можливо з більшою кількістю кубітів метод Гровера і даватиме кращий результат, що буде перевірено подальшими дослідженнями.

Подальший розвиток вимагає вдосконалення представлення багатокубітних гейтів шляхом використання одно- та двокубітних гейтів, а також вдосконалення оснастки для роботи з квантовими комп'ютерами та розробки багатокубітних гейтів та впровадження їх у діючі зразки квантових комп'ютерів.

**Висновки**

Хоча квантовий симулятор вказує на те, що схеми повинні надавати правильний результат із ймовірністю близькою до максимальної, результати реальних експериментів не є навіть близько такими вдалими. Причини таких результатів та можливість отримання кращих результатів підлягають подальшим дослідженням.

З результатів проведених досліджень, можна зробити висновок. що реалізація задач криптології на доступних для громадськості квантових комп'ютерах має свої складнощі, котрі потрібно вирішувати.

ЛИТЕРАТУРА

1. Lov K. Grover, A fast quantum mechanical algorithm for database search, 1996. URL: https://arxiv.org/pdf/quant-ph/9605043.pdf(Last accessed: 15.02.2020)
2. Горбенко, І. Д. Прикладна криптологія. [Текст]: монографія / І. Д. Горбенко, Ю. І. Горбенко; ХНУРЕ. – Х.: Форт, 2012. - 868 с.
3. Сутність та особливості реалізації методу Гровера на класичному комп'ютері для симетричного криптоаналізу / Ю. І. Горбенко, Є. Ю. Каптьол // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. – 2018. Вип. 195. – С. 89-100.

**КАПТЬОЛ Євгеній Юрійович** – магістр, науковий співробітник-консультант; Приватне акціонерне товариство "Інститут інформаційних технологій", м. Харків, вул. Бакуліна, 12, 61166; e-mail: kaptevg@iit.kharkov.ua; kaptevg@gmail.com; ORCID: 0000-0001-8612-2196.

Наукові інтереси:

*– криптологія, використання квантових комп'ютерів в криптології.*

**ГОРБЕНКО Іван Дмитрович** – д. т. н., професор; професор кафедри БІСТ, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 6, м.Харків, Україна, 61022; e-mail: GorbenkoI@iit.kharkov.ua.

Наукові інтереси:

*– прикладна криптологія, системи , комплекси та засоби КЗІ.*

**УДК 629.7.036.3**

КОВАЛЬОВ А.В., ЛИСИЦЯ О.Ю., МИХАЙЛЕНКО Т.П., ПЕТУХОВ І.І.

# ОСОБЛИВОСТІ МОДЕЛЮВАННЯ ПРОЦЕСІВ В МАСЛЯНІЙ ПОРОЖНИНІ ОПОРИ РОТОРА ГАЗОТУРБІННОГО ДВИГУНА

Коректне прогнозування потокорозподілу масло-повітряної суміші є важливим питанням при проектуванні маслосистем газотурбінних двигунів (ГТД). Вирішення цього завдання може підвищити не тільки ефективність роботи маслосистеми, але і двигуна в цілому. Проте процеси, що мають місце в масляній порожнині, дуже складні і на сьогодні досліджені мало, а існуючі методи розрахунку не враховують повною мірою особливості робочого процесу в опорі. Більш спрощені математичні моделі розглядають теплогідравлічні процеси за участю двох фаз – масла і повітря, проте в реальних камерах має місце і третя фаза – плівка масла. Остання утворюється на внутрішніх стінках камери та багато в чому визначає коефіцієнт тепловіддачі від внутрішньої поверхні камери до потоку.

Фізична модель процесу включає кілька складових, кожна з яких має бути описана якнайбільш повно. Масло, потрапляючи через форсунку на підшипник, розтікається по його поверхні. Під дією відцентрових сил краплі масла відокремлюються від підшипника і рухаються до внутрішньої поверхні масляної порожнини. Під час свого руху краплі взаємодіють з потоком повітря, обмінюючись імпульсом та енергією. На стінках камери деякі краплі формують масляну плівку, що рухається під дією аеродинамічних сил на міжфазній поверхні, сил тяжіння і в'язкості. На утворену плівку впливають як краплі, що потрапляють від підшипника, так і повітряний потік, який може зривати і нести краплі з поверхні плівки. Внаслідок такої складної взаємодії плівка має неоднорідну товщину, а це додатково впливає на теплообмін між стінками і масло-повітряною сумішшю [1].

Взаємозв'язок теплових та гідродинамічних процесів, складність геометрії масляної порожнини, граничних умов зумовлюють необхідність використання методів моделювання на основі механіки багатофазних середовищ та обчислювальної гідродинаміки [2-4]. Проте залишаються відкритими питання вибору структури математичної моделі багатофазного потоку і граничних умов, методу CFD-моделювання. Крім того, моделювання ускладнюють невизначеності, пов'язані з описом поведінки масляної плівки, ідентифікації параметрів крапельного потоку від підшипника і умов міжфазної взаємодії в пристінній області.

В якості досліджуваної геометрії масляної порожнини була прийнята спрощена камера підшипника, для якої у відкритому доступі є результати експериментальних досліджень [5]. Для проведення чисельного моделювання була сформована геометрична модель камери (рис. 1) та побудована сітка (рис. 2). З метою зменшення витрат часу на розрахунки при збереженні прийнятної їх точності використовувалася структурована сітка із загальним числом елементів 2 млн. з мінімальним кроком поблизу стінок 0,1 мм і максимальним – в основному об'ємі 0,5 мм (рис. 2).

Для моделювання трифазного потоку в масляній порожнині була створена CFD-модель в рамках підходу Ейлера з використанням моделей багатофазного потоку «VOF» та «Eulerian», які реалізовані в ANSYS Fluent 2019 R2. В якості моделей турбулентності використовувалися «k-ε Realizable» та «k-ω SST», обидві з яких продемонстрували прийнятну узгодженість з експериментальними даними. Похибка не перевищувала 15%. Для коректного врахування особливостей потоку в пристінній області використовувалася функція Enhanced Wall Treatment.

В результаті розрахунків за допомогою створеної CFD-моделі було отримано поля та вектори швидкості (рис. 3), поля температури (рис. 4), тиску як для суміші в цілому, так і для окремих фаз. Розраховано розподіл об'ємної частки масла по внутрішній порожнині.

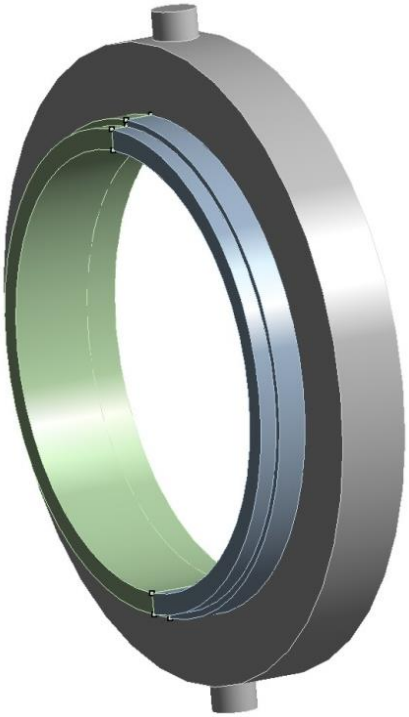


*Рис. 1. Геометрична модель камери*

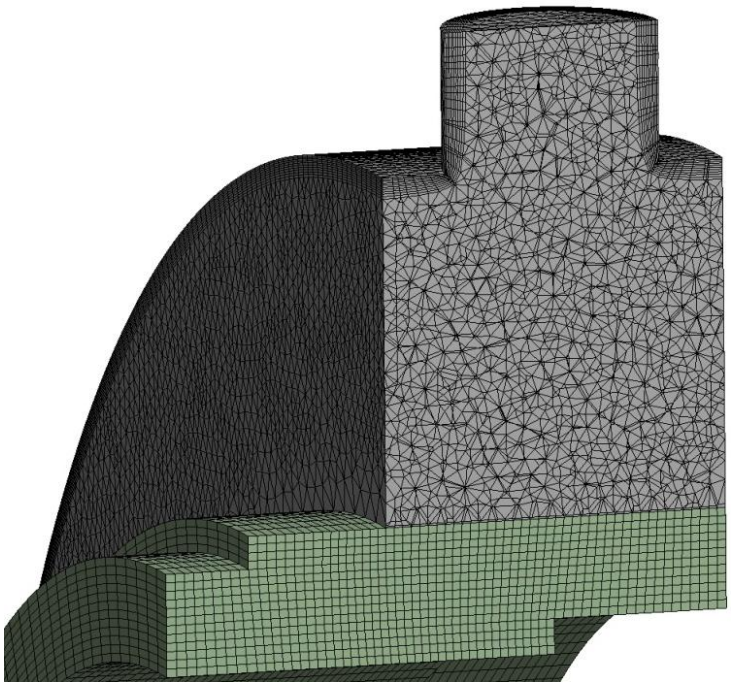


*Рис. 2. Розрахункова сітка*

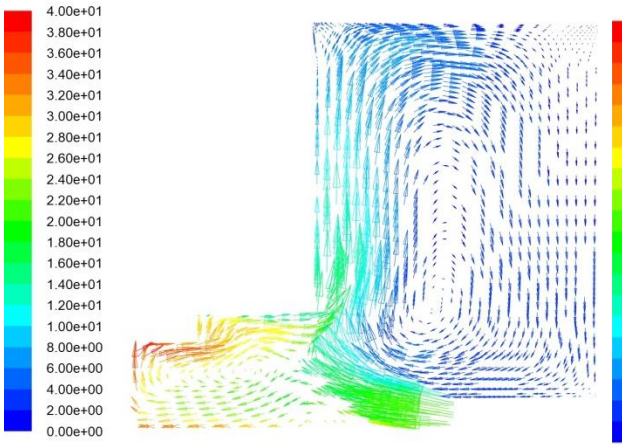



*Рис. 3. Вектори швидкості повітря(м/с)*

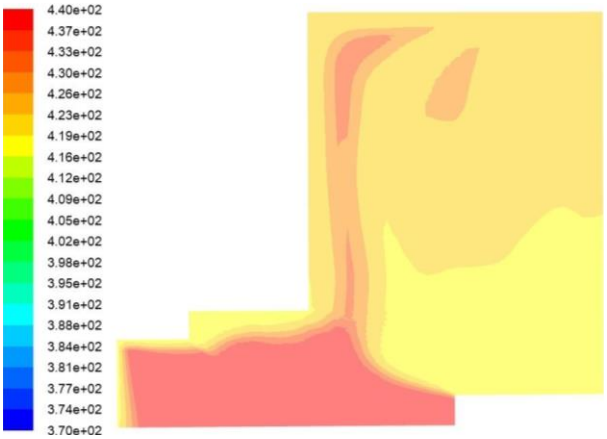



*Рис. 4. Поле температури суміші (К)*

Аналізуючи вектори швидкості для різних варіантів розрахунку, були виявлені певні невідповідності в розмірі вихрових зон і їх розташуванні. Біля входу масла в камеру спостерігалося утворення додаткових локальних вихорів. У подібних точках простору для різних розрахункових випадків відмінності в швидкості окремих фаз досягали 7 м/с, а суміші – 5 м/с. Найбільша невідповідність спостерігалася в пристінних областях і місцях різкого повороту потоку, що може бути пояснено як складністю фізики самих процесів, так і недосконалістю математичного апарату для їх опису.

Окремо досліджувалося питання утворення плівки масла на стінках камери. Для цього використовувалася модель «ейлерової плівки» Eulerian Wall Film, яка враховує взаємодію повітряно-крапельного потоку з плівкою масла, що сформувалася на стінках. Лінії току масляної фази та розподіл плівки масла показано на рис. 5 та рис. 6, відповідно.

Як видно з рис. 5, краплі масла під впливом відцентрових сил і безпосередньо повітряного потоку скупчуються на стінці камери, утворюючи плівку. Згідно з результатами розрахунку (рис. 6), товщина масляної плівки по окружності камери змінна. В середньому товщина плівки складає 0,3… 1 мм. Швидкість руху плівки визначається силою тяжіння і зсуву на межі поділу фаз «плівка-повітря».

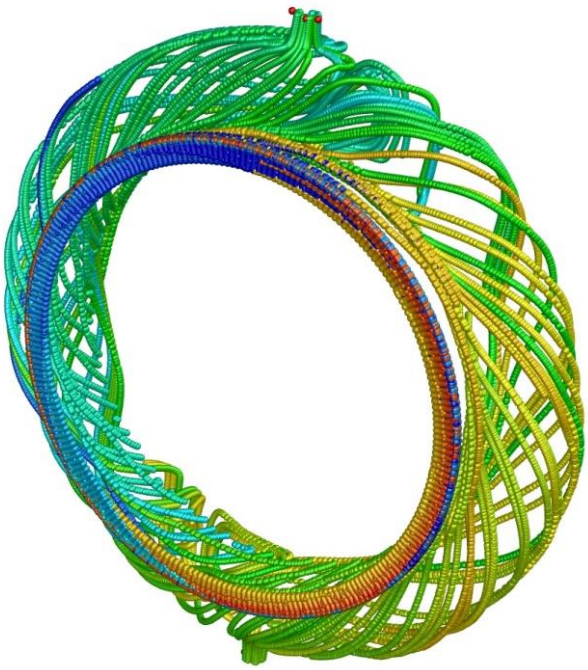


*Рис. 5. Лінії току масляної фази*

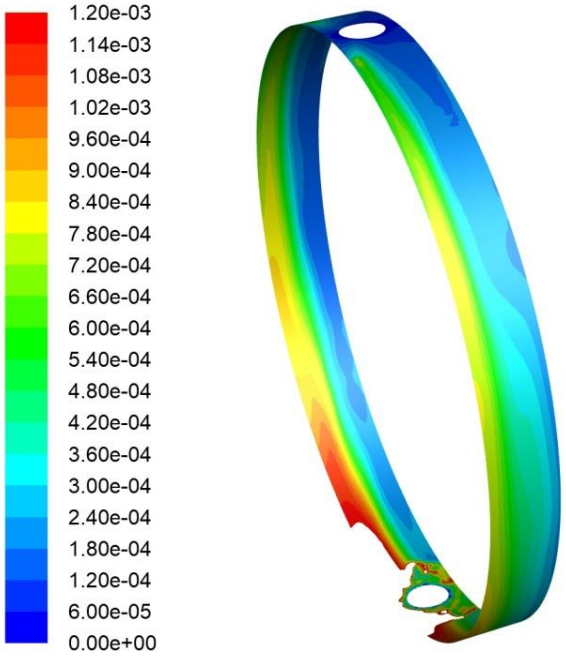



*Рис. 6.Розподіл товщини плівки масла (мм)*

Сила тяжіння також впливає на товщину плівки. Через її асиметричний розподіл (рис. 6), складний рух повітря і масляних крапель (рис. 5) спостерігаються різні значення термічного опору приграничного шару, а значить, і коефіцієнта тепловіддачі. Цей факт підтверджується результатами розрахунків за розробленою авторами моделлю.

Таким чином, прийнята концепція моделювання та сформована CFD-модель дозволяють виконати чисельне дослідження трифазного потоку в масляних порожнинах газотурбінних двигунів на етапі проектування. CFD-модель повинна містити алгоритми, що враховують формування пристінної масляної плівки і її взаємодію з прилеглим крапельно-повітряним потоком. На основі отриманих результатів чисельного дослідження було зроблено висновки щодо впливу режимних параметрів, внутрішньої геометрії, теплофізичних властивостей середовищ, розміру дисперсної фази на потокорозподіл в масляній порожнині.

Найбільшу узгодженість з експериментальними значеннями коефіцієнта тепловіддачі показала модель Ейлера з використанням нестаціонарного вирішувача типу Pressure-Based. Нерівномірність потокорозподілу в камері підшипника призводить до змінного значення коефіцієнта тепловіддачі, що значно впливає на тепловий потік в масляну порожнину опори ротора, на температурний стан масла і елементів опори. Істотний вплив сили тяжіння на формування пристінної масляної плівки і коефіцієнт тепловіддачі вимагає обов'язкового врахування гравітації при моделюванні робочого процесу в масляній порожнині опори.

Необхідно також зазначити, що запропонована модель не враховує повною мірою зміну діаметра крапель масла, що вилітають з підшипника, їх траєкторію руху і дроблення. Залишається також не до кінця вирішеним питання задання граничних умов на поверхні підшипника і температурного поля деталей, що примикають до досліджуваної області. Особливо це актуально для тих масляних порожнин, в яких підшипник одночасно є входом і виходом для масла, що подається в камеру форсунками. Відповіді на ці питання можуть бути отримані головним чином в результаті експериментальних досліджень реальних камер підшипника ГТД.

Отримані результати дозволять підвищити точність розрахунку температурного стану елементів опори ротора ГТД, дадуть можливість проведення інженерних розрахунків щодо впливу геометрії, частоти обертання ротора і витрат фаз на тепловіддачу в масляній порожнині, що зменшить у подальшому обсяг експериментальних робіт при проектуванні подібних систем.

ЛІТЕРАТУРА

1. Kakimpa B., Morvan H., Hibberd S. The numerical simulation of multi-scale oil films using coupled VOF and Eulerian thin-film model. ASME Turbo Expo 2016: Turbomachinery Techn. Conf. and Expos., Seoul, South Korea. 2016. Vol. 1: Aircraft Engine. Fans and Blowers. Marine. P. 10. DOI: 10.1115/GT2016-56747.
2. Tatar V., Piskin A. Numerical Investigation On Bearing Chamber Wall Heat Transfer. ASME Turbo Expo 2018: Turbomachinery Techn. Conf. and Expos., Oslo, Norway. 2018. Vol. 1: Aircraft Engine. Fans and Blowers. Marine. P. 8. DOI: 10.1115/GT2018-75721.
3. Prabhakar A., Ambrose S., Morvan H. Numerical Investigation of Two Phase Flow in a Dual Drive Booster. ASME Turbo Expo 2019: Turbomachinery Techn. Conf. and Expo., Phoenix, Arizona, USA. 2019. Vol. 1: Aircraft Engine; Fans and Blowers; Marine. P. 8. DOI: 10.1115/GT2019-90347.
4. M. Berthold et al. Multiphase CFD Modeling Of External Oil Flow From A Journal Bearing. ASME Turbo Expo 2018: Turbomachinery Techn. Conf. and Expo., Oslo, Norway. 2018. Vol. 1: Aircraft Engine. Fans and Blowers. Marine. P. 12. DOI: 10.1115/GT2018-77130.
5. Busam, S., Glahn A., Wittig S. Internal Bearing Chamber Wall Heat Transfer as a Function of Operating Conditions and Chamber Geometry. J. Eng. Gas Turbines Power. 2000. Vol. 122, Issue 2. P. 7. DOI: 10.1115/1.483209.

**КОВАЛЬОВ Артем Вікторович –** аспірант; Національний аерокосмічний університет ім. М.Є. Жуковського "ХАІ", вул. Чкалова 17, м. Харків, Україна, 61070. E-mail: kovalev.205khai@gmail.com. Номер ORCID: https://orcid.org/0000-0002-9493-9769

Наукові інтереси:

- CFD-моделювання термогідравлічних процесів.

**ЛИСИЦЯ Олексій Юрійович** – к. т. н., доцент; Національний аерокосмічний університет ім. М.Є. Жуковського "ХАІ", вул. Чкалова 17, м. Харків, Україна, 61070. E-mail: lis.o@ukr.net. Номер ORCID: https://orcid.org/0000-0002-5679-8459

Наукові інтереси:

- CFD-моделювання теплогідравлічних процесів в двигунах та енергетичних установках.

**МИХАЙЛЕНКО Тарас Петрович** – к. т. н., доцент, доцент кафедри аерокосмічної теплотехніки; Національний аерокосмічний університет ім. М.Є. Жуковського "ХАІ", вул. Чкалова 17, м. Харків, Україна, 61070. E-mail: t.mykhailenko@khai.edu. Номер ORCID: https://orcid.org/0000-0003-4708-673X.

Наукові інтереси:

- теплогідравлічні процеси в двигунах та енергетичних установках.

**ПЕТУХОВ Ілля Іванович** – к. т. н., доцент, доцент кафедри аерокосмічної теплотехніки; Національний аерокосмічний університет ім. М.Є. Жуковського "ХАІ", вул. Чкалова 17, м. Харків, Україна, 61070. E-mail: ilya2950@gmail.com. Номер ORCID: http:// orcid.org/ 0000-0002-0645-7912.

Наукові інтереси:

- термогідравлічні процеси у багатофазних потоках.

**УДК 004.056, 004.056.55, 004.91**

КОНДРЯ Ю.О.

# ПОРІВНЯЛЬНИЙ АНАЛІЗ ЕЛЕКТРОННИХ ЦИФРОВИХ ПІДПИСІВ НА ОСНОВІ ЗАДАЧ ТЕОРІЇ РЕШІТОК ТА БАГАТОВИМІРНИХ КВАДРАТИЧНИХ СИСТЕМ

**Вступ**

Поява квантових комп'ютерів абсолютно точно зробить революцію в інформаційній безпеці, під загрозою перебуває безпека будь-яких криптографічних систем з відкритим ключем.

До недавнього часу причин для занепокоєння не було, так як у творців квантового комп'ютера є дві фундаментальні проблеми. По-перше, стан квантової системи, а по-друге, «кубіти» (розряди квантового комп'ютера) дуже нестабільні, утримати їх в потрібному стані вкрай важко, і чим більше кубітів, тим важче. Загрозу для сучасних криптографічних систем потенційно представляють комп'ютери з сотнями і навіть тисячами кубітів. Однак намітився несподівано швидкий прогрес: з'явилося багато публікацій по даній тематиці.

В силу вище сказаного, очевидною стає необхідність подальшого розвитку постквантової криптографії. Так як схеми електронного підпису повністю втратять свою криптостійкість в разі появи квантового комп'ютера, то саме для цих криптографічних функцій першочерговою необхідністю є пошук нових постквантових аналогів.

**Постановка задачі**

Більшість поширених криптографічних алгоритмів, що використовуються при шифруванні і роботі з електронними підписами, засновані на тому, що розкладання чисел на прості множники, дискретне логарифмування і деякі інші математичні задачі мають дуже високу складність. Звичайні комп'ютери принципово не здатні виконати необхідні обчислення за розумний час.

У 1994 році вчений Пітер Шор з дослідницького центру Bell Labs виявив, що універсальні квантові комп'ютери здатні вирішувати ці завдання за поліноміальний час. Іншими словами, так швидко, що популярні алгоритми шифрування з відкритим ключем втрачають сенс. Через два роки американський математик Лов Гровер винайшов квантовий алгоритм пошуку, який значно прискорює розшифровку симетричних криптоалгоритмів.

Винаходи Шора і Гровера не знищили сучасну криптографію з єдиної причини: квантових комп'ютерів достатньої потужності поки не існує.

Багато експертів вважають, що квантові комп'ютери, що здатні долати будь-які сучасні криптосистеми з відкритим ключем, з'являться протягом найближчих десяти років. За оцінкою фахівців NIST [1], до 2030 року злом алгоритму RSA з ключем довжиною 2000 біт буде займати лічені години. Це буде недешево (знадобиться не менше мільярда доларів), але цілком реально.

Алгоритм AES можна буде врятувати, істотно подовживши ключ, але алгоритми RSA, ECDSA, ECDH, DSA і багато інших стануть небезпечні.

Відомо кілька криптоалгоритмів, заснованих на математичних проблемах, які поки не виходить спростити за рахунок застосування квантових комп'ютерів. В NIST високо оцінюють алгоритми асиметричного шифрування, засновані на решітках: вони прості, ефективні і добре паралелізуються.

З появою реальних квантових комп'ютерів перехід на постквантові протоколи може стати занадто різким, що спричинить за собою великі фінансові втрати. Тому розглядати, аналізувати і впроваджувати постквантову криптографію в існуючі системи потрібно починати вже зараз.

**Метою роботи** є аналіз і порівняльні дослідження схем ЕЦП постквантового періоду для виявлення найбільш перспективної схеми.

**Обґрунтування критеріїв та показників ефективності ЕЦП в умовах квантового криптоаналізу**

В 2016 році Національний Інститут Стандартів і Технологій США (NIST) оголосив

про старт конкурсу на створення нових постквантових алгоритмів і стандартів замість старих.

Результати першого туру були оголошені 30 січня 2019 року [1, 2]. До другого туру пройшли 26 кандидатів: 17 алгоритмів шифрування і розподілу ключів і 9 схем електронного підпису.

Схеми електронного підпису, що пройшли до другого туру представлені у таблиці 1.

*Табл.1 Схеми ЕЦП, що пройшли до другого туру*

| **Постквантовий підхід** | **Алгоритм** |
|---|---|
| Схеми на решітках | CRYSTALS-DILITHIUM |
| | FALCON |
| | qTESLA |
| Багатовимірні квадратичні системи | GeMSS |
| | LUOV |
| | MQDSS |
| | Rainbow |
| Електронні підписи на хеш-функціях | Picnic |
| | SPHINCS+ |

Конкурс є повністю відкритим. Організатори закликають всі криптографічні наукові співтовариства, навіть вже вибулих кандидатів прийняти участь в розгляді заявок, для більш скрупульозного виявлення недоліків.

Три основні критерії, за якими здійснювалася оцінка кандидатів, були наступні: безпека, швидкодія і використання ресурсів пам'яті, характеристики алгоритмів і нюанси реалізацій [2, 3].

В результаті аналізу постквантових підходів були обрані два: на основі задач теорії решіток (Lattices) та багатовимірних завадостійких кодів (Multivariate), що привертають увагу завдяки своєї швидкості та довжині ключів (табл. 2).

*Табл.2 Порівняння постквантових підходів*

| Постквантовий підхід | **Теорія решіток** | **Багатовимірні квадратичні системи** |
|---|---|---|
| **Обґрунтування складності** | Розв’язання задач теорії решіток в особливих решітках. Знаходження «гарного» базису решітки | Вирішення систем багатовимірних квадратичних рівнянь |
| **Швидкодія** | Добре реалізується на спеціальному програмному забезпеченні | Добре реалізується на апаратних засобах |
| **Переваги** | Безліч сфер застосування. Обґрунтування складності в найгіршому випадку | Швидкість. Невелика довжина ключів |
| **Недоліки** | Відсутність точного методу | Неспроможність обґрунтування безпеки. Підвищена довжина відкритого ключа |

Різні криптографічні схеми, крім забезпечення належного рівня безпеки, оцінюються за іншими параметрами, наприклад, за швидкодією (тобто за швидкістю виконання процедур, як на стороні відправника, так і на стороні одержувача), необхідною пам’яттю обробного пристрої (як відправника, так і одержувача) та за довжинами закритих ключів, які необхідно зберігати, і за деякими іншими параметрами [4].

Для визначення перспективних кандидатів постквантового підпису сформулюємо ряд критеріїв порівняння досліджуваних алгоритмів:

- захищеність алгоритму підпису, яка оцінюється рівнем безпеки (безумовний критерій);
- довжина ключів в байтах (умовний критерій);

- довжина підпису в байтах (умовний критерій).

Розглянемо кандидатів другого етапу більш детально (табл. 3).

*Табл.3 Схеми ЕЦП, що пройшли в другий тур конкурсу NIST*

| **Пост-квантовий підхід** | **Алгоритм** | **Конкретна реалізація** | **Закритий ключ, байт** | **Відкритий ключ, байт** | **Довжина підпису, байт** | **Категорія безпеки NIST** |
|---|---|---|---|---|---|---|
| Теорія решіток | CRYSTALS-Dilithium | Delithium_ medium | 2 800 | 1 184 | 2 044 | 1 |
| | | Delithium_ recommended | 3 504 | 1 472 | 2 701 | 2 |
| | | Dilithium_ very_high | 3 856 | 1 760 | 3 366 | 3 |
| | Falcon | Falcon1024 | 8 193 | 1 793 | 1 330 | 5 |
| | | Falcon512 | 4 097 | 897 | 690 | 1 |
| | | Falcon768 | 6 145 | 1 441 | 1 077 | 3 |
| | qTesla | qTesla_128 | 2 112 | 4 128 | 3 104 | 1 |
| | | qTesla_192 | 8 256 | 8 224 | 6 176 | 3 |
| | | qTesla_256 | 8 256 | 8 224 | 6 176 | 5 |
| Багато-вимірні квадратичні системи | LUOV | luov-48-49-242 | 32 | 7 536 | 1 746 | 2 |
| | | luov-64-68-330 | 32 | 19 973 | 3 184 | 4 |
| | | luov-80-86-399 | 32 | 40 248 | 4 850 | 5 |
| | | luov-8-117-404 | 32 | 100 989 | 521 | 5 |
| | | luov-8-63-256 | 32 | 15 908 | 319 | 2 |
| | | luov-8-90-351 | 32 | 46 101 | 441 | 4 |
| | Rainbow | Ia | 100 209 | 152 097 | 64 | 1 |
| | | Ib | 114 308 | 163 185 | 78 | 1 |
| | | Ic | 143 385 | 192 241 | 104 | 1 |
| | | IIIb | 409 463 | 564 535 | 112 | 3 |
| | | IIIc | 537 781 | 720 793 | 156 | 3 |
| | | IVa | 376 141 | 565 489 | 92 | 4 |
| | | Vc | 1 274 317 | 1 723 681 | 204 | 5 |
| | | VIa | 892 079 | 1 351 361 | 118 | 5 |
| | | VIb | 1 016 868 | 1 456 225 | 147 | 5 |
| | GeMSS | GeMSS128 | 14 208 | 417 408 | 48 | 1 |
| | | GeMSS192 | 39 440 | 1 304 192 | 88 | 3 |
| | | GeMSS256 | 82 056 | 3 603 792 | 104 | 5 |
| | MQDSS | mqdss-48 | 32 | 62 | 32 882 | 2 |
| | | mqdss-64 | 48 | 88 | 67 800 | 4 |

Для схем підписів визначальним параметром є довжина підпису, а також довжина відкритого ключа підписувача, тому при подальшому аналізі будемо особливу увагу приділяти наступному параметру: $|pk| + |Sign|$, де $|pk|$ – довжина відкритого ключа, а $|Sign|$ – довжина підпису.

Найбільш перспективними є схеми на основі решіток, багатовимірних квадратичних многочленів і схеми на основі хеш-функцій. Кожна зі схем має різні реалізації в залежності від забезпечуваного рівня безпеки відповідно до вимог конкурсу.

Для схем електронного підпису визначальними параметрами є довжини ключів (особливо відкритого ключа) і підпису. У меншій мірі – час генерації, підписання та перевірки. Тому, пропонується виключити схеми GeMSS і Rainbow, так як довжини ключів в цих схемах досягають декількох сотень Кбайт, також, пропонується виключити підписи, засновані на хеш-функціях – Picnic і Sphincs+ – в силу їх обмеженості за кількістю генеруємих підписів на одній парі ключів.

Тому для подальшої вибірки були обрані наступні схеми: CRYSTALS-DILITHIUM, FALCON, qTESLA, LUOV, MQDSS.

Перший рівень безпеки забезпечують тільки схеми на решітках. Багатовимірні квадратичні схеми мають більш високі рівні безпеки. Якщо брати до уваги тактові витрати, то найцікавішою є схема qTesla_128. Що стосується параметрів, то явно виграє Falcon512.

Серед схем другого та третього рівнів найбільшою швидкодією володіє CRYSTALS-Dilithium, а MQDSS – найменшою. MQDSS-48 має короткі ключі, але занадто велику довжину підпису, що абсолютно неприйнятно для гібридних систем. Якщо порівнювати параметри, то знову явну перевагу мають схеми на решітках, зокрема, схема Falcon768.

На четвертому та п'ятому рівнях безпеки багатовимірні квадратичні схеми LUOV мають занадто великі довжини відкритих ключів, а схема mqdss-64 – велику довжину підпису.

**Висновки**

В результаті проведеного порівняльного аналізу існуючих постквантових підходів були знайдені переваги та недоліки цих підходів, було показано перевагу криптоперетворень ЕЦП над іншими алгоритмами за застосованими критеріями порівняння та було визначено найбільш перспективного кандидата – схему FALCON (у всіх трьох конкретних реалізаціях), яка на всіх рівнях безпеки забезпечувала найменшу довжину ключів і підпису з усіх представлених алгоритмів.

ЛІТЕРАТУРА

1. Alagic J., Alperin-Sheriff J., Apon D., Cooper D., Dang Q., Yi-Kai Liu., Miller C., Moody D., Peralta R., Perlner R., Robinson A., SmithTone D. Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process, NISTIR 8240, National Institute of Standards and Technology, Gaithersburg, Maryland, January 2019, 27 pp. URL: http://dx.doi.org/10.6028/NIST.IR.8240 (Last accessed: 03.03.2020).
2. Bernstein DJ, Lange T. Post-quantum cryptography: dealing with the fallout of physics success, IACR, 2017. 20 p.
3. Chen L., Jordan S., Liu Y.K., Moody D., Peralta R., Perlner R., Smith-tone D. Report on Post-Quantum Cryptography, NISTIR 8105, National Institute of Standards and Technology, Gaithersburg, Maryland, April 2016, 10pp. URL: https://doi.org/10.6028/NIST.IR.8105 (Last accessed: 28.02.2020).
4. Gorbenko I., Ponomar V., Yesina M. Research of usage possibility and post-quantum algorithms advantages depend on application conditions. 2017. Issue 3(7). URL: https://periodicals.karazin.ua/cscs/article/download/10005/9921 (Last accessed: 02.03.2020).

**КОНДРЯ Юлія Олегівна** – студентка факультету комп'ютерних наук, кафедри безпеки інформаційних систем і технологій ХНУ імені В. Н. Каразіна, майдан Свободи, 6, м. Харків, Україна, 61022; e-mail: uliakondria@gmail.com; ORCID: 0000-0002-2806-583X.

Наукові інтереси:

- *криптографічний захист інформації, криптографічні системи і протоколи, постквантова криптографія.*

**УДК 519.87**

КОСОЛАП А.И.

# МУЛЬТИМОДАЛЬНЫЕ ЗАДАЧИ В КОМПЬЮТЕРНЫХ СИСТЕМАХ

С каждым годом внутренняя структура компьютерных систем усложняется. Для разработки компьютеров, компьютерных сетей и их программного обеспечения требуется огромное количество ресурсов, специалистов и времени. При моделировании таких систем широко используется математическое и компьютерное моделирование, которое чаще всего приводит к оптимизационным моделям. Такие модели содержат большое число переменных, часто дискретных, для решения которых необходимы эффективные численные методы. Дискретные модели могут быть преобразованы к непрерывным. После такого преобразования задачи становятся мультимодальными, часто с большим числом локальных экстремумов. Для решения таких задач в последнее время используются эволюционные алгоритмы [1], в основе которых лежит случайный поиск. Это означает, что такие алгоритмы не гарантируют нахождения лучшего решения. Как показывают многочисленные эксперименты на мультимодальных тестовых задачах эти алгоритмы часто находят решения далекие от оптимальных. Кроме того, они содержат много регулируемых параметров, от значений которых зависит результат решения. Все это вынуждает строить новые методы для решения мультимодальных задач. Мы предлагаем для решения таких задач использовать метод точной квадратичной регуляризации [2]. Численные эксперименты показали его значительное преимущество при решении мультимодальных тестовых задач [3].

Рассмотрим задачу равномерного распределения заданий в многопроцессорной системе. Известно время $t_i$ – выполнения $i$-го задания, число заданий $n$ и количество процессоров $m$. Необходимо равномерно распределить задания по процессорам, чтобы многопроцессорная система завершила обработку всех заданий за минимальное время. Это равносильно решению следующей задачи

$$\min\{\sum_{i=1}^{n}(\sum_{j=1}^{m}t_j x_{ij})^2 \mid \sum_{j=1}^{m}x_{ij}=1, i=1,...,n, x=0\vee 1\},$$

где переменные принимают только булевы значения

$$x_{ij}=\begin{cases}1, & \text{если } i\text{-е задание выполняется } j\text{-м процессором}\\ 0, & \text{в противном случае}\end{cases}$$

Эта комбинаторная задача преобразуется к квадратичной мультимодальной задаче

$$\min\{\sum_{i=1}^{n}(\sum_{j=1}^{m}t_j x_{ij})^2 \mid \sum_{j=1}^{m}x_{ij}=t_j, i=1,...,n, \sum_{i=1}^{n}\sum_{j=1}^{m}x_{ij}^2\ge\sum_{j=1}^{m}t_j^2\}.$$

Мультимодальность полученной задачи порождается последним ограничением.

Актуальной является задача оптимального размещения сенсорных датчиков для сбора данных. Часто возникает задача увеличения числа установленных датчиков и определения координат размещения новых датчиков. Обозначим координаты установленных датчиков через $a^i$, $i$=1,…,$m$, а координаты новых датчиков через $x^j$, $j$=1,…,$n$. Зададим расстояния между каждой парой датчиков: $d_{ij}$ – расстояние между $i$-м установленным датчиком и $j$-м новым, а $h_{jk}$ – расстояние между $j$-м и $k$-м новыми датчиками. Тогда для определения координат новых датчиков необходимо решить квадратичную систему уравнений

$$\| a^i - x^j \|^2 = d_{ij}^2, \| x^j - x^k \|^2 = h_{jk}^2, \forall ijk.$$

Эта система не всегда имеет решение, поэтому введем новые переменные

$$\| a^i - x^j \|^2 = d_{ij}^2 + u_{ij}, \| x^j - x^k \|^2 = h_{jk}^2 + v_{jk}, \forall ijk.$$

и минимизируем их значения

$$\min\{\| u \|^2 + \| v \|^2 \mid \| a^i - x^j \|^2 = d_{ij}^2 + u_{ij}, \| x^j - x^k \|^2 = h_{jk}^2 + v_{jk}, \forall ijk\}.$$

Мы получили квадратичную мультимодальную задачу для определения координат датчиков.

Задача оптимального размещения микроэлементов на печатных схемах давно привлекает исследователей, проектировщиков и производителей интегральных схем. Многие изготовители печатных схем разработали пакеты программ для их проектирования [4]. Однако в этих пакетах реализуется преимущественно рациональное размещение микроэлементов. Оптимальное размещение микроэлементов возможно посредством построения новых моделей и использования метода точной квадратичной регуляризации.

Новая модель размещения микроэлементов на пластине заключается в следующем. Имеется прямоугольная пластина со сторонами $a$ и $b$, на которой необходимо расположить заданное число микроэлементов. Часть микроэлементов соединены между собой проводниками. Чем короче суммарная длина проводников, тем выше быстродействие данной печатной схемы. Эта длина зависит от расположения микроэлементов на пластине. При построении математической модели этой задачи основная сложность заключается в определении условий непересечения микроэлементов. Обычно микроэлементы являются прямоугольниками, которые, как правило, располагаются по горизонтали или вертикали пластины. Условия непересечения легко выписать для кругов. Так, два круга не пересекаются, если расстояние между их центрами не меньше суммы их радиусов. Это равносильно неравенству

$$(x_i - x_j)^2 + (y_i - y_j)^2 \geq (r_i + r_j)^2,$$

где $(x_i, y_i)$ – центр $i$-го круга, а $r_i$ – его радиус. Каждый микроэлемент (прямоугольник) можно заполнить непересекающимися кругами. Тогда непересечение двух микроэлементов равносильно непересечению заполняющих их кругов. Расположение каждого микроэлемента (прямоугольника) на пластине однозначно определяется координатами трех его вершин. Эти вершины должны удовлетворять условию

$$(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 + (x_{i+1} - x_{i+2})^2 + (y_{i+1} - y_{i+2})^2 = (x_i - x_{i+2})^2 + (y_i - y_{i+2})^2.$$

Теперь центры кругов, которые определяют микроэлемент, будут определяться через искомые координаты вершин прямоугольников. Далее необходимо вычислить суммарную длину соединений между микроэлементами. Эти соединения можно представить графом $G(N, V)$, где $N$ – множество его вершин (микроэлементов), а $V$ – множество дуг (соединений). Тогда целевой функцией данной задачи будет следующая

$$\min\{\sum_{i,j \in V} [(x_i - x_j)^2 + (y_i - y_j)^2]\}.$$

Необходимо учесть условия, что все круги полностью располагаются на пластине

$$r_i \leq x_i \leq a - r_i, r_i \leq y_i \leq b - r_i.$$

Мы снова получили квадратичную мультимодальную задачу.

К квадратичным оптимизационным задачам относятся задачи о разделении областей в евклидовом пространстве на заданное число непересекающихся подобластей. Область разбита на $m$ частей, если для каждой подобласти определена точка притяжения. Каждая точка притяжения однозначно определяет подобласть. Это точки области ближайшие к данной точке притяжения в евклидовой метрике. Пусть задано число $m$ подобластей на которое необходимо разбить данную область. Тогда точками притяжения могут быть центры непересекающихся шаров максимального радиуса, вписанные в данную область. Рассмотрим упаковку квадрата кругами. В этой задаче необходимо минимизировать сторону квадрата, содержащего $m$ непересекающихся кругов единичного радиуса

$$\min\{ z \mid (x_i - x_j)^2 + (y_i - y_j)^2 \geq 4, \forall i \neq j, 1 \leq x_i \leq z - 1, 1 \leq y_i \leq z - 1, i = 1,..., m\}.$$

В общем случае, задача оптимизации записывается в виде

$$\min\{ f_0(x) \mid f_i(x) \leq 0, i = 1,..., m, x \in E^n\}, \quad (1)$$

где $E^n$ – $n$-мерное евклидовое пространство, а функции $f_i(x)$, $\forall i$ дважды дифференцированы.

Используем квадратичную регуляризацию для преобразования задачи (1) к следующей

$$\max\{\| z \|^2 \mid f_0(x) + s + (r-1) \| z \|^2 \leq d, f_i(x) + r \| z \|^2, i = 1,..., m\}, \quad (2)$$

где $z = (x, x_{n+1})$, $s$ и $r$ – параметры, а $d$ – новая переменная. Параметр $r > 0$ выбираем таким, чтобы допустимое множество задачи (2) было выпуклым. Это легко сделать, если функции $f_i(x)$ квадратичные. Тогда гессиан этих функций за счет квадратичного слагаемого $r\|z\|^2$ будет матрицей с преобладающей главной диагональю. Такие матрицы являются положительно

определенными, а соответствующие им функции будут выпуклыми. Параметр $s$ должен удовлетворять условию $s \geq ||x^*||^2 - f_0(x^*)$, где $x^*$ – решение задачи (1). Другими словами, выбор $s$ должен быть таким, чтобы первое ограничение в точке максимума задачи (2) было активным.

В задаче (2) необходимо найти минимальное значение $d^*$, для которого ее решение $z^*$ удовлетворяет условию $r||z^*||^2 = d^*$. Решение задачи (2) при фиксированном значении $d$ находим прямо-двойственным методом внутренней точки [5]. При программной реализации этого метода мы использовали его в комбинации с методом Франка-Вулфа.

Для начала процедуры дихотомии необходимо определить минимально возможное значение $d_0$. Это значение является решением следующей задачи выпуклой оптимизации

$$\min\{ d \mid f_0(x) + s + (r-1)\,||\,z\,||^2 \leq d,\ f_i(x) + r\,||\,z\,||^2, i = 1,\ldots, m\}. \qquad (3)$$

Если для решения задачи (3) выполняется условие $r||z||^2 = d$, то это решение совпадает с решением задачи (1). Чаще всего будем иметь $r||z||^2 < d$, тогда значение $d$ будем увеличивать с определенным шагом до достижения равенства $r||z||^2 = d$ с заданной точностью.

Рассмотрим решение задачи (2) в самом простом случае, когда ее выпуклое допустимое множество является отрезком $[a, b]$. В этом случае, задача (2) будет иметь два локальных максимума, если точка минимума $||z||^2$ на отрезке достигается в его внутренней точке. В противном случае задача (2) будет иметь единственный максимум. Если отрезок $[a, b]$ неортогонален биссектрисе положительного ортанта, то существует такое его смещение вдоль биссектрисы на величину $h$ при котором задача максимума $||z||^2$ на смещенном отрезке $[a + h, b + h]$ будет одноэкстремальной. Однако использование точной квадратичной регуляризации приводит к задаче

$$\max\{|\,||\,z\,||^2| - ||\,z - h\,|| + s + 2\,||\,z\,||^2 \leq d, z \in [a + h, b + h]\}, \qquad (4)$$

которая в общем случае уже не будет одноэкстремальной. Решение выпуклой задачи (3) для задачи (4) будет достигаться в точке глобального максимума задачи (4) либо точка глобального максимума будет получена в результате изменения $d$. Результат решения задачи (4) обобщается на более общий случай решения задачи (2) на многограннике. В этом случае отрезками являются ребра многогранника. Эффективное решение задачи (2) на многограннике позволяет решить задачу (2) и в общем случае, так как любое выпуклое множество аппроксимируется многогранником.

Решим задачу распределения 13 заданий с временами выполнения $t_j$ = (10, 20, 15, 18, 12, 10, 8, 16, 20, 24, 16, 18, 12) на семипроцессорной компьютерной системе. Результат решения этой задачи приведен ниже (см. табл. 1).

*Табл. 1 Решение задачи распределения заданий*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $x_{ij}^*$= | 0 | 0 | 10 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 20 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 15 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 18 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 12 | 0 | 0 |
| | 0 | 10 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 8 | 0 |
| | 0 | 0 | 0 | 16 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 20 | 0 |
| | 24 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 16 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 18 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 12 |

со значением целевой функции $f(x^*) = 5697$. Решение этой задачи программой, реализующей метод ветвей и границ, дало следующий результат $f(x) = 5701$, что хуже. Метод точной квадратичной регуляризации дал лучшие результаты и при решении других рассмотренных выше задач.

Рассмотрим задачу упаковки квадрата $m = 13$ кругами. Плотность упаковки, получена методом точной квадратичной регуляризации, равна 0,733375. Результат решения представлен на рис. 1. Этот результат чуть лучше известного 0,733265 [6]. Координаты центров кругов приведены в табл. 2.

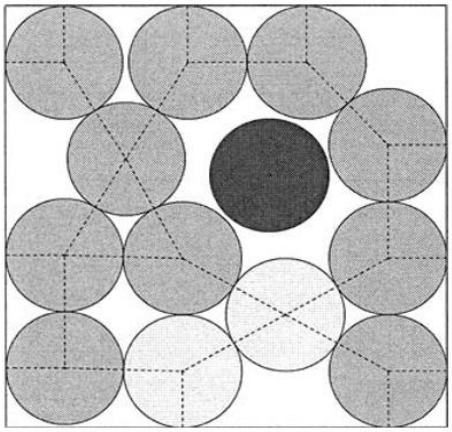


*Рис. 1 Упаковка 13 кругов в квадрат*

*Табл. 2 Координаты центров кругов*

| | | | | | | |
|---|---|---|---|---|---|---|
| $x_i$ | 1 | 1 | 1 | 3,011018 | 3,000138 | 2,0299 |
| $y_i$ | 1 | 3,03269 | 6,462488 | 1 | 3,00074 | 4,7489 |
| 3,068331 | 4,737523 | 4,357451 | 6,462488 | 6,462488 | 6,462488 | 5,0684 |
| 6,462488 | 2,011023 | 4,584883 | 1 | 3,026055 | 5,026135 | 6,4625 |

## ЛИТЕРАТУРА

1. Kenneth V. P., Storn R. M., Lampinen J. A. Differential Evolution. A Practical Approach to Global Optimization. Berlin, Heidelberg: Springer-Verlag, 2005. 542 p.
2. Косолап А. И. Глобальная оптимизация. Метод точной квадратичной регуляризации. Днепропетровск: ПГАСА, 2015. 164 с.
3. Косолап А. И. Глобальная оптимизация. Численные эксперименты. Днепр: ПГАСА, 2017. 112 с.
4. Карпов А.В., Калабанов С.А., Шагиев Р.И. Современные программные средства проектирования и моделирования печатных плат радиотехнических систем и свч-устройств. Учебно-методическое пособие. Казань, 2014. 30 с.
5. Nocedal J., Wright S.J. Numerical optimization. Springer, 2006. 685 p.
6. Szabo P.G., Markot M.Cs., Csendes T., Specht E., Casado L.G., Garcia I. New approaches to circle packing in a square. Springer, 2007. 237 p.

**КОСОЛАП Анатолий Иванович –** д.ф.-м.н., проф., зав. каф. специализированных компьютерных систем Украинского государственного химико-технологического университета, пр. Гагарина, 8, Днепр, Украина, 49005; e-mail: anivkos@ua.fm; ORCID ID: 0000-0001-73386707.

Научные интересы:

– *методы оптимизации.*

**УДК 004.452**

КРИВОГУЗОВ М.А. ЛАЗУРИК В.М.

# ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ БАЗ ДАННЫХ ВРЕМЕННЫХ РЯДОВ

## Постановка проблемы

Для обеспечения возможности хранения и обработки данных, а также организации контролируемого доступа к ним разработаны системы управления базами данных (СУБД) [1]. В связи с тем, что потребности в хранении большого объема данных непрерывно растут, постоянно разрабатываются новые и совершенствуются уже существующие технологии. Среди разнообразных видов информации, которая хранится в базах данных, следует выделить такую, в которой данные зависят от времени. Это, так называемые, временные ряды. Область применения таких данных обширна, а способы статистической обработки разнообразны. Поскольку объем временн***ы***х данных огромен, на первый план выходят задачи обеспечения быстрого доступа к ним.

В работе уделено внимание способам хранения временных рядов. Рассмотрены возможности использования для этих целей как реляционных и NoSQL баз данных, так и специализированных Time Series Data Bases (TSDB). На примере решения задачи сбора информации о посещении сайта проанализированы возможности использования InfluxDB.

## Временные ряды и их анализ

Временно́й ряд (или ряд динамики) − собранный в разные моменты времени статистический материал о значении каких-либо параметров исследуемого процесса [2]. Временные ряды – один из наиболее часто встречающихся в аналитической практике объектов. Каждому результату наблюдения (измерению) соответствует время, когда это наблюдение было сделано, или его порядковый номер по шкале времени. При анализе временных рядов учитываются не только базовые статистические закономерности, но и взаимосвязь измерений со временем. Особенностью применения математико-статистических методов для анализа временных рядов является возможность выявления структуры ряда и его прогнозирование. Прогноз будущих значений временного ряда используется для эффективного принятия решений. Временные ряды (ВР) применяются для прогнозирования [3]:

- спроса с целью снизить складские издержки и оптимизировать график работы персонала;
- нагрузки на службу доставки в пиковые интервалы времени или при пиковых нагрузках;
- нагрузки на контактный центр с целью обеспечить требуемую доступность контактного центра при минимуме затрат на фонд оплаты труда;
- трафика, чтобы обеспечить пропускной канал для устойчивой работы;
- аномалий, чтобы выявить проблемы в работе оборудования и нестандартные ситуации;
- погоды, продаж и т.д.

При прогнозировании временных рядов применяются модели:

- экспоненциального сглаживания (простое экспоненциальное сглаживание, метод Хольта, сглаживание с трендом и сезонностью Хольта – Винтерса);
- скользящего среднего (простое скользящее среднее или с сезонностью рядов);
- трендовые модели (простой тренд – линейный, логарифмический, полином, экспоненциальный или тренд с сезонностью – линейный тренд с сезонностью, логарифм с сезонностью);
- bootstrapping (прогноз нерегулярных, редких продаж).

## СУБД для работы с временными рядами

Для хранения данных, зависящих от времени, могут быть использованы разные СУБД, выбор которой определяется, прежде всего, требованиями решаемой задачи. В традиционной реляционной базе данных (РБД), например MySQL, для простейшего случая достаточно иметь поле с временной меткой `timestamp` в качестве первичного ключа и поля со значениями,

соответствующими этому времени замера. При этом быструю выборку горячих данных можно обеспечить за счет большого размера буферного пула, а для исторических данных использовать SSD диск. В случае использования РБД основными являются операции `Insert`, в то время как `Update` и `Delete` практически не используются. Операции `Select` тоже несколько отличаются от традиционного чтения данных, т.к. должны осуществляться над частью данных, представляющих собой «окно» в каком-то интервале времени. В PostgreSQL [4] использование оконных функций дает доступ к нескольким строкам сразу, например, вычисление скользящего среднего по трем строкам. В более сложных случаях в РБД временные ряды можно хранить в виде больших двоичных объектов (BLOB), а подготовку данных к выполнению запроса, так называемое, "причесывание" осуществлять при помощи функций, определяемых пользователем.

Для работы с временными рядами может быть использована NoSQL технология. Примером является OpenTSDB, которая представляет собой набор независимых демонов, предоставляющих доступ в HBase, и командных утилит [5]. Схема данных, применяемая в OpenTSDB, определяет, по сути, стандарт представления временных рядов в Big Data [6]. В связи с развитием публичных облачных платформ, следует уделить внимание облачной службе данных NoSQL от Microsoft, а именно Cassandra для Azure CosmosDB. Cassandra является базой данных с открытым исходным кодом и популярным решением для хранения данных временных рядов. Cassandra не является колоночной базой данных в привычном ее понимании. Выглядит она больше как строчная, но в каждой строке может быть разное количество столбцов, за счет чего легко организовать колоночное представление. Технически Cassandra позволяет хранить до 2 миллиардов столбцов (отсчетов для временного ряда) в одной строке, или при необходимости использовать группировку данных в различных строках. Согласно исследованию [6], выбор Cassandra для NoSQL решений в том случае, когда скорость поступления данных является определяющим фактором, является предпочтительным по сравнению с РБД.

Несмотря на то, что работать с временными рядами позволяют и РДБ и NoSQL БД, во многих случаях использование их оказывается неэффективным. Универсальные СУБД обладают большими возможностями для большого круга задач, хранение данных, зависящих от времени, для них одна из таких задач, и не основная. В связи с этим для хранения и обработки временных рядов были созданы СУБД, использующие специализированные алгоритмы БД. Такие БД оптимизированы для обработки массивов чисел, индексированных по времени. На рис.1 показан график роста популярности TSDB из-за их способности эффективно хранить данные и обеспечивать доступ к большим объемам данных.

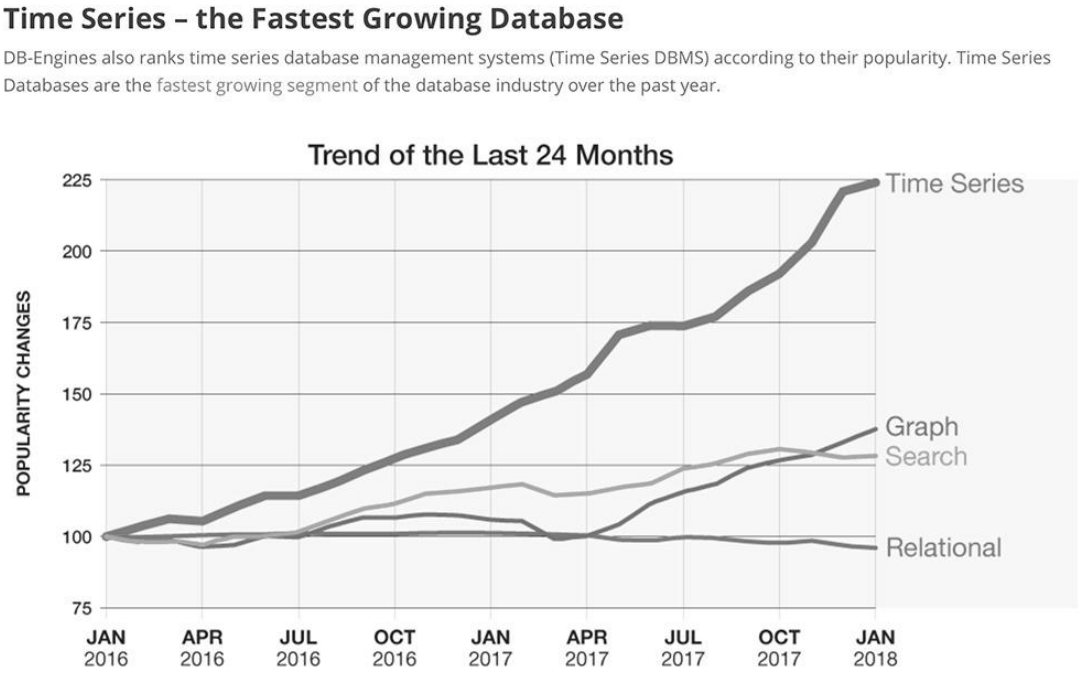



*Рис. 1- Рост популярности баз данных временных рядов*

Базы данных временных рядов помимо возможности создания, чтения, обновления и удаления ВР поддерживают ряд основных вычислений, которые работают на ряды в целом, например, умножая, складывая или иным образом комбинируя различные ВР в новый временной ряд. Они могут также фильтровать по произвольным образцам, при этом значения одного ряда могут являться фильтром для другого. Простой синтаксис является залогом привлекательности специальных баз данных временных рядов [7].

Среди специальных TSDB преобладают продукты, основанные на свободной лицензии: Open TSDB, InfluxDB, Geras, Druid и пр. Некоторые из них «заточены» под относительно узкий спектр задач. Так, YAWNDB и SiteWhere позиционируются как инструмент для специалистов в области веб-технологий.

## InfluxDB для хранения и обработки статистики посещения сайта

**Постановка задачи**. Для оценки возможностей применения TSDB выбрана задача сбора статистики посещения сайта и ее дальнейший анализ. Необходимо получить информацию по посещению пользователями сайта, находящегося в открытом доступе, и установить, например, в какие дни недели больше всего пользователей посещает сайт, в какое время суток меньше всего пользователей, и т.д. Также интересно определить среднее время нахождения пользователя на сайте за определенный период, количество пользователей и т.д..

**Выбор TSDB**. Для решения задачи выбрана InfluxDB – база данных временных рядов, разработанная InfluxData. Она распространяется под свободной лицензией, имеет открытый исходный код, и уже используется в enterprise крупными компаниями для мониторинга серверов, кластеров и т.д. Помимо наличия требований, характерных для всех TSDB, InfluxData обеспечивает:

- легкое развертывание (контейнер Docker);
- интеграцию с разнообразными системами, в том числе и собственными;
- собственный гибкий язык запросов Flux;
- отличную визуализацию данных, разнообразных метрик и т.д.;
- SDK для множества языков программирования (Go, Python, Node.js, Ruby).

**Терминология**. Термины, используемые в TSDB и InfluxDB:

- точка данных (`point`) – набор полей с одинаковой отметкой времени (строка в SQL);
- временная метка (`timestamp`) – дата и время, определённые в формате RFC3339;
- `Unix`-время – система описания моментов времени, принятая в `Unix` и других POSIX (набор стандартов, описывающих интерфейсы между операционной системой и прикладной программой) совместимых операционных системах, определяется как количество секунд, прошедших с полуночи 1 января 1970 года;
- тег (`tag`) – метаданные, определяющие данные в измерении, InfluxDB автоматически индексирует теги для ускорения запроса;
- поле (`field`) – данные (столбец в SQL), поля не индексируются.

Для решаемой задачи тег – имя сайта, поле – количество пользователей на сайте.

**Инструментарий**. Для решения поставленной задачи и взаимодействия с БД backend использует платформу Node.js и TypeScript (типизированный расширенный набор JavaScript, который компилируется в простой JavaScript). Клиентская часть приложения каждые 5 секунд информирует серверную часть о нахождении определенного клиента на сайте. Для реализации такого поведения использована технология WebSocket, которая позволяет установить персистентное (устойчивое) соединение между клиентом и сервером и обеспечить прямую передачу данных от клиента к серверу без необходимости клиенту делать каждый раз новый запрос к серверу. Информацию от клиентов сервер накапливает в буфере и каждые 10 секунд записывает ее в БД, при этом очищая буфер.

**Запись данных**. Для записи данных в БД используется Line Protocol – набор правил ручной (без использования плагинов/серверных агентов) записи данных в базу данных InfluxDB. Line Protocol представляет собой текстовую строку, разбитую на несколько концептуальных частей. На рис. 2 показан синтаксис Line Protocol для записи данных посещения сайта.

```
visitors,website=ezic.io amount=2000 1582254658889600200
          |      --------------- ---------           |
          |              |              |             |
          |              |              |             |
        +-----------+--------+-+---------+-+---------+
        |measurement|,tag_set| |field_set| |timestamp|
        +-----------+--------+-+---------+-+---------+
```

*Рис.2. Строка Line Protocol*

На рисунке: `measurement` – имя измерения; `tag_set` – теги, которые будут включены в точку данных (измерение и набор тегов разделены запятой без пробелов); первый пробел отделяет `measurement` и `tag_set` от значений; `field_set` – поля для точки данных (каждое поле указывается как пара ключ-значение со знаком = без пробелов); второй пробел отделяет набор полей от необязательной временной метки; `timestamp` – временная метка для точки данных во времени `Unix` с точностью до наносекунды.

**Язык запросов Flux**. Flux разработан InfluxDB, его синтаксис подобен Javascript. Flux запрос для аналитики посещения сайта показан на рис.3.

```
from(bucket:"xe6/websites_analytics")
  |> range(start:-1h)
  |> filter(fn:(r) =>
    r._measurment == "visitors" and
    r.website == "ezic.io"
  )
  |> aggregateWindow(every: 1m, fn: mean)
```

*Рис.3 – Запрос Flux: получить количество посетителей сайта*

В приведенном коде используется БД `xe6/websites_analytics`, анализируется посещение сайта `ezic.io` за последний час, фильтрация данных осуществляется по количеству пользователей сайта `visitors` и тегу `website = ezic.io`. С интервалом в 1 минуту вызывается функция `aggregateWindow` для того, чтобы объединить полученные данные и вычислить для них среднее значение.

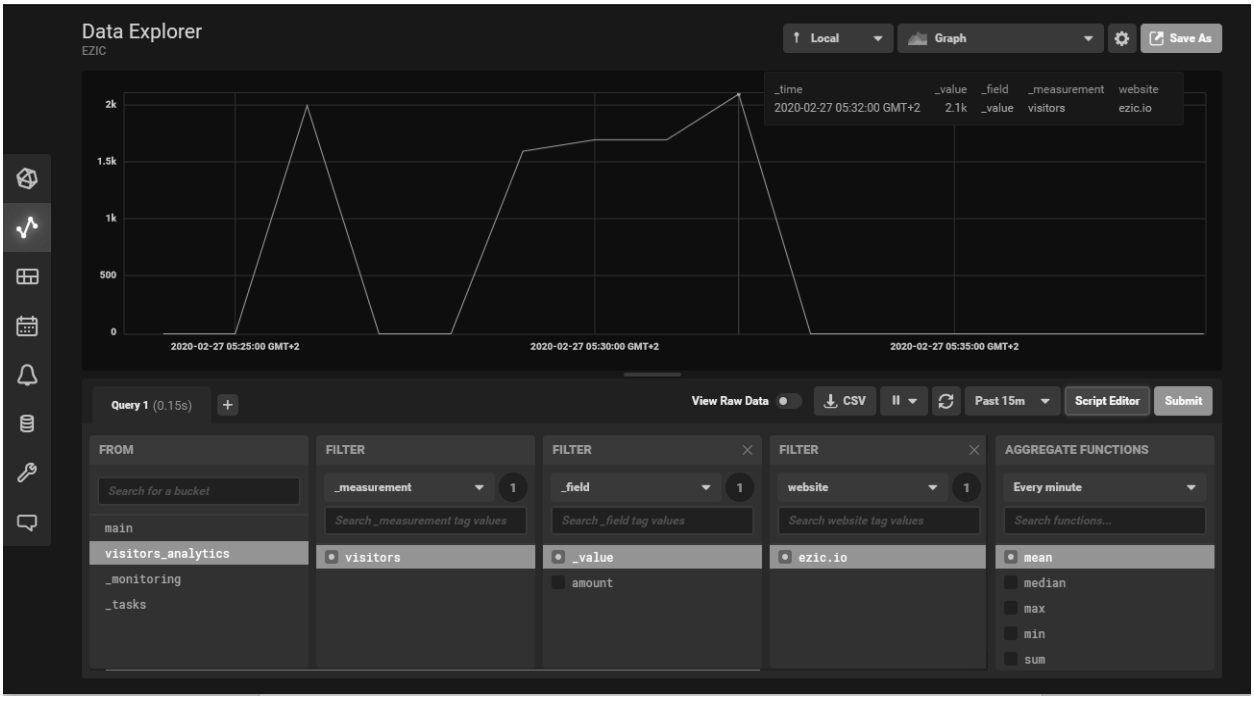



*Рис. 4 – Конструктор запросов InfluxDB*

**Конструктор запросов InfluxDB**.

На рис.4. представлен вид конструктора запросов, которым InfluxDB обеспечивает пользователя. Конструктор использует элементы веб интерфейса, работать с ним просто. Запрос, скомпонованный так, как показано на рис.4, будет скомпилирован, вид готового запроса представлен на рис.5.

```
from(bucket: "visitors_analytics")
|> range(start: v.timeRangeStart, stop: v.timeRangeStop)
|> filter(fn: (r) => r._measurement == "visitors")
|> filter(fn: (r) => r._field == "_value")
|> filter(fn: (r) => r.website == "ezic.io")
|> aggregateWindow(every: 1m, fn: mean)
|> yield(name: "mean")
```

*Рис. 5 – Запрос в InfluxDB*

Для объединения операций Flux использует конвейерные (`pipe`) операторы перенаправления (**`|>`**). После каждой функции или операции Flux возвращает таблицу или коллекцию таблиц, содержащих данные. Оператор **`|>`** передает эти таблицы в следующую функцию или операцию, где они проходят дальнейшую обработку. Flux структурирует все данные в таблицах. Когда данные передаются из источников данных, Flux форматирует их через запятую (CSV), как аннотированные значения, представляющие таблицы. Такой синтаксис позволяет легко связывать воедино разнообразные функции для создания сложных запросов.

## Заключение

В работе описаны некоторые возможности использования базы данных временных рядов InfluxDB. На основании анализа принципов TSDB и работы, проведенной для решения простой задачи хранения и обработки статистики посещения сайта, сделаны выводы о целесообразности применения технологии Time Series Database. Оно оправдано в том случае, когда необходимо стабильно получать существенное количество данных на временной шкале и иметь возможность их анализировать различными гибкими методами, а так же иметь доступ к данным в конкретный момент времени. Использование InfluxDB для этих целей предпочтительно. Однако, учитывая то, что объем данных, с которым работают TSDB, огромен, существенным недостатком их использования является сложность записи данных в БД. Для этих целей целесообразно применение third-party расширений или программ-агентов. Существует большое количество расширений и программ, которые можно настроить под сбор разнообразных метрик/данных на временной шкале и интегрировать в любую систему.

Для решения простых задач, в которых данные зависят от времени, а интервал, с которым они записываются в БД, не маленький, а также не требуется сложной аналитической обработки этих данных или прогнозирования, могут быть использованы РБД или NoSQL БД. Однако, у таких БД низкая производительность, отсутствует удобная и эффективная выборка данных в контексте временных рядов, сложная поддержка и администрирование серверной инфраструктуры, отсутствуют готовые решения для сбора большого количества данных в реальном времени.

Для дальнейшего исследования TSDB планируется выбрать более сложную задачу для решения, поскольку хранение и обработка статистики посещения сайта простая задача, она оперирует небольшим объемом данных и не использует все возможности TSDB для анализа данных и прогнозирования.

ЛИТЕРАТУРА

1. Когаловский М.Р. Энциклопедия технологий баз данных. М.: Финансы и статистика, 2002. 800 с. ISBN 5-279-02276-4. 9.
2. Временной ряд. [Электронный ресурс] Режим доступа https://ru.wikipedia.org/wiki/%D0%92%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B9_%D1%80%D1%8F%D0%B4
3. Кирилл Косолапов. Временные ряды в прогнозировании спроса, нагрузки на КЦ, товарных рекомендациях и поиске аномалий. [Электронный ресурс] Режим доступа https://habr.com/ru/post/477206/
4. Документация PostgreSQL 10.2. [Электронный ресурс] Режим доступа https://postgrespro.ru/docs/postgresql/10/high-availability
5. The scalable Time Series database. [Электронный ресурс] Режим доступа http://opentsdb.net/
6. Намиот Д.Е. Базы данных временных рядов и средства обработки. [Электронный ресурс] Режим доступа: http://linkstore.ru/articles/time-series0.pdf
7. Что такое базы данных временных рядов (time series database). [Электронный ресурс] Режим доступа: https://www.xelent.ru/blog/chto-takoe-bazyi-dannyih-vremennyih-ryadov--time-series-database-/
8. The Definitive Guide To InfluxDB In 2019. ). [Электронный ресурс] Режим доступа: https://devconnected.com/the-definitive-guide-to-influxdb-in-2019/

**КРИВОГУЗОВ Максим Андреевич** – студент факультета компьютерных наук Харьковского национального университета имени В. Н. Каразина.

Научные интересы:

– *разработка программного обеспечения;*

– *исследование и расширение Web технологий*.

**ЛАЗУРИК Валентина Михайловна** – старший преподаватель кафедры искусственного интеллекта и программного обеспечения факультета компьютерных наук Харьковского национального университета имени В. Н. Каразина.

Научные интересы:

– *разработка компьютерных систем для моделирования процессов в радиационных технологиях;*

– *организация баз данных.*

**УДК 539.3**

КРЮТЧЕНКО Д. В., МОСКАЛЕНКО Р.П., УСАТОВА О.О.

# КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ВИМУШЕНИХ КОЛИВАНЬ РІДИНИ У ГОРИЗОНТАЛЬНОМУ ЦИЛІНДРИЧНОМУ РЕЗЕРВУАРІ, ЧАСТКОВО ЗАПОВНЕНОМУ РІДИНОЮ

**Вступ**

Явища плескань рідини в частково заповнених резервуарах притаманні широкому колу конструкцій, які використовуються в нафтохімічній, аерокосмічній промисловості та при транспортуванні. Велику кількість публікацій в цьому напрямку присвячено дослідженню коливань рідини в жорстких оболонках обертання. При цьому вільна поверхня має форму кола. Це дає змогу звести тривимірні крайові задачі, що розглядаються, до одновимірних. Інтегрування здійснюється за меридіаном оболонки. Але коливання рідини в горизонтально розташованих циліндричних оболонках досліджені недостатньо. Ці тривимірні задачі в загальному випадку не допускають зменшень вимірності. Але в зв'язку з широким використанням таких оболонок на практиці, особливо при транспортуванні, стає актуальним питання визначення власних форм коливань таких оболонок та дослідження вимушених коливань, пов'язаних з сейсмічними та імпульсними навантаженнями.

**Аналіз досліджень і публікацій**

Для аналізу вільних та вимушених коливань рідин в жорстких резервуарах в останні роки широко використовуються числові методи. Серед них зазначимо методи скінченних та граничних елементів. В роботах [1-3] за допомогою методів скінченних та граничних елементів досліджені власні та вимушені коливання рідини в пружних та жорстких оболонках обертання. В [4] проведено числове дослідження параметричних коливань рідини в призматичному резервуарі з перегородками. Зазначимо, що в більшості досліджень з вимушених коливань рідини в резервуарах розглядаються або вертикальні кругові циліндричні оболонки, або призматичні резервуари. Значно менша кількість досліджень присвячена вивченню коливань в горизонтальних, частково заповнених рідиною оболонках. Серед цих праць відмітимо роботу [5].

**Формулювання задачі**

Розглядається горизонтальний циліндричний резервуар з радіусом $R$ та довжиною $L$, частково заповнений ідеальною нестисливою рідиною, Рис.1. Вважається, що рух рідини внаслідок прикладених зовнішніх навантажень є безвихровим.

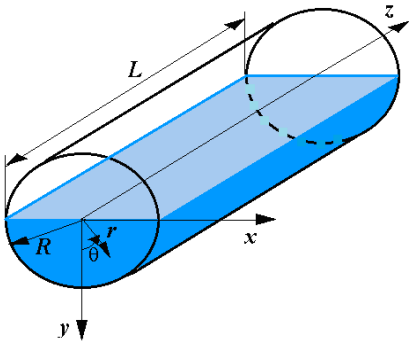



*Рис. 1 Горизонтальний циліндричний резервуар, частково заповнений рідиною*

Введемо потенціал швидкостей $\varphi(x,y,z,t)$, такий що $\mathbf{V} = \operatorname{grad}\varphi$. У зазначених умовах існує потенціал швидкостей, який задовольняє рівнянню Лапласа в циліндричних координатах

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\varphi}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2\varphi}{\partial\theta^2}+\frac{\partial^2\varphi}{\partial z^2}=0 \qquad (1)$$

в області $\Omega=\{r<R, 0<z<L, -\pi/2<\theta<\pi/2\}$, тобто вражаємо, резервуар заповнений наполовину. Граничні умови для рівняння (1) є такими:

на бічних стінках оболонки $S_w = \{r = R, 0 < z < L, -\pi/2 < \theta < \pi/2\}$

$$\frac{\partial \varphi}{\partial r} = 0, \tag{2}$$

на днищах $S_w = \{r < R, z = 0, z = L, -\pi/2 < \theta < \pi/2\}$

$$\frac{\partial \varphi}{\partial z} = 0, \tag{3}$$

на вільній поверхні $S_w = \{r < R, 0 < z < L, \theta = \pm\pi/2\}$ мають виконуватись динамічна та кінематична умови

$$-\frac{\partial \varphi}{\partial y} = \frac{\partial \zeta}{\partial t}, \quad \frac{\partial \varphi}{\partial y} = \frac{\partial \varphi}{\partial r}\frac{\partial r}{\partial y} + \frac{\partial \varphi}{\partial \theta}\frac{\partial \theta}{\partial y} = \sin\theta \frac{\partial \varphi}{\partial r} + \frac{\cos\theta}{r}\frac{\partial \varphi}{\partial \theta}\bigg|_{\theta=\pm\pi/2} = \frac{\partial \varphi}{\partial r}, \quad \frac{\partial \varphi}{\partial t} - g\zeta = 0. \tag{4}$$

Тут $\zeta$ - функція, що описує положення вільної поверхні.

Тиск рідини за дії горизонтального та вертикального навантажень з прискореннями $a_x(t), a_z(t)$ визначається за допомогою лінеаризованого інтегралу Коші – Лагранжа

$$p - p_0 = -\rho_l \left[ \frac{\partial \varphi}{\partial t} + \left(a_y(t) - g\right)\zeta + a_z(t) z \right]. \tag{5}$$

З кінематичної та динамічної умов маємо після диференціювання другої рівності в (4) отримаємо

$$\frac{\partial^2 \varphi}{\partial t^2} + g \frac{\partial \varphi}{\partial r}\bigg|_{S_0} = 0. \tag{6}$$

Таким чином, сформульовано крайову задачу (1)-(3), (6) для визначення невідомого потенціалу швидкостей. До цих умов додаємо умову розв'язності Неймана

$$\int_{S_0} \frac{\partial \varphi}{\partial n} dS_0 = 0 \tag{7}$$

Завдяки симетрії контейнеру відносно площин $z = L/2$ та $\theta = 0$ вираз для шуканого потенціалу має враховувати цю симетрію. Надалі використовуємо метод розділення змінних, а саме зобразимо функцію $\varphi$ у вигляді

$$\varphi(r, \theta, z, t) = Q(t)F(r)\Theta(\theta)Z(z). \tag{8}$$

Внаслідок симетрії розв'язку відносно площини $\theta = 0$ здобудемо вираз

$$\Theta(\theta) = \cos n\theta, \quad n = 0,1,... \tag{9}$$

З рівняння (1) одержимо

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial F}{\partial r}\right)Z(z)\cos n\theta - \frac{n^2}{r^2}F(r)Z(z)\cos n\theta + \frac{\partial^2 Z}{\partial z^2}F(r)\cos n\theta = 0 \tag{10}$$

З рівняння (10) знайдемо

$$\frac{1}{rR}\frac{d}{dr}\left(r\frac{dF}{dr}\right) - \frac{n^2}{r^2} + \frac{1}{Z}\frac{d^2 Z}{dz^2} = 0, \quad \frac{1}{rF}\frac{d}{dr}\left(r\frac{dF}{dr}\right) - \frac{n^2}{r^2} = -\frac{1}{Z}\frac{d^2 Z}{dz^2} = \beta. \tag{11}$$

З (11) визначимо таке диференційне рівняння для знаходження функції $Z(z)$

$$Z'' + \beta Z = 0. \tag{12}$$

Для того, щоб були виконані умови симетрії, потребуємо, щоб $\beta = \alpha^2$. Тоді частинний розв'язок рівняння (12) буде таким $Z = \cos \alpha z$. Для забезпечення симетрії відносно площини $z = L/2$ встановлюємо, що $\cos 0 = \cos \alpha L = 1$, тобто $\alpha L = 2\pi k$. Звідси

$$\alpha_k = 2\pi k / L, \quad Z_k = \cos \alpha_k z \tag{13}$$

При такому виборі сталої розділення виконується умова непротікання (3) на днищах оболонки. Для знаходження функції $R(r)$ з (11) знайдемо диференційне рівняння

$$\frac{1}{rF}\frac{d}{dr}\left(r\frac{dF}{dr}\right) - \frac{n^2}{r^2} = \alpha_k^2. \tag{14}$$

Рівняння (14) є модифікованим рівнянням Бесселя, а саме

$$r^2\frac{d^2F}{dr^2}+r\frac{dF}{dr}-\left(n^2+\alpha_k^2r^2\right)F=0. \tag{15}$$

Для надання рівнянню (15) стандартного вигляду робимо заміну змінної $\alpha_k r=\rho$. Здобудемо

$$\rho^2\frac{d^2F}{d\rho^2}+\rho\frac{dF}{d\rho}-\left(n^2+\rho^2\right)F=0. \tag{16}$$

Обмежимось випадком аксіально-симетричних коливань, тобто вважаємо, що $n=0$. Загальним розв'язком рівняння (16) буде лінійна комбінація функцій Інфельда та Макдональдса

$$F(\rho)=C_1I_0(\rho)+C_2K_0(\rho),\quad F_k(\alpha_k r)=C_1I_0(\alpha_k r)+C_2K_0(\alpha_k r). \tag{17}$$

Задовольняючи крайову умову (2), знайдемо

$$F_k(\alpha_k r)=-\alpha_k I_0(\alpha_k r)K_1(\alpha_k R)-\alpha_k I_1(\alpha_k R)K_0(\alpha_k r). \tag{18}$$

Таким чином, отримані форми аксіально-симетричні форми коливань рідини в напівзаповненому горизонтальному резервуарі

$$\varphi_k(r,z,t)=\dot{c}_k(t)F_k(\alpha_k r)\cos(\alpha_k z). \tag{19}$$

Згідно з кінематичною умовою для вільної поверхні одержимо

$$\zeta_k(r,z,t)=c_k(t)\alpha_k F_k'(\alpha_k r)\cos(\alpha_k z). \tag{20}$$

Далі використовуємо такі розкладення

$$\varphi(r,z,t)=\sum_{k=1}^{\infty}\dot{c}_k(t)\varphi_k(r,z),\qquad \zeta(r,z,t)=\sum_{k=1}^{\infty}c_k(t)\zeta_k(r,z) \tag{21}$$

Припустимо, що $c_k(t)=\exp(i\omega t)C_k$. Тоді з врахуванням (20),(21) та умови (5) маємо

$$-\omega^2\sum_{k=1}^{\infty}C_k\varphi_k(r,z)+g\sum_{k=1}^{\infty}C_k\zeta_k(r,z)=0. \tag{22}$$

З умови (22), враховуючи ортогональність власних форм коливань [1-2], здобудемо частоти коливань рідини в жорсткому горизонтальному циліндричному резервуарі.

$$\frac{\omega_k^2}{g}=\frac{\alpha_k\int_0^R rF_k'(\alpha_k r)F_k(\alpha_k r)dr}{\int_0^R rF_k(\alpha_k r)F_k(\alpha_k r)dr} \tag{24}$$

На рис.2 зображені дві перші аксіально-симетричні форми коливань вільної поверхні в горизонтальному циліндричному резервуарі при $R=1, L=2$.

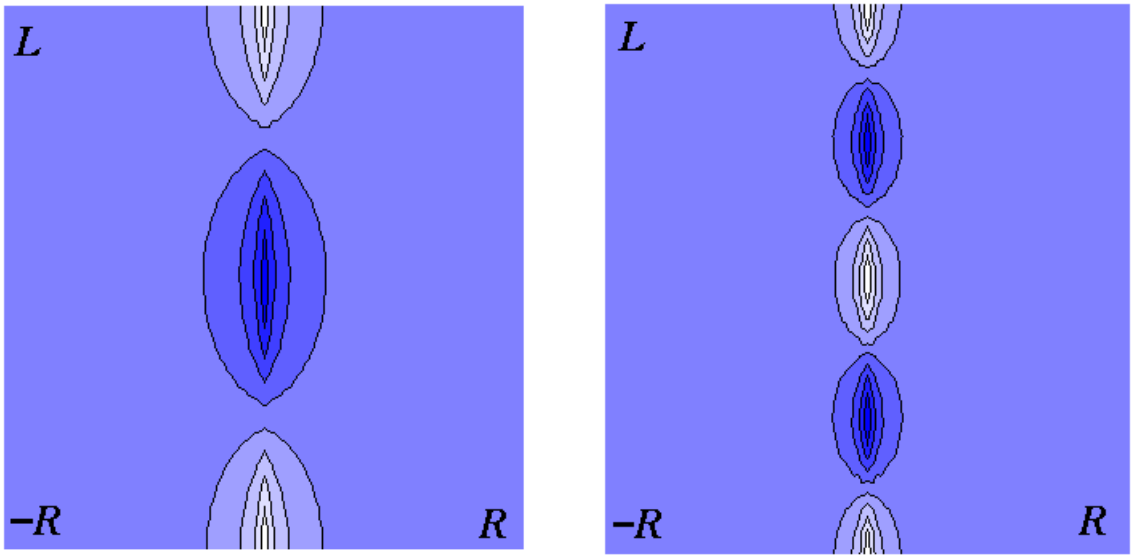



*Рис.2. Перші форми коливань вільної поверхні рідини, $n=0$.*

Цим формам коливань відповідають частоти $\omega_1$ = 12.32828500Гц, $\omega_2$ = 17.43609394 Гц.

Вимушені коливання розглянуті при дії горизонтальних гармонічних, імпульсних та сейсмічних зовнішніх навантажень [6].

**Висновки**

Отримані частоти та форми коливань для горизонтального циліндричного резервуара, частково заповненого рідиною. Форми коливань використано як базисні функції при дослідженні вимушених коливань рідини в цих резервуарах. Досліджено поведінку рідини за дії інтенсивних зовнішніх силових впливів.

ЛІТЕРАТУРА

1. Gnitko, V., Naumemko, Y., Strelnikova E. Low frequency sloshing analysis of cylindrical containers with flat and conical baffles, //International Journal of Applied Mechanics and Engineering 22 (4) ,pp.867-881, 2017.
2. Шувалова Ю.С., Крютченко Д.В., Стрельникова Е.А. Интегральные уравнения в задаче о свободных и вынужденных колебаниях жидкости в жестких резервуарах. Вісник Херсонського національного технічного університету, випуск 3, с. 455-459.
3. Еселева Е.В. Собственные колебания сосудов высокого давления при взаимодействии с жидкостью / Е.В. Еселева, В.И. Гнитько, Е.А. Стрельникова // Пробл. машиностроения. –2006. – Т. 9, №1, – С.105 - 118.
4. V. S. Sanapala, M. Rajkumar, K. Velusamy, and B. S. V. Patnaik, Numerical simulation of parametric liquid sloshing in a horizontally baffled rectangular container, *Journal of Fluids and Structures*, vol. 76, pp. 229–250, 2018.
5. Amabili, M. "Free vibration of partially filled, horizontal cylindrical shells," *Journal of Sound and Vibration*, vol. 191, no. 5, pp. 757–780, 1996.
6. Дегтярьов К.Г., Крютченко Д.В., Москаленко Р.П., Пальчиков Р.Г. Комп'ютерне моделювання вимушених коливань елементів конструкцій, що взаємодіють з рідиною, за умови дії гармонічних, імпульсних та сейсмічних впливів Вісник Харківського національного університету імені ВН Каразіна, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління», Т. 43, с.20-29, 2019

**КРЮТЧЕНКО Денис Володимирович** – аспірант, Інститут проблем машинобудування ім. А. М. Підгорного НАН України, вул. Пожарського, 2/10, м. Харків, Україна, 61046; e-mail: wollydenis@gmail.com; ORCID: 0000-0002-6804-6991.

Наукові інтереси:

*– математичне моделювання коливань рідини в резервуарах під дією сейсмічного та імпульсного навантаження.*

**МОСКАЛЕНКО Роман Павлович** – аспірант Харківского національного університету імені В. Н. Каразіна, факультет комп'ютерних наук. телефон: 050 598 7682; e-mail: rmphd2016@gmail.com; orcid: 0000-0002-5167-2793.

Наукові інтереси*:*

*– математичне моделювання резонансних коливань, дослідження довговічності елементів конструкцій з тріщинами.*

**УСАТОВА Ольга Александровна,** – аспирант, Институт проблем машиностроения им. А. Н. Подгорного НАНУ, ул. Пожарского, 2/10, Харьков, 61046, Украина, ORCID 0000-0001-1267-2723.

Наукові інтереси:

*– математичне моделювання коливань рідини в резервуарах, течія рідини в нанотрубках.*

**УДК 539.3**

КРЮТЧЕНКО Д.В.

# КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ВИМУШЕНИХ КОЛИВАНЬ РІДИНИ В ПРИЗМАТИЧНОМУ РЕЗЕРВУАРІ З ВЕРТИКАЛЬНИМИ ПЕРЕГОРОДКАМИ

**Вступ**

У всьому світі виникають такі явища, як землетруси, теракти, аварії. Тому при проектуванні резервуарів для зберігання легкозаймистих та отруйних речовин, важливо враховувати, які сили можуть діяти на резервуар та рідину, яка в ньому зберігається. Виплескування легкозаймистих та токсичних речовин при дії раптово прикладеного зовнішнього навантаження може призвести до техногенної катастрофи. Отже демпфування коливань вільної поверхні рідини є актуальним науково-технічним завданням. Зазвичай, у якості демпферів слугують перегородки різного типу. У даній роботі розглядаються вимушені коливання рідини в призматичному резервуарі з вертикальними перегородками.

**Аналіз досліджень і публікацій**

Для зменшення амплітуди коливань рідини були запропоновані та досліджені різні методи, наприклад, встановлення перегородок в резервуарах, частково заповнених рідиною [1-6]. У більшості робіт зроблено акцент на горизонтальні перегородки у призматичних резервуарах. У роботі [3] запропоновано підхід до аналізу впливу конічних перегородок на частоти коливань вільної поверхні рідини. В роботах [1, 3] було визначено, що форма перегородки та її положення відіграє важливу роль при проектуванні резервуарів. У цих роботах розглядалися вільні коливання рідини в баках з перегородками при частковому заповненні рідиною.

**Формулювання задачі**

Розглядається призматичний резервуар з вертикальними хрестовими перегородками. Всі чотири відсіки резервуара заповнені ідеальною рідиною, Рис.1. Рівень заповнення, $h$ у всіх відсіках однаковий. Вважається, що рідина, яка знаходиться у резервуарі, є нестисливою та ідеальною, а її рух, викликаний дією зовнішнього навантаження, є безвихровим.

Вважаємо, що чотири відсіки резервуара частково заповнені ідеальною нестисливою рідиною, при чому рівень заповнення $h$ у всіх відсіках однаковий. Припускаємо, що рух рідини є безвихровим. Зауважимо, що якщо рух починається із стану спокою, то він залишається потенційним протягом всього наступного часу згідно з теоремою Томпсона [7]. Використовуємо припущення існування потенціалу швидкостей $\varphi(x,y,z,t)$ згідно з [5], тоді

$$V_x = \frac{\partial \varphi}{\partial x}, V_y = \frac{\partial \varphi}{\partial y}, V_z = \frac{\partial \varphi}{\partial z}.$$

У цих умовах існує потенціал швидкостей, який задовольняє рівнянню Лапласа

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} = 0.$$

Тиск рідини на поверхні резервуара, за дії горизонтального та вертикального прискорень, визначається за допомогою лінеаризованого інтегралу Коши – Лагранжа

$$p - p_0 = -\rho_l \left[ \frac{\partial \varphi}{\partial t} + \left(a_z(t) + g\right) z + a_x(t) x \right].$$

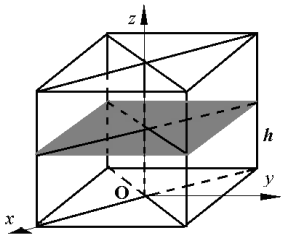



*Рис. 1 Призматичний резервуар з вертикальними перегородками*

Розглянуті коливання призматичного резервуара під дією гармонічного навантаження.

$$F = a_0 \cos\omega_0 t\,,$$

де $a_0$, $\omega_0$ - амплітуда та частота сили, що змущує.

Крім того, розглянуто імпульсне навантаження у вигляді $F = Q_0 a(t)$,
де $Q_0$ є заданим розподіленим тиском,

$$a(t) = \begin{cases} 1, & t < T, \\ 0, & t \geq T. \end{cases}$$

час $T$ є періодом дії імпульсного навантаження.

Сейсмічне навантаження моделювалось таким чином $F = Q_0 a_S(t)$,
де $Q_0$ є заданим розподіленим тиском,

$$a_S(t) = \begin{cases} Q_0 \sin(12\pi t/5)\cos(2\pi t), & t < T, \\ 0, & t \geq T, \end{cases}$$

час $T$ є періодом дії сейсмічного навантаження.

**Висновки**

Досліджено поведінку рідини в призматичному резервуарі під дією гармонічних, імпульсних та сейсмічних навантажень. Вертикальні перегородки впливають на резонансні частоти коливань вільної поверхні рідини. Коливання збільшують свою частоту. Цей метод дозволяє відбудуватися від небажаних частот при проектуванні призматичних резервуарів для зберігання рідини.

ЛІТЕРАТУРА

1. Gnitko V., Degtyariov K., Naumenko V., Strelnikova E. Coupled BEM and FEM analysis of fluid-structure interaction in dual compartment tanks. Int. Journal of Computational Methods and Experimental Measurements. 2018, vol. 6, no. 6, pp. 976–988.
2. Gavrilyuk I., Hermann M, Lukovsky I., Solodun O., Timokha A. Natural Sloshing Frequencies in Truncated Conical Tanks. Engineering Computations. 2008, vol. 25, no. 6, pp. 518–540.
3. Gnitko V., Naumemko Y., Strelnikova E. Low frequency sloshing analysis of cylindrical containers with flat and conical baffles. International Journal of Applied Mechanics and Engineering. 2017, no. 22 (4), pp. 867–881.
4. Gnitko V., Degtyarev K., Naumenko V., Strelnikova E. Reduced Boundary Element Method for Liquid Sloshing Analysis of Cylindrical and Conical Tanks with Baffles // Int. Journal of Electronic Engineering and Computer Sciences. – 2016. – V. 1(1). – P. 14 – 27.
5. Wang, J., Sun, S, «Study on liquid sloshing characteristics of a swaying rectangular tank with a rolling baffle», *J Eng Math* **119,** 23–41 (2019). https://doi.org/10.1007/s10665-019-10017
6. Шувалова Ю.С., Крютченко Д.В., Стрельникова Е.А.. Интегральные уравнения в задаче о свободных и вынужденных колебаниях жидкости в жестких резервуарах. Вісник Херсонського національного технічного університету, випуск 3, с. 455-459.
7. Ibrahim R.A. Liquid Sloshing Dynamics. Cambridge University Press. New York.2005.

**КРЮТЧЕНКО Денис Володимирович** – аспірант, Інститут проблем машинобудування ім. А. М. Підгорного НАН України, вул. Пожарського, 2/10, м. Харків, Україна, 61046; e-mail: wollydenis@gmail.com; ORCID: 0000-0002-6804-6991.

*Наукові інтереси– математичне моделювання коливань рідини в резервуарах під дією сейсмічного та імпульсного навантаження.*

**УДК 65.0(075.8)**

ЛАДОВЩИК Л.М., БЕРДНІКОВ А.Г.

# МОДЕЛЬ УПРАВЛІННЯ ЯКІСТЮ ІТ-ПРОЕКТУ

**Вступ**

Сьогодні, задля досягнення успіху в конкурентній боротьбі, необхідно забезпечити оптимальне поєднання налагоджених бізнес-процесів в структурах управління з динамічними і орієнтованими на кінцевий результат проектними підходами. Особливу актуальність при управлінні проектами в даний час відіграє саме менеджмент якості проектів, оскільки саме якісний продукт в умовах сьогоднішніх реалій здатен перемогти в конкурентній боротьбі.

Основою ефективного планування управління якістю є оптимальне співвідношення «ціна/якість». Відомо, що чим краще продукт, тим він дорожчий. Якість і прибуток зростають прямо пропорційно, оскільки підвищення першого призводить до конкурентоспроможності продукту на ринку програмного забезпечення. Саме тому з'являється попит і, відповідно, зростає прибуток. А оптимізація (зменшення) витрат відбувається за допомогою вмілого використання інструментів контролю якості.

**Математичні моделі аналізу якості**

Управління якістю проекту вимагає системного підходу, реалізація якого в сучасній практиці здійснюється у вигляді створення стандартизованих систем менеджменту якості, що представляють собою сукупність документованих методик і засобів планування, забезпечення та контролю якості, що виконуються спеціально призначеними структурними одиницями організації (підприємства або проекту).

На одному з перших етапів – плануванні – проводиться збір та аналіз даних, які подаються на вході проекту. Американське керівництво з управління проектами PMBOK виокремлює два найуживаніших метода аналізу даних: порівняльний аналіз витрат і прибутків (cost-benefit analysis) та цінність і вартість якості (cost of quality).

Аналіз витрат та прибутків достатньо поширений метод. Він застосовується для оцінювання сильних та слабких сторін альтернатив з метою визначити найкращий можливий сценарій. Такий порівняльний аналіз допомагає менеджеру проекту визначити чи результативні дії щодо планування в області якості.

Математично різницева модель аналізу витрат та прибутків може бути представлена у вигляді наступної формули:

$$V_i = \sum_l^r b_{ij} - \sum_k^s c_{ik}, \quad (1)$$

Аналогічно может бути представлена й моделі відношення прибутків та витрат:

$$V_i = \frac{\sum_l^r b_{ij}}{\sum_k^s c_{ik}}, \quad (2)$$

де $b$ - прибутки,

$c$ - витрати за певні періоди виконання проекту.

Наразі частіше використовують математичні моделі, що дозволяють «звести» рівномірні фінансові потоки до одного моменту часу шляхом так званого дисконтування:

$$PV(B-C)_{iT} = \sum_{t=1}^{T} \frac{B_{it}}{(1+I)^t} - \sum_{t=1}^{T} \frac{C_{it}}{(1+I)^t}, \quad (3)$$

$$PV(B/C)_{iT} = (\sum_{t=1}^{T} \frac{B_{it}}{(1+I)^t})/(\sum_{t=1}^{T} \frac{C_{it}}{(1+I)^t}), \quad (4)$$

де $I$ - значення процентної ставки;

$B_{it}$ - сумарний прибуток за певний період $t$ для варіанту $i$;

$C_{it}$ - сумарні витрати за період $t$ для варіанту $i$;

Наступним методом аналізу даних є аналіз цінності та вартості якості. Цей метод є розширеним застосуванням попереднього. Основним «розширенням» є аналіз ланцюгів створення цінності та вартості.

В першу чергу при аналізі економіки якості відбуваються процеси зв'язку в ланцюг цінностей та витрат користувача. Однак очікувані виробником цінність та вартість можуть не співпадати з очікуваннями споживача.

Першочергова задача - досягти цільових величин з найменшими загальними витратами, зокрема, шляхом мінімізації витрат на неякісну продукцію. Результат та ефективність цілей повинні бути розглянуті окремо для кожного процесу.

В сучасних ІТ компаніях планування якості програмного продукту відбувається на старті проекту при зборі вимог до програми. На організаційному рівні визначаються рамки процесів та стандартів, що приведуть до високої якості, а саме:

- специфікація;
- огляд програмного коду (code review);
- стандарти написання коду (зазвичай визначаються політикою компанії).

На проектному рівні визначається застосування необхідних процедур та заходів на проекті, наприклад таких як ретроспектива (своєрідний «розбір польотів»), а також визначається план відстеження якості на проекті (метрики).

**Контроль якості ІТ-проекті**

Контроль якості повинен здійснюватися напротязі всього проекту для того, щоб за допомогою достовірних даних формально продемонструвати дотримання критеріїв приймання спонсора та замовника.

На ІТ проектах контроль якості продукту зводиться до різних способів тестування. Тестування - це організоване дослідження, що проводиться за певним планом (test cases), з метою отримати об'єктивну інформацію щодо якості продукту. Тестування призначене для виявлення помилок (багів) та інших проблем, пов'язаних програмним продуктом.

А ось для контроля якості проекту існують окремі інструменти. Для збору даних використовують контрольні листки, контрольні списки, вибірковий контроль, анкети та опитування.

Для аналізу та відображення даних використовуються наступні методи, аналіз виконання, аналіз першопричини, діаграми причинно-наслідкових зв'язку, контрольні карти, гістограми, діаграма розкиду.

Діаграма Ісікави є найзручнішою на наш погляд, при необхідності видалити не «симптоми», а першопричини проблеми, оскільки з її допомогою всі причини стають наглядними. Її побудова зазвичай пов'язана зі складнощами. Однак ця діаграма є досить ефективною.

Розглянемо побудований приклад цієї діаграми на основі проблеми пов'язаної з розробкою програмного продукту в ІТ-проекті, а саме: затримка демо-релізів та зрив строків.

Приклад побудованої діаграми приведений нижче, на рис. 1.

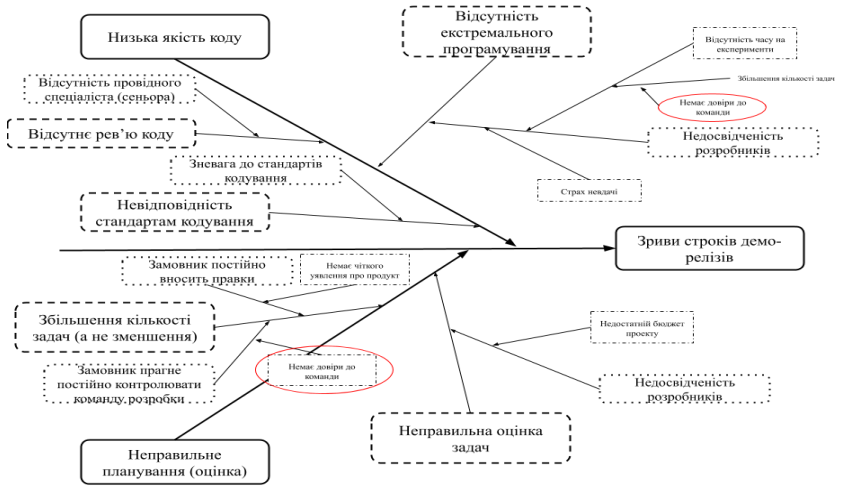



*Рис. 1 Схема побудованої діаграми Ісікави*

**Дослідження моделі**

Ми провели ряд математичних досліджень, в яких змінювали спеціалістів, їхню кількість та заробітну плату відповідно.

Математичні розрахунки базувалися на моделі відношення прибутків та витрат (формула 2). До сумарних витрат *c* входили заробітна плата спеціалістів та накладні витрати, що складають приблизно 15% від собівартості програмного продукту.

Результати досліджень відображені на графіку (Рис. 2), а саме: залежність сумарного коефіцієнту (заробітна плата за людино-годину на проекті від деякого показника якості проекту (маржинальності).

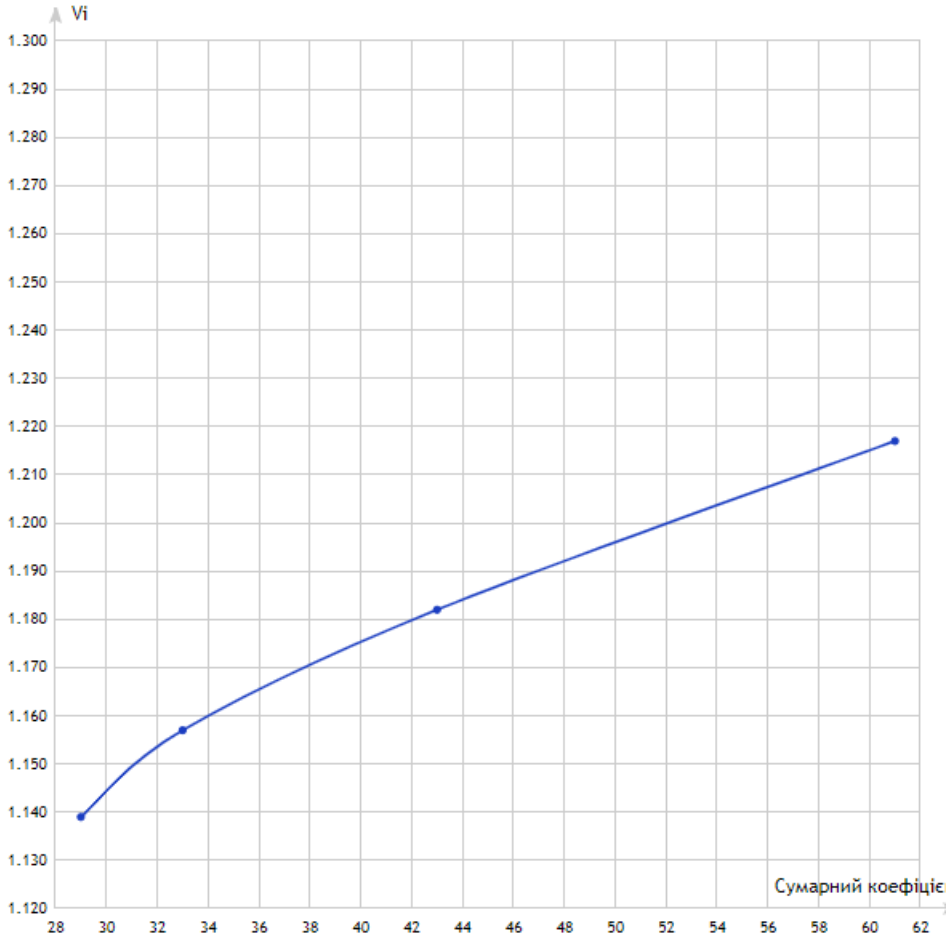



*Рис. 2 Залежність якості проекту (маржинальності) від сумарного коефіцієнту*

Дослідження показали, що при збільшенні маржинальності проекту, проектний менеджер може наймати кваліфікованішу команду. А якщо маржинальність залишається сталою, то розробляти проект надалі невигідно, оскільки витрати в залежності від кваліфікації команди ростуть, а прибуток залишається сталим.

**Висновок**

Однією з ефективних та достатньо простих моделей планування та контролю якості на проекті є модель відношення прибутків та витрат. Він застосовується для оцінювання сильних та слабких сторін альтернатив з метою визначити найкращий можливий сценарій. Такий порівняльний аналіз допомагає менеджеру проекту визначити чи результативні дії щодо планування в області якості.

На досліджуваному проекті, після проведення дослідження, було прийнято рішення про заміну декількох некваліфікованих спеціалістів одним кваліфікованим. Ми змогли залишитися в рамках впровадженого бюджету та строків.

ЛІТЕРАТУРА

1. Управление проектом. Основы проектного управления : учебник / кол. авт.; под ред. проф. М. Л. Разу. - М. : КНОРУС, 2006 - 768 с.;
2. Метод освоєного обсягу в управлінні проектами [Електронний ресурс]. – URL: http://jak-zrobutu.pp.ua/rizne/333-metod-osvoyenogo-obsyagu-v-upravlnn-proektami.html;
3. ANSI/PMI 99-001-2017 Руководство к своду знаний по управлению проектом (Руководство PMBOK) - Шестое издание / Институт управления проектами
4. Ильина, О. Н. Методология управления проектами. Становление, современное состояние и развитие / О.Н. Ильина. - М.: Вузовский учебник, Инфра-М, 2015. - 208 с.
5. Управление проектами / И.И. Мазур и др. - М.: Омега-Л, 2012. - 960 с.
6. Косов В.В. Методические рекомендации по оценке эффективности инвестиционных проектов / В.В. Косов, В.Н. Лившиц, А.Г. Шахназаров. – М.: Экономика, 2010.
7. Планирование качества. URL: https://studme.org/1240100821077/menedzhment/planirovanie_kachestva . Дата звернення: 23.04.2019
8. Контроль качества проекта. URL: https://studme.org/1188082021078/menedzhment/obespechenie_kachestva_proekta . Дата звернення: 27.04.2019
9. Качество проекта. Учебные материалы. URL: https://works.doklad.ru/view/EITxhfA0zkU.html. Дата звернення: 29.04.2019

**ЛАДОВЩИК Лада Максимівна** - студент; Харківський національний університет ім. В.Н. Каразіна; м. Харків, майдан. Свободи, 6, 61022; e-mail: lada.ladovshchyk@gmail.com.

Наукові інтереси:

*– організація менеджменту при управлінні проектами;*

*– проектування автоматизованих систем управління технологічними процесами.*

**БЕРДНІКОВ Анатолій Георгійович** – к. т. н., доцент; доцент кафедри автоматизації та комп'ютерно інтегрованих технологій; Харківський національний університет ім. В.Н. Каразіна; м. Харків, майдан. Свободи, 6, 61022; e-mail: a.berdnikov@karazin.ua .

Наукові інтереси:

*– оптимізація робіт по керуванню проектами.*

*– проектування автоматизованих систем управління;*

*– організація менеджменту при управлінні проектом.*

**УДК 539.12**

ЛАЗУРИК В.Т., ЛАЗУРИК В.М., ПОПОВ Г., САВАН С., ЗИМЕК З.

# ТЕСТИРОВАНИЕ МЕТОДА PFSEM НА БАЗЕ ГЛУБИННОГО РАСПРЕДЕЛЕНИЯ ДОЗЫ В КЛИНЕ ИЗ ДРЕВЕСИНЫ БЕРЕЗЫ

## Введение

Практическое использование дозиметрического клина в радиационных технологиях регламентируется международными стандартами, в которых определены процедуры измерения глубинного распределения дозы в клине, методы обработки проведенных измерений и интерпретация результатов обработки измерений [1,2]. В стандартах предлагается методом дозиметрического клина определять наиболее вероятную энергию $\boldsymbol{E_P}$ и среднюю энергию $\boldsymbol{E_{Av}}$ электронов в пучке. Эти характеристики электронов в пучке широко используются для контроля стабильности потока электронного излучения в радиационно-технологических процессах. Однако, характеристики электронного излучения, полученные с использованием стандартного дозиметрического клина, в ряде случаев, непригодны для описания пространственного распределения дозы в облучаемых объектах. Действительно, пространственное распределение дозы существенно зависит от плотности и эффективного атомного номера материалов облучаемых объектов. Как правило, при радиационной стерилизации медицинских изделий и продуктов питания, плотность и эффективный атомный номер материалов облучаемых объектов в несколько раз меньше, чем для алюминия, на основе которого изготовлен стандартный дозиметрический клин. Кроме того, для материалов с малой плотностью, пространственное распределение дозы в облучаемых объектах существенно зависит от углового распределения электронов в пучке. Поэтому, для дальнейшего развития радиационных технологий представляют интерес дозиметрические клинья, созданные на основе материалов с малой плотностью и эффективным атомным номером.

Современным методом определения характеристик электронного излучения на основе измерений глубинного распределения дозы в дозиметрическом клине является метод параметрической подгонки полуэмпирической модели (PFSEM-метод) [3-5]. Этот метод хорошо проверен в процессах радиационной стерилизации при использовании алюминиевого дозиметрического клина [6-8]. В связи с изложенным выше, актуальной научно-практической задачей является проведение тестовых испытаний PFSEM метода на основе результатов измерения дозы, в дозиметрических клиньях, созданных на основе материалов с малой плотностью и эффективным атомным номером. В этой работе проводится тестирование метода PFSEM на основе результатов измерения глубинного распределения дозы в специально изготовленном клине из древесины березы, которые были проведены на радиационно-стерилизационной линии Института Ядерной Химии и Технологий в Варшаве, Польша [9].

## Глубинное распределение дозы в клине из древесины березы

Измерения глубинных распределений дозы электронного излучения проводили с использованием дозиметрических пленок расположенных в специально изготовленном клине из древесины березы. Характеристики клина следующие:

плотность древесины березы – 0.536 г/см$^3$;

эффективный атомный номер – 6.28 и атомный вес – 11.86;

размеры – 12.5*46.5*45.5 см. и коэффициент наклона – 0.265.

Процедуры облучения и обработки дозиметрических пленок описаны в работе [6].

Результаты измерений представлены на Рис. 1. Величина дозы $D(x)$ нормирована на расчетное значение дозы $D(0)$ на поверхности клина, на которую падает электронный пучок. Глубина в клине $X$ указана в стандартных массовых единицах длины – g/cm$^2$. Вертикальными пунктирными линиями отмечены границы областей глубин, которые используются при

проведении процессов радиационной стерилизации. Отметка $X_1$ указывает область глубин, которая используется при одностороннем облучении и $X_2$ при двухстороннем облучении.

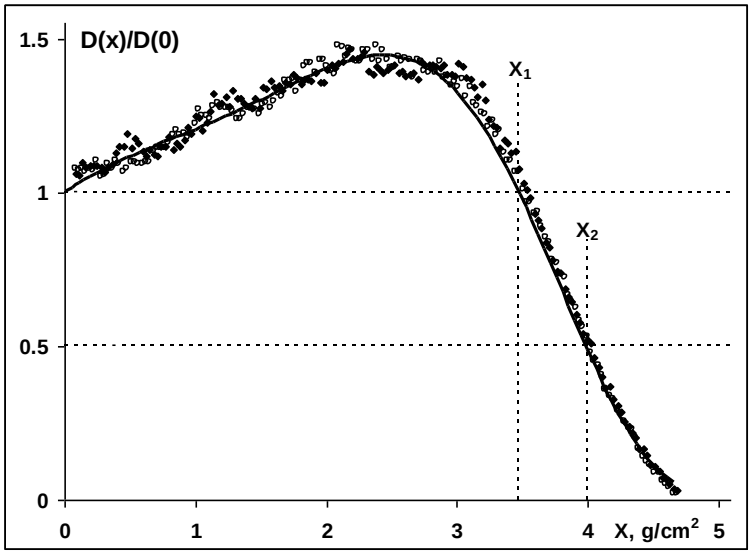



*Рис. 1 Глубинное распределение дозы в клине из древесины березы. Точки - результаты измерений, непрерывная кривая – аппроксимация измерений PFSEM-методом.*

При определении характеристик электронного излучения PFSEM методом не используются абсолютные значения измеренной величины дозы и, следовательно, погрешность метода определяется лишь величиной воспроизводимости результатов измерений. Оценка среднеквадратического отклонения воспроизводимости измерений $\Delta$ выполнена на основе расчета среднеквадратического отклонения значений величин дозы, измеренных на первой $D_1(x_i)$ и второй $D_2(x_i)$ дозиметрических пленках, для каждой из регистрируемых глубин $x_i$. Для областей глубин определяемых отметками $X_1$ и $X_2$, оценки среднеквадратичных погрешностей измерений близки и не превышают значений в 2%**.** Как видно из рис. 1, погрешность аппроксимации измерений PFSEM-методом сравнима с погрешностью проведенных измерений.

**Обработка результатов измерений PFSEM методом**

Проведена обработка результатов измерений величин дозы на двух дозиметрических пленках с использованием PFSEM метода. Для каждой из пленок были определены параметры модели электронного излучения радиационно-стерилизационной линии, на которой были проведены измерения. Параметры модели электронного излучения, полученные по данным для первой ***Ex₁,*** и второй ***Ex₂*** пленок представлены в Табл.1. Для сравнения, в строке ***Av_Ex*** таблицы представлены параметры модели электронного излучения, полученные для осредненных данных по этим двум измерениям.

*Табл.1Характеристики электронного излучения.*

| | ***E₁*** [MeV] | ***E₀*** [MeV] | ***X₀*** [g/cm$^2$] |
|---|---|---|---|
| ***Ex₁*** | 8.67 | 8.71 | 0.016 |
| ***Ex₂*** | 8.7 | 8.82 | 0.056 |
| ***Av_Ex*** | 8.69 | 8.77 | 0.037 |

Как видно из таблицы и Рис. 1, модельный параметр смещения ***X₀*** мал по сравнению с глубинами $X_1$ и $X_2$ использования распределений дозы в радиационных технологиях. Это свидетельствует о корректном выборе начальной точки при обработке глубинного распределения дозы на пленках. Отметим, что наблюдается малое отличие модельных энергетических параметров при однопараметрической подгонке ***E₁*** и двухпараметрической подгонке ***E₀*** данных полученных в этих двух измерениях.

Результаты тестовых испытаний PFSEM метода иллюстрируются на Рис.2. На рисунке точками показаны осредненные по двум измерениям значения распределения дозы. Сплошная кривая – аппроксимация осредненных значений распределения дозы с использованием PFSEM-метода и компьютерной визуализации полуэмпирической модели глубинного распределения энергии электронного пучка в веществе [5]. Гистограмма – расчет с использованием программного модуля ModeRTL системы RT-Office [10] в соответствии с определенными PFSEM методом значениями характеристик электронного излучения $\boldsymbol{E_0 = 8.77}$ и $\boldsymbol{X_0 = 0.037}$ (см. Табл. 1) в рамках двухпараметрической модели пучка электронов [7].

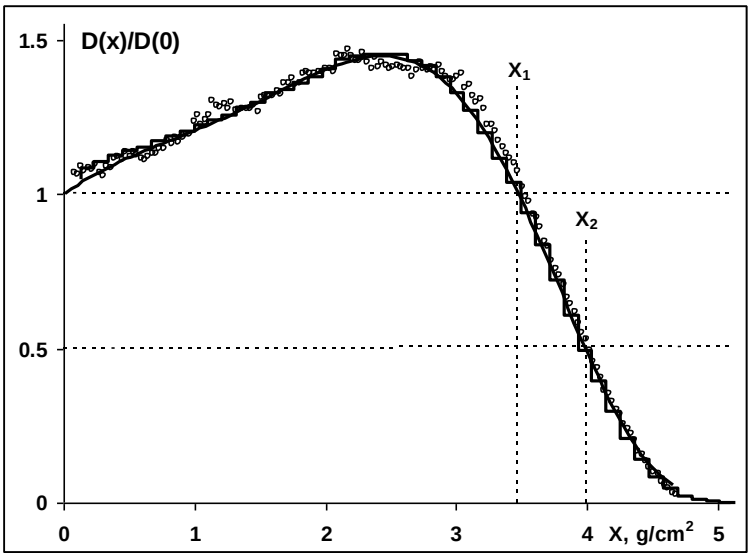



*Рис. 2 Сравнение результатов измерений и расчета с использованием PFSEM метода. Точки – осредненное глубинное распределение дозы по двум результатам измерений, непрерывная кривая – аппроксимация измерений PFSEM-методом, гистограмма - расчет с использованием ModeRTL модуля системы RT-Office.*

Как видно на Рис.2. наблюдается удовлетворительное согласие

- между осредненными значениями измерений и аппроксимацией этих значений с использованием PFSEM-метода и компьютерной визуализации полуэмпирической модели;

- между осредненными значениями измерений и результатами расчета, с использованием модуля ModeRTL, глубинного распределения дозы в клине из березовой древесины, облученной электронным пучком, параметры которого были определены PFSEM-методом в рамках двухпараметрической модели пучка электронов [7].

**Выводы**

Проведены измерения глубинного распределения дозы в специально изготовленном клине из древесины березы и оценена неопределенность результатов измерений. С использованием PFSEM метода, определены характеристики электронного пучка радиационно-технологической установки, на которой было проведено облучение. В детальной физической модели процесса облучения клина, методом Монте-Карло, рассчитано глубинное распределение дозы электронного излучения, для которого были проведены измерения.

Сравнение глубинных распределений дозы полученных в результате измерений и расчета методом Монте-Карло, на основе параметров электронного пучка определенных PFSEM методом, позволяет сделать следующие заключения:

- выполненная PFSEM методом аппроксимация результатов измерений глубинного распределения дозы в клине из древесины березы, имеет погрешность сравнимую с величиной неопределенности проведенных измерений;

- наблюдается удовлетворительное согласие результатов измерений и расчета методом Монте-Карло глубинного распределения дозы в клине из древесины березы облучаемом электронного излучения, характеристики которого были определены PFSEM методом.

Таким образом, показана возможность использования PFSEM метода для клиньев из материалов с малой плотностью и малыми эффективными атомными номерами.

ЛИТЕРАТУРА

1. ISO/ASTM Standard 51649. Practice for Dosimetry in an E-Beam Facility for Radiation Processing at Energies between 300 keV and 25 MeV // Annual Book of ASTM Standards. 2005, v.12.02.
2. Radiation Dosimetry: Electron Beams with Energies between 1 and 50 MeV // International Commission on Radiation Units and Measurements. 1984, report 35, Bethesda, MD, USA.
3. Lazurik V.T., Pochynok A.V. Dosimetry of electrons on the base of computer modeling the depth-dose distribution of irradiation // Journal of Kharkiv University. Mathematical modeling. Information technologies series. – 2010. – No.925. – P.114 – 122.
4. Pochynok A.V., Lazurik V.T., Sarukhanyan G.E. The parametric method of the determination of electron energy on the data obtained by the method of a dosimetric wedge //Bulletin Kherson National Technical University. – 2012. – Vol. 2(45). – P.298-302.
5. Lazurik V.M., Tabata T., Lazurik V.T. A Database for Electron-Material Interactions // Radiation Physics and Chemistry. – 2001. – Vol.60. – P. 161-162.
6. Lazurik V.T., Lazurik V.M., Popov G., Zimek Z. Determination of electron beam parameters on radiation-technological facility for simulation of radiation processing // East Eur. J. Phys. 2014, v. 1(3), P. 74-78.
7. Lazurik V.T., Lazurik V.M., Popov G., Zimek Z. Two-parametric model of electron beam in computational dosimetry in radiation processing // Radiation Physics and Chemistry. 2016, v. 124, P. 230-234.
8. Lazurik V.T., Lazurik V.M., Popov G., Zimek Z. Method of Dosimetry Based on a Two-Parametric Model of Electrons Beam for Radiation Processing // Problems of Atomic Science and Technology. 2017, № 6, P. 137-141.
9. Bigolas J., Kulinski S., Maciszewski W., Pachan M., Plawski E, Zimek Z. Current approach to design of high-power electron accelerators to match actual requirements of radiation technology in Poland // Radiation Physics and Chemistry. – 2001. – Vol. 60. – P.161-162.
10. Lazurik V.T., Lazurik V.M., Popov G., Rogov Yu., Zimek Z. Information system and software for quali-ty control of radiation processing // Book. 232 p. IAEA: Collaborating Center for Radiation Processing and Industrial Dosimetry, Warsaw, Poland. 2011.

**ЛАЗУРИК Валентин Тимофеевич** – д. физ.-мат. н., профессор; декан факультета компьютерных наук, Харьковский национальный университет имени В.Н. Каразина, пл.Свободы 4, Харьков, Украина, 61022; e-mail: vtlazurik@karazin.ua; ORCID:0000-0002-8319-0764.

*Научные интересы: модели и методы компьютерного моделирования физических явлений.*

**ЛАЗУРИК Валентина Михайловна** – старший преподаватель кафедры искусственного интеллекта и программного обеспечения факультета компьютерных наук, Харьковский национальный университет имени В.Н. Каразина, пл. Свободы 4, Харьков, Украина, 61022; e-mail: lazurik@hotmail.com; ORCID: 0000-0002-3340-9780.

*Научные интересы: разработка компьютерных систем обработки научных данных.*

**ПОПОВ Геннадий Федорович** – к. ф.-м. н., доцент кафедры моделирования систем и технологий факультета компьютерных наук, Харьковский национальный университет имени В.Н. Каразина, пл. Свободы 4, Харьков, Украина, 61022; e-mail: popov_gen@yahoo.com; ORCID: 0000-0002-4794-8427.

*Научные интересы: моделирование и бенчмаркинг в радиационных технологиях.*

**САВАН Салах** – к. т. н., старший научный сотрудник факультета компьютерных наук, Харьковский национальный университет имени В.Н. Каразина, пл. Свободы 4, Харьков, Украина, 61022; e-mail: salah_sowan@yahoo.com; ORCID: 0000-0001-5404-6045.

*Научные интересы: компьютерное моделирование процессов в радиационных технологиях.*

**ЗИМЕК Збигнев** – профессор; заведующий центром радиационной химии и технологии Института ядерной химии и технологий, Польша, 03-195, Варшава, ул. Дородная 16, e-mail: Z.Zimek@ichtj.waw.pl; ORCID: 0000-0002-8653-5609..

*Научные интересы: разработка промышленных радиационно-технологических процессов.*

**УДК 612.039.517**

ЛЕЛЕКО Ю.Я., ГАНН В.В.

# РЕАКТОР НА СФЕРИЧЕСКОЙ СТОЯЧЕЙ ВОЛНЕ ЯДЕРНОГО ГОРЕНИЯ С ВНЕШНЕЙ ОТРИЦАТЕЛЬНОЙ ОБРАТНОЙ СВЯЗЬЮ ПО РЕАКТИВНОСТИ

## Введение

В работе развита нейтронная кинетика в несжимаемой нейтроно-размножающей среде с учетом выгорания. Рассмотрен сферически симметричный реактор, в котором среда двигается с ускорением по направлению к началу координат со скоростью $V(r)=V_R(R/r)^2$. Материал, попадающий в начало координат удаляется из реактора, а в периферийную область непрерывно поступает $^{238}U$ с той же скоростью. Органы управления реактора (поглощающие стержни), обеспечивающие его критичность и отрицательную обратную связь по реактивности – покоятся относительно корпуса реактора.

Активная зона реактора состоит из четырёх областей: внешняя инертная область состоит из $^{238}U$; промежуточная область, в которой происходит наработка топлива $^{239}Pu$ по схеме $^{238}U + n = {}^{239}U \rightarrow {}^{239}Np \rightarrow {}^{239}Pu$; внутренняя область ядерного горения, в которой происходит деление ядер $^{239}Pu$; центральная область, содержащая выгоревшее топливо. Показано, что в такой системе возникает сферическая стоячая волна ядерного горения, которая двигается радиально от центра, навстречу поступающему топливу. Волна горения состоит из двух областей: внешней, в которой происходит наработки и частичное выжигание топлива $^{239}Pu$, и внутренней $^{239}Pu$, в которой происходит деление ядер $^{239}Pu$ и их частичное воспроизводство из $^{238}U$. Получены распределения нейтронного потока, концентраций изотопов $^{238}U$, $^{239}Np$ и $^{239}Pu$, а также удельной мощности в стоячей волне. Исследована область существования сферических стоячих волн горения. Показано, что реактор на стоячей волне горения характеризуется всего двумя комбинациями ядерных сечений и одной универсальной функцией, определяющей границу устойчивости такой системы, причём область существования стоячих сферических волн горения оказывается шире чем область существования плоских одномерных бегущих волн горения в бесконечной среде. Построена диаграмма состояний такого реактора и определены границы его устойчивости.

Проведено компьютерное моделирование сферической волны ядерного горения, в режимах как бегущей волны в неподвижной среде, так и стоячей волны в движущейся среде. Компьютерная модель реактора, выполненная с использованием кода MCNPX, представляет собой шар, радиусом – 2 м, заполненный топливом на основе двуокиси урана. В режиме бегущей сферической волны ядерное горение начинается в центральной области активной зоны, содержащей обогащенный уран. Когда концентрация $^{239}Pu$ в $^{238}U$ становится достаточно высокой благодаря его наработке по схеме $^{238}U + n = {}^{239}U \rightarrow {}^{239}Np \rightarrow {}^{239}Pu$, тогда появляется сферическая волна горения, она отрывается от запальной области и продолжает радиальное движение к краям активной зоны в течение 150 лет. В наших модельных расчётах скорость волны горения составляла 0.5 см/год при мощности 240 МВт. Проведено компьютерное моделирование реактора в режиме стоячей сферической волне ядерного горения. Скорость движения среды и мощность реактора выбираются таким образом, что волна горения покоилась относительно корпуса реактора. Проведено сопоставление теоретических результатов с данными численного моделирования. Доказана возможность существования сферической стоячей волны ядерного горения в бридер-реакторе при подпитке обеднённым ураном.

## Теория сферической волны ядерного горения

Рассмотрим процесс ядерного горения в несжимаемой среде на основе урана, движущейся с постоянной скоростью V вдоль оси x неподвижной системы координат x, y, z, в которой профиль волны и органы управления реактора (поглощающие стержни) – покоятся.

Простейшее описание нейтронной кинетики, наработки и выгорания ядерного топлива в этом случае можно получить, перейдя в систему координат x', y', z', в которой топливо будет покоиться:

$$\frac{1}{\mathrm{v}}\frac{\partial \Phi}{\partial t} = \hat{D}\Phi + (\nu\Sigma_f - \Sigma_a)\Phi + S\,, \qquad (1)$$

$$\frac{\partial n_8}{\partial t} = -n_8\sigma_{a8}\Phi\,,\ \frac{\partial \tilde{n}_9}{\partial t} = \sigma_{89}\,n_8\Phi - \frac{\tilde{n}_9}{\tau_{89}}\,,\ \frac{\partial n_9}{\partial t} = \frac{\tilde{n}_9}{\tau_{89}} - \sigma_{a9}\,n_9\Phi\,,\ \frac{\partial n_c}{\partial t} = 2\sigma_{f9}\,n_9\Phi\,, \qquad (2)$$

К уравнениям (1)-(2) следует добавить граничные условия:

$$\Phi(\infty, t) = 0,\ \Phi(-\infty, t) = 0,\ n_9(\infty, t) = 0,\ \tilde{n}_9(\infty, t) = 0,\ n_c(\infty, t) = 0,\ n_8(\infty, t) = n_8(0)\,, \qquad (3)$$

Тогда система уравнений (1)-(3) примет вид:

$$\frac{1}{\mathrm{v}}\frac{\partial \Psi}{\partial t} + \frac{\vec{V}}{\mathrm{v}}\frac{\partial \Psi}{\partial \vec{r}} = \hat{D}\Psi + (\nu\Sigma_f - \Sigma_a)\Psi + S\,,\ \frac{\partial n_8}{\partial t} + \vec{V}\frac{\partial n_8}{\partial \vec{r}} = -n_8\sigma_a\Psi\,, \qquad (5)$$

$$\frac{\partial \tilde{n}_9}{\partial t} + \vec{V}\frac{\partial \tilde{n}_9}{\partial \vec{r}} = \sigma_{89}\,n_8\Psi - \tilde{n}_9/\tau_{89}\,,\ \frac{\partial n_9}{\partial t} + \vec{V}\frac{\partial n_9}{\partial \vec{r}} = \tilde{n}_9/\tau_{89} - \sigma_a\,n_9\Psi\,,\ \frac{\partial n_c}{\partial t} + \vec{V}\frac{\partial n_c}{\partial \vec{r}} = 2\sigma_f\,n_9\Psi\,, \qquad (6)$$

Далее получим уравнения в зависимостях от флюенса для сферической волны ядерного горения. Используя преобразование для уравнений (5) и (6): $\vec{r} = \vec{r}' + \vec{V}(r)t;\ \Phi(\vec{r}', t) = \Psi(\vec{r}, t)$, где $\vec{V}(\vec{r}) = -\vec{r}\,V_R R^2/r^3$ и введя вместо радиуса r новую переменную: $\varphi(r) = \sigma_a\int_r^\infty \frac{\Psi(r')}{V(r')}dr' = \frac{\sigma_a}{V_R R^2}\int_r^\infty \Psi(r')r'^2\,dr'$, пропорциональную флуенсу *F(x)*, получим:

$$\frac{D\sigma_a^2}{2V_R^2R^4}\frac{\partial}{\partial\varphi}\left(r^4(\varphi)\frac{\partial\Psi^2}{\partial\varphi}\right) + \frac{\nu}{k_{eff}}\Sigma_f - \Sigma_a - 0\,,\ \frac{dn_8}{d\varphi} = -n_8\,,\ \frac{dn_9}{d\varphi} = n_8\sigma_{89}/\sigma_a - n_9\,, \qquad (7)$$

$$\frac{dn_c}{d\varphi} = 2n_9\sigma_f/\sigma_a\,,\ P = 4\pi V_R Q R^2/\sigma_a\int\Sigma_f d\varphi\,,\ \Sigma_f = \sigma_f\,n_9\,,\ \Sigma_a = \sigma_a(n_8 + n_9) + \sigma_c n_c\,, \qquad (8)$$

Интегрируя непосредственно уравнения (7) и (8) получаем уравнение для плотности нейтронного потока:

$$n_8(\varphi) = n_0 e^{-\varphi}\,,\ n_9(\varphi) = \sigma_{89}/\sigma_a n_0\varphi e^{-\varphi}\,,\ n_c(\varphi) = \sigma_f\sigma_{89}/\sigma_a^2 n_0[1-(1+\varphi)e^{-\varphi}]\,, \qquad (9)$$

$$\frac{D\sigma_a}{2n_0V_R^2}\frac{d}{d\varphi}\left(\frac{r^4(\varphi)}{R^4}\frac{d\Psi^2}{d\varphi}\right) = 2\beta + (1-2\beta)e^{-\varphi} + \left(\frac{\sigma_{89}}{\sigma_a} - 2\beta\right)\varphi e^{-\varphi} - \frac{\sigma_f\sigma_{89}}{\sigma_a^2}\nu(1-\rho)\varphi e^{-\varphi}\,, \qquad (10)$$

где $\beta = \dfrac{\sigma_c\sigma_f\sigma_{89}}{\sigma_a^3}$

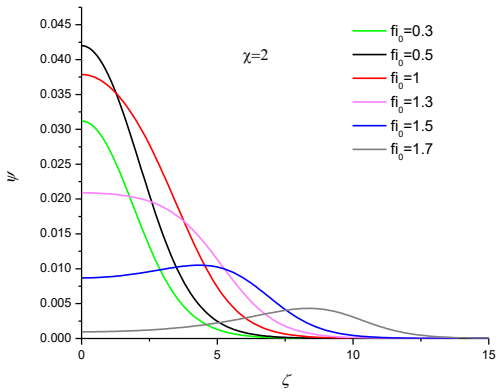



*Рис. 1 Радиальные зависимости нейтронного потока ψ(ζ) в стоячих волнах, χ = 2*

На Рис. 1 приведены радиальные профили нейтронного потока в стоячих волнах горения с различными значениями параметра $\varphi_0$ для зоны с $\chi = 2$. Из Рис. 1 видно, что с увеличением значения $\varphi_0$ максимум волны горения удаляется от центра зоны и уходит на бесконечность при $\varphi_0 = \chi$. При этом, минимальные размеры зоны: $r_{\min} \sim 1.5\sqrt[4]{\frac{D}{\sigma_a n_0}}$ достигаются уже при $\varphi_0 \sim 0.4$.

**Компьютерное моделирование сферической волны горения.**

Компьютерная модель TWR представляет собой шар радиусом – 2 м, заполненного топливом на основе двуокиси урана. В режиме бегущей волны ядерное горение начинается в центральной области активной зоны, содержащей обогащенный уран. Когда концентрация $^{239}$Pu в $^{238}$U становится достаточно высокой благодаря его наработке по схеме $^{239}$U + n = $^{239}$U → $^{239}$Np → $^{239}$Pu, тогда возникает сферическая волна горения, она отрывается от запальной области и продолжает движение к краям активной зоны. В наших модельных расчётах скорость перемещения максимума волны горения составляла 0.5 см/год при мощности 240 МВт. На Рис. 2 приведены графики, иллюстрирующие движение бегущей волны в течение 30 лет.

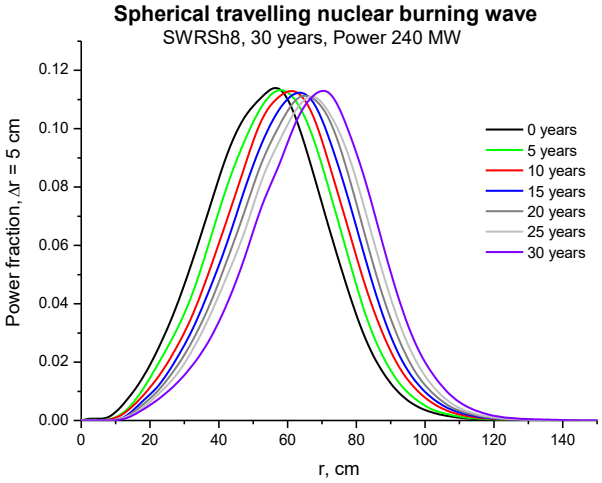



*Рис. 2 Распределение мощности по слоям бегущей сферической волны горения на протяжении 30 лет при полной мощности 240 МВт.*

Математическая модель SWR включала перемещение топлива навстречу волне горения со скоростью, обеспечивающей стационарность процесса горения в реакторе. На Рис. 3 показан результат моделирования стоячей волны ядерного горения на протяжении 20 лет. Из этого рисунка следует, что выбранные параметры системы обеспечивают стационарность сферической волны ядерного горения при подпитке реактора обеднённым урановым топливом. Математическое моделирование сферического реактора на стоячей и бегущей волне горения дало хорошее согласие аналитических результатов с данными численного моделирования.

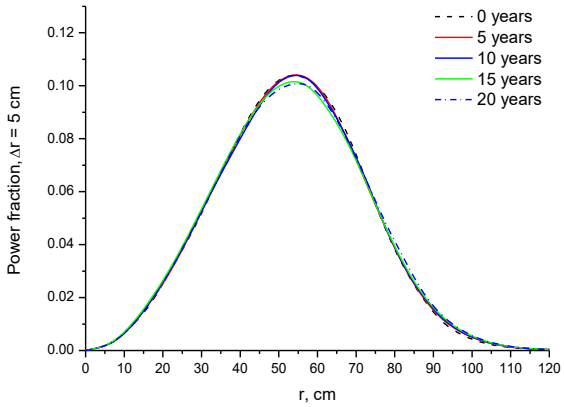



*Рис. 3 Радиальное распределение мощности по слоям стоячей сферической волны горения на протяжении 20 лет.*

ЛИТЕРАТУРА

1. Феоктистов Л. П. Нейтронно-делительная волна // Докл. Акад. Наук СССР. 1989, т. 309, с. 864-867.
2. Ellis T., Petroski R. Traveling-Wave Reactors: A Truly Sustainable and Full-Scale Resource for Global Energy Needs // Proceedings of ICAPP ‘10 San Diego, CA, USA, June 13-17, 2010 Paper 10189.
3. Gann V. V., Gann A. V. BENCHMARK on traveling wave fast reactor with negative reactivity feedback obtained with MCNPX code // 4 International Conference “Current Problems in Nuclear Physics and Atomic Energy” (NPAE-Kyiv2012) September 3 - 7, 2012, Kyiv, Ukraine. Proceedings Part II. P. 421- 425.
4. Лелеко Ю.Я., Ганн В.В., Ганн А.В. Моделирование реактора на стоячей волне ядерного горения // Труды Международной Научно-Технической Конференции «Компьютерное моделирование в наукоёмких технологиях». Харьков, 26-31мая 2016, Харьков 2016, стр.206-209.
5. TERRAPOWER, LLC Traveling Wave Reactor Develop Program Overview // http://dx.doi.org/10.5516/NET.02.2013.520.
6. Yu.Y. Leleko, V.V. Gann, A.V. Gann. NUCLEAR REACTOR ON CYLINDRICAL STANDING BURNING WAVE WITH AN EXTERNAL NEGATIVE REACTIVITY FEEDBACK // Problems of Atomic Science and Technology. 2017, № 2 (108), pp. 138-143.
7. V.V.Gann, Yu.Y.Leleko, A.V.Gann COMPUTER SIMULATION OF NUCLEAR REACTOR ON CYLINDRICAL STANDING BURNING WAVE // Proceedings of NUCLEAR 2017 The 10th International Conference on Sustainable Development through Nuclear Research and Education, Pitesti, 2017, May 24-26, pp. 161-168.
8. V.V. Gann, Yu.Y. Leleko, A.V. Gann. COMPUTER SIMULATION OF NUCLEAR REACTOR ON SPHERICAL STANDING BURNING WAVE // Proceedings of the International Conference Nuclear for New Europe, Portorož, Slovenia, September 9 -12, 2019, pp. 1007.1-1007.8.
9. Yu.Y. Leleko, V.V. Gann, A.V. Gann. SPHERICAL STANDING BURNING WAVE WITH EXTERNAL AUTOMATIC REACTIVITY CONTROL // Problems of Atomic Science and Technology. 2019, № 5 (123), pp. 18-24.
10. Лелеко Ю.Я., Ганн В.В Критические размеры стоячих волн ядерного горения // Тезисы докладов XV Международной Научно-Технической Конференции молодых учёный и специалистов «Проблемы современной ядерной энергетики». Харьков, 13-15 ноября 2019, Харьков 2019, стр.62-63.

**ГАНН Владимир Васильевич** – д. ф.-м. н., старший научный сотрудник ННЦ ХФТИ, улица Академическая, 1, Харьков-108, Украина, 61108; e-mail:gann@kipt.kharkov.ua; ORCID: 0000-0002-3451-1840.

Научные интересы:

– *математическое моделирование радиационных процессов.*

**ЛЕЛЕКО Юрий Яковлевич** – младший научный сотрудник ННЦ ХФТИ, улица Академическая, 1, Харьков-108, Украина, 61108; e-mail: makswell.com@gmail.com; ORCID: 0000-0001-9946-5569.

Научные интересы:

– *математическое моделирование ядерных установок.*

**УДК 004.452**

ЛИТВИНОВ Н.А. ЛАЗУРИК В.М.

# ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ ПАНЕЛИ С ИСПОЛЬЗОВАНИЕ RDF ХРАНИЛИЩА

## Введение

Конец двадцатого века стал началом новой цифровой эпохи. Глобальные изменения коснулись сферы информационных технологий, поэтому двадцать первый век характеризуется массовой общественной компьютеризацией и информатизацией. В связи с этим лавинообразно растет разработка программного обеспечения, возникают и реализуются новые идеи, отмирают старые. Судьба новых технологий после проверки временем складывается по-разному. Некоторые приживаются, со временем совершенствуются и расширяются, другие отвергаются. Но есть и третья категория разработок, которые вроде бы и хороши, но развитие их либо происходит очень медленно, либо вообще откладывается до лучших времен.

В 2001 году Тим Бернерс-Ли [1,2] вместе с двумя другими исследователями представили новую концепцию хранения и распространения информации в интернете. Они посмотрели на Всемирную паутину как на единую децентрализованную систему. Было введено название «Семантическая паутина» или «Гигантский глобальный граф» как следующий шаг в развитии Всемирной паутины. Концепция семантической паутины была принята и продвигается консорциумом Всемирной паутины (W3C−World Wide Web Consortium). Были разработаны стандарты:

- Resource Description Framework (RDF) – модель для представления данных, а в особенности, метаданных;
- SPARQL – протокол и язык запросов для данных RDF;
- Uniform Resource Identifier (URI) − унифицированный идентификатор ресурса.

Стоило ожидать, что такая интересная и революционная идея будет иметь реальное воплощение, стремительное и быстрое развитие. Но, не сложилось, в силу разных причин.

В работе рассмотрена RDF модель данных, инструменты и программные средства для работы с RDF хранилищами. На примере разработки информационной панели создано RDF хранилище данных о детских дошкольных учреждениях и сформированы запросы с использованием языка SPARQL. Обсуждаются трудности, возникшие на пути использования RDF хранилищ.

## RDF модель

Модель RDF представляет собой утверждения о ресурсах в виде, пригодном для машинной обработки, и является частью концепции семантической паутины [3]. В основе описания ресурса в RDF лежит понятие триплета, который представляет собой граф из двух вершин, соединенных ребром (рис.1). Утверждение, высказываемое о ресурсе, имеет вид «субъект — предикат — объект».

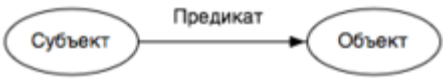



*Рис. 1- RDF триплет*

Субъектом в RDF тройке является ресурс или узел в графе, объект − другой узел или литеральное значение, предикат (ребро) – отношение между ними. Ресурсы определяются с помощью URI, представляющего собой символьную строку. URI является либо URL, либо URN, либо одновременно обоими. При этом URL помимо идентификации ресурса, предоставляет ещё и информацию о его местонахождении, а URN только идентифицирует ресурс в определённом пространстве имён, но не указывает его местонахождение [4]. Множество RDF утверждений образуют ориентированный граф, в котором вершинами являются субъекты и объекты, а рёбра отображают отношения. На рис.2 представлен пример

графа с набором триплетов [5], определяющим человека по имени Петр Иванов как автора некоторой статьи в журнале «Вопросы биологии».

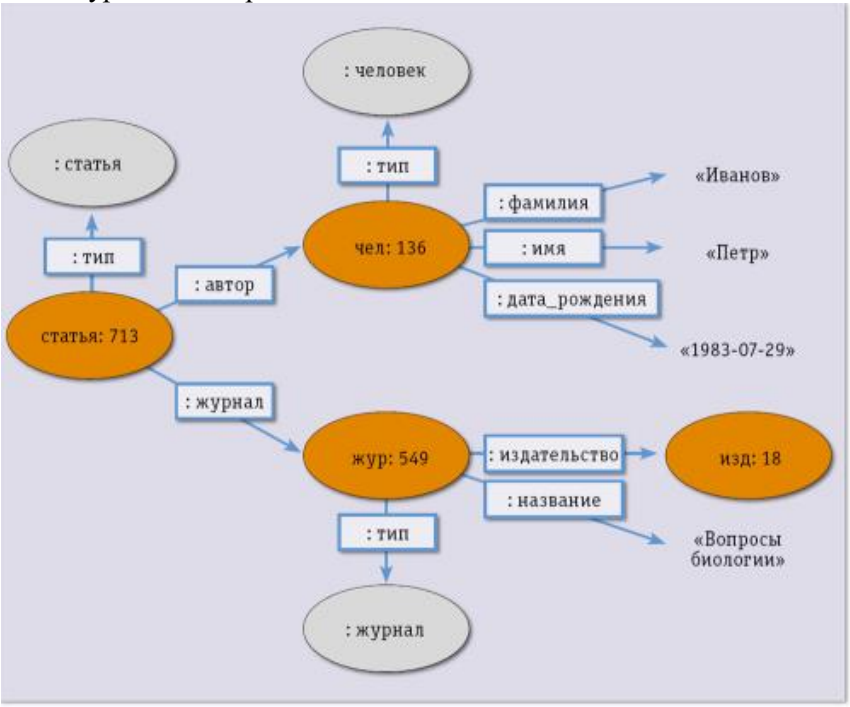



*Рис. 2 – Ориентированный граф модели RDF*

Для представления RDF в текстовом виде используются такие форматы: RDF/XML − стандартный на базе XML; N-Triples − простой; Turtle, N3 – компактные, удобные; JSON-LD − на базе JSON; RDFa и Microdata − формат RDF-разметки HTML-страниц [3].

RDF – это информационная модель, она не определяет семантику описываемого. Чтобы отразить семантику конкретного хранилища, необходимы словари, в которых представлены термины, имеющие одинаковый смысл, таксономия как иерархия терминов, и онтология для определения концепций и отношений между ними.

Если первоначально модель RDF была частью стека семантической паутины, то в настоящее время она используется, когда существует необходимость в представлении высококачественно связанных данных, позволяя при этом устранить неоднозначность и точно идентифицировать информацию.

### Инструментальные средства для разработки RDF баз данных

Для обработки RDF данных существуют языки запросов: SPARQL (стандарт W3C [6]), RQL, RDQL. Если перечислить некоторые инструментальные средства, существующие в мире, начиная с 2012 года и по сей день, список получится примерно таким:

- Apache Jena − Java API для разработки приложений Semantic Web. Продукт включает в себя несколько хранилищ, собственное хранилище троек (Jena TDB), интерфейс к реляционному хранилищу (Jena SDB), хранилище в памяти (In-Memory), а также средства для поддержки собственных хранилищ.
- GraphDB – RDF графовая база данных или триплетное хранилище, создано Ontotext. Может выполнять семантический вывод в масштабе, позволяя пользователям создавать новые семантические факты из существующих фактов, имеется способность визуализировать тройки.
- OpenLink Software Virtuoso – имеет собственное RDF хранилище, полную реализацию SPARQL, возможность чтения данных RDF из файлов формата XML и Turtle, обеспечивает хорошую производительность.
- Eclipse RDF4J – модульная среда Java с открытым исходным кодом для работы с данными RDF. Обеспечивает анализ, хранение, запросы к данным, предлагает простой API и две

RDF базы данных (хранилище в памяти и собственное хранилище), поддерживает SPARQL.

- Мультимодельные системы управления базами данных (СУБД), Oracle поддерживает реляционную, графовую модели данных, обеспечивает хранение документов и RDF данных. IBM DB2 тоже поддерживает RDF.

**Использование RDF хранилища для поиска детского учреждения**

Для реализации задачи поиска детского учреждения в качестве главного источника идентификации сущностей был выбран Dbpedia. DBpedia − проект, направленный на извлечение структурированной информации из данных, созданных в рамках проекта Википедии и публикации её в виде доступных под свободной лицензией наборов данных [7]. Базы данных DBpedia описывают более чем 6 млн понятий. Второстепенные источники не существуют в действительности. RDF допускает возможность использования в триплете ресурса из любого источника, как того, что существует в сети, так и несуществующего. Пусть ресурс www.kindergarten.com содержит информацию, необходимую для оценки надёжности детских учреждений, а www.newyorknews.com − для оценки районов, в которых они находятся.

Задание для реализации выглядело как поиск детского сада для малыша на Украине. Но при проверке доступной информации возникли существенные трудности. Информация, на основании которой формируется хранилище, прежде всего, должна быть полной, доступной и проверяемой. Данные RDF по Украине, которые содержатся в DBpedia не полные, детальная информация заканчивается на регионах и городах, улицы представлены в виде строки, а не ресурса. В то время как полная информация о детском садике должна быть доступна по цепочке: город → районы → детские учреждения района → описание конкретного детского сада. Поэтому, для реализации работающего примера, данные которого могут быть проверены, были взяты детские сады США, город New York, район Brooklyn.

Для формирования RDF хранилища был использован редактор исходного кода Visual Studio Code с установленным расширением Language Support for RDF related language syntax для подсветки синтаксиса. Созданный файл хранилища code.n3 содержит триплеты, представляющие собой текстовые строки в формате N-Triples. В листинге 1 содержится фрагмент кода, который определяет тип сущности New York City как City и декларирует район Brooklyn как его часть. Префикс − фактически ссылка на источник либо на путь к перечню ресурсов источника.

Листинг 1.

```
@prefix dbo: <http://dbpedia.org/ontology>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

 dbr:New_York_City rdf:type dbo:City
 dbr:Brooklyn dbo:isPartOf dbr:New_York_City..
 dbr:Brooklyn rdfs:label "Brooklyn".
```

При формировании данных о детских учреждениях достаточной информацией является совокупность следующих полей: название, адрес, дата основания, количество детей, описание учреждения. На листинге 2 приведены сформированные в виде триплетов данные о Brooklyn Free School.

Листинг 2.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix dbo: <http://dbpedia.org/ontology/>.

dbr:Brooklyn_Free_School rdfs:label "Brooklyn Free School"@en.
dbr:Brooklyn_Free_School dbo:address "372 Clinton Avenue"@en.
dbr:Brooklyn_Free_School dbo:foundingYear "2004-01-01"^^xsd:date.
dbr:Brooklyn_Free_School dbo:numberOfStudents "80"^^xsd:integer.
dbr:Brooklyn_Free_School dbo:abstract
    '''The Brooklyn Free School is a private, ungraded,democratic free
    school in Fort Greene, Brooklyn, founded in 2004. Students range
```

```
    in age from 4 to 18 years old. The school follows the noncoercive
    philosophy of the 1960s/70s free school movement schools, which
    encourages self-directed learning and protects child freedom of
    activity. . . . The school is funded through sliding-scale
    tuition, grants, and donations.''' @en.
```

В приведенном фрагменте кода присутствуют предикаты `rdf:type` и `dbo:isPartOf`. Описание `rdf:type` можно найти, перейдя по URI префикса. Для формирования URI с использованием `dbo:isPartOf`, необходимо сложить URI префикса с именем сущности: `http://dbpedia.org/ontology + isPartOf -> http://dbpedia.org/ontology/isPartOf`.

Аналогично можно сформировать информацию о рейтинге детских учреждений из источника www.kindergarten.com, обозначив его URI как префикс kdr: `@prefix kdr: <http://kindergarten.com/rating/>`.Среди всей доступной информации самой важной можно считать оценку персонала − staffAssessment, оценку питания – nutritionAssessment, досуга − leisureAssessment, общую оценку − overallRating и впечатление от детского учреждения – review (листинг 3).

Листинг 3.

```
dbr:Brooklyn_Free_School kdr:overallRating "7"^^xsd:integer.
dbr:Brooklyn_Free_School kdr:staffAssessment "6"^^xsd:integer.
dbr:Brooklyn_Free_School kdr:equipmentAssessment "7"^^xsd:integer.
dbr:Brooklyn_Free_School kdr:nutritionAssessment "8"^^xsd:integer.
dbr:Brooklyn_Free_School kdr:leisureAssessment "7"^^xsd:integer.
dbr:Brooklyn_Free_School kdr:review
    '''BFS is a wonderful, creative environment for a child, and even
    though there is no testing or formal assessments. . . '''@en.
```

Используя источник www.newyorknews.com, можно подготовить информацию о новостях. Здесь важными данными являются: само название новости (dc:title), его описание (dc:desctiption), дата события (dc:date) и хэштеги (dbr:Hashtag), описывающие суть событий. Благодаря хэштегам появляется возможность в дальнейшем посмотреть, какие благоприятные или неблагоприятные события происходили в определённых районах, и показать детские учреждения, находящиеся неподалёку (листинг 4).

Для того чтобы разместить хранилище и работать с ним используем Apache jena fuseki сервер. Скачать его можно с официального сайта, для работы сервер использует порт 3030. Чтобы войти на главную страницу графического интерфейса, следует перейти по ссылке *http://localhost:3030/* . Вид интерфейса представлен на рис.3. На главной странице присутствует информация о работоспособности сервера и существующих datasets. Dataset содержит все загруженные из RDF хранилища триплеты в пригодном для использования виде. Чтобы создать новый dataset, нужно перейти на вкладку *manage datasets*, нажать на кнопку *add one*, ввести желаемое название и завершить создание кнопкой *create dataset*.

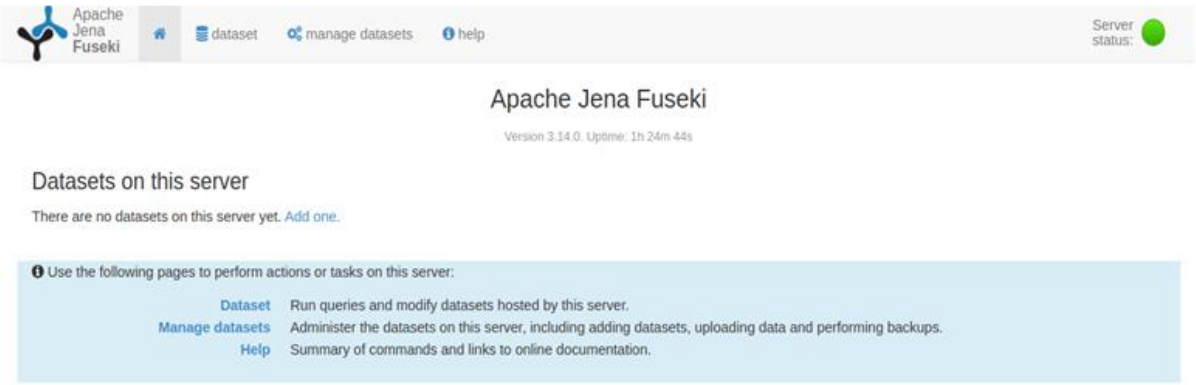



*Рис. 3 – Главная страница Apache jena fuseki*

Листинг 4.

```
@prefix nyn: <http://newyorknews.com/>.
```

```
nyn:News_c1edf922-9f0a-4cda rdf:type nyn:News.
nyn:News_c1edf922-9f0a-4cda nyn:News_UUID
    "c1edf922-9f0a-4cda".
nyn:News_c1edf922-9f0a-4cda dc:title
    "The roof fell off at a local school"@en.
nyn:News_c1edf922-9f0a-4cda dc:desctiption
    '''As a result of a severe thunderstorm . . . '''@en.
nyn:News_c1edf922-9f0a-4cda dbo:location dbr:Clinton_Avenue.
nyn:News_c1edf922-9f0a-4cda dc:date
    "2004-04-12T13:20:00Z"^^xsd:dateTimeStamp.
nyn:News_c1edf922-9f0a-4cda dbr:Hashtag "accident".
nyn:News_c1edf922-9f0a-4cda dbr:Hashtag "naturalPhenomenon".
```

Для добавления триплетов в новый dataset необходимо нажать на кнопку *upload data* (рис.4).

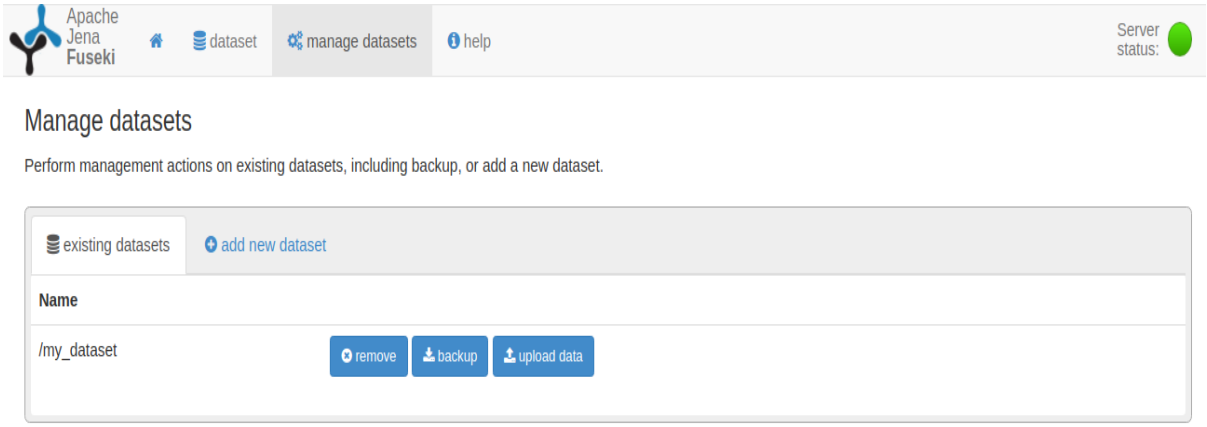



*Рис. 4 –Создание нового dataset*

Далее выбираем файлы с триплетами кнопкой *select files…* и подтверждаем добавление кнопкой *upload all*. В случае если какой-либо файл содержит ошибки, полоска прогресса будет выделена оранжевым цветом и выше написана причина возникновения ошибки. На этом этапе выявляются синтаксические ошибки. В случае отсутствия ошибок, все триплеты будут помещены в dataset, а полоска выделится зелёным. Вкладка *query* – работа с запросами (рис.5).

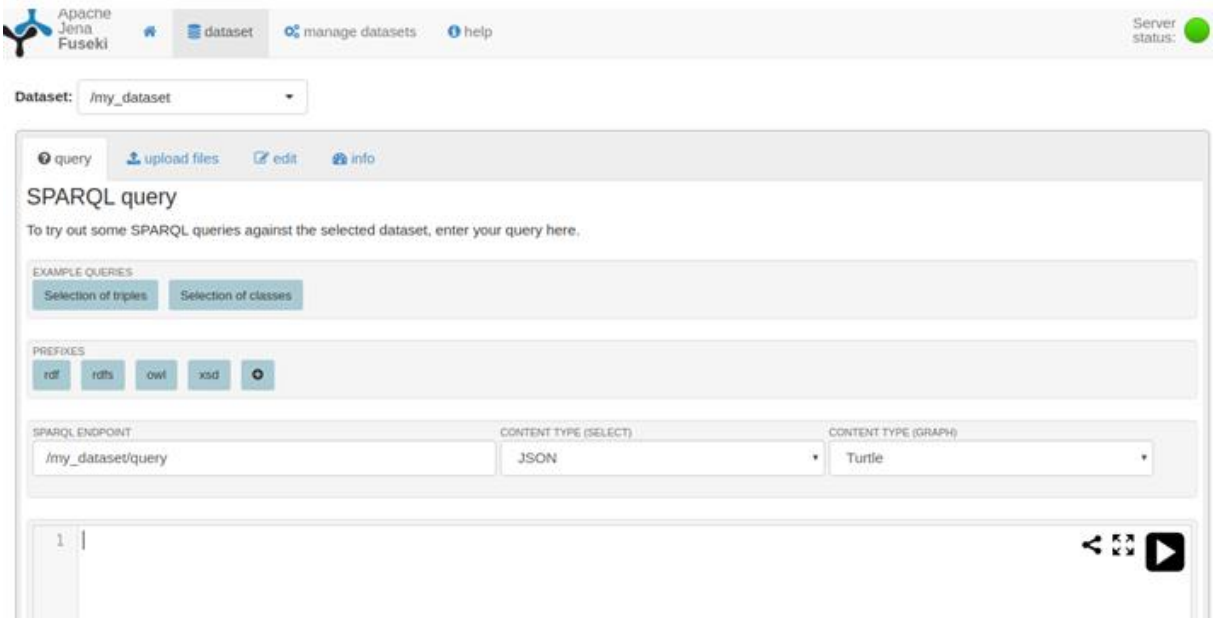



*Рис. 5 –Страница для формирования запроса*

Для создания запросов используется язык запросов SPARQL. Для того, чтобы получить информацию о детском учреждении Brooklyn Free School, о его рейтинге и всех новостях, которые имели место в районе его расположения, составим запрос, в котором объединим все три источника информации. Текст запроса приведен в листинге 5.

Листинг 5.

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT *
WHERE  {   {
SELECT ?school ?school_predicate ?school_object
WHERE { ?school ?school_predicate ?school_object.
FILTER (?school = dbr:Brooklyn_Free_School) }  }
UNION  {
SELECT ?news ?news_predicate ?news_data
WHERE { dbr:Brooklyn_Free_School dbo:city ?city.
?streets dbo:location ?city.
?news dbo:location ?streets.
?news ?news_predicate ?news_data } } }
```

В первых трех строках запроса определяются префиксы для формирования ресурсов в SPARQL. В операторе `SELECT` использована конструкция `UNION`, которая позволяет объединить запрос на получение информации о рейтинговых показателях в рассматриваемом детском учреждении и сведений о событиях, происходящих вблизи его. Запрос отображает главную особенность RDF хранилищ и языка SPARQL, а именно, получение информации из разных источников и выполнение запроса на основании логических умозаключений, учитывающих отношения между сущностями. Фрагмент результата представлен на рис.6.

| | | | |
|---|---|---|---|
| 18 | <http://dbpedia.org/resource/Brooklyn_Free_School> | <http://dbpedia.org/ontology/abstract> | graduates. The school was the first free school in New York City since 1975. It started in a rented portion of a Park Slope Methodist church, but moved to a brownstone in Fort Greene. Students participate in the design of classes and in the school's governance, which is done at a weekly Democratic Meeting. Staff and students all have equal votes. The school is funded through sliding-scale tuition, grants, and donations."@en |
| 19 | <http://dbpedia.org/resource/Brooklyn_Free_School> | <http://kindergarten.com/rating/overall_rating> | "7"^^xsd:integer |
| 20 | <http://dbpedia.org/resource/Brooklyn_Free_School> | <http://dbpedia.org/ontology/address> | "372 Clinton Avenue"@en |
| 21 | <http://dbpedia.org/resource/Brooklyn_Free_School> | <http://kindergarten.com/rating/leisure_assessment> | "7"^^xsd:integer |
| 22 | <http://dbpedia.org/resource/Brooklyn_Free_School> | <http://kindergarten.com/rating/equipment_assessment> | "7"^^xsd:integer |
| 23 | <http://dbpedia.org/resource/Brooklyn_Free_School> | <http://kindergarten.com/rating/review> | "BFS is a wonderful, creative environment for a child, and even though there is no testing or formal assessments, it is very clear how much the children are learning--not just facts and figures, but critical thinking skills, and creative and holistic ways of approaching a variety of topics and subjects."@en |
| 24 | <http://dbpedia.org/resource/Brooklyn_Free_School> | <http://dbpedia.org/ontology/city> | <http://dbpedia.org/resource/Brooklyn> |
| 25 | <http://dbpedia.org/resource/Brooklyn_Free_School> | <http://dbpedia.org/ontology/numberOfStudents> | "80"^^xsd:integer |
| 26 | <http://dbpedia.org/resource/Brooklyn_Free_School> | <http://www.w3.org/2000/01/rdf-schema#label> | "Brooklyn Free School"@en |

*Рис. 6 – Фрагмент результата выполнения SPARQL запроса*

**Заключение**

В работе рассмотрены особенности использования RDF хранилищ данных. Уделено внимание RDF модели данных, проведен поиск программных средств для работы. Проанализированы форматы представления RDF данных. Реализована задача поиска детского учреждения, в которую вошла разработка RDF хранилища и размещения его на сервере Apache jena fuseki. С использованием SPARQL сформированы запросы на получение информации о детских дошкольных учреждениях. Получены результаты выполнения запроса в среде Apache jena fuseki. Проанализированы трудности, возникшие на пути реализации задачи.

ЛИТЕРАТУРА

1. Tim Berners-Lee. Semantic Web Road map (09.1998). [Электронный ресурс] Режим доступа: https://www.w3.org/DesignIssues/Semantic.html
2. Тим Бёрнерс-Ли. Гигантский Глобальный Граф. [Электронный ресурс] Режим доступа: http://goodarticles.narod.ru/ggg.html
3. Resource Description Framework. [Электронный ресурс] Режим доступа: https://ru.wikipedia.org/wiki/Resource_Description_Framework.
4. URI. [Электронный ресурс] Режим доступа: https://ru.wikipedia.org/wiki/URI.
5. В. Головков, А. Портнов, В. Чернов. RDF — инструмент для неструктурированных данных. [Электронный ресурс] Режим доступа: https://www.osp.ru/os/2012/09/13032513/
6. Сайт Консорциума Всемирной паутины W3C. [Электронный ресурс] Режим доступа: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/
7. DBpedia. Материал из Википедии — свободной энциклопедии. [Электронный ресурс] Режим доступа: https://ru.wikipedia.org/wiki/DBpedia

**ЛИТВИНОВ Никита Андреевич** – студент факультета компьютерных наук Харьковского национального университета имени В. Н. Каразина.

Научные интересы:

*– разработка программного обеспечения.*

**ЛАЗУРИК Валентина Михайловна** – старший преподаватель кафедры искусственного интеллекта и программного обеспечения факультета компьютерных наук Харьковского национального университета имени В. Н. Каразина.

Научные интересы:

*– разработка компьютерных систем для моделирования процессов в радиационных технологиях;*

*– организация баз данных.*

**УДК 519.223**

МАКСИМУК А.Р, БАКУМЕНКО Н.С.

# КОМП'ЮТЕРНА МОДЕЛЬ КЛАСИФІКАЦІЇ СТАНІВ МЕДИКО-БІОЛОГІЧНОЇ СИСТЕМИ ЗА ДОПОМОГОЮ МЕТОДУ ЛОГІСТИЧНОЇ РЕГРЕСІЇ

У сучасному світі, під час технічного процесу, дуже багато завдань в медичних і біологічних системах є невирішеними. Однією з важливих задач, що зустрічаються в медичній практиці, є надійна діагностика захворювання. Деякі захворювання досить важко діагностуються. Тому застосування сучасних інтелектуальних методів аналізу даних для підтримки прийняття рішень в медицині є актуальним [1].

Метою даної роботи є розробка моделі класифікації станів медико-біологічної системи, за допомогою якої можна буде визначити до якого класу (в нашому випадку - «Пацієнт вижив» і «Пацієнт помер»), відноситься хворий, виходячи із зібраних даних лабораторних досліджень.

Для досягнення поставленої мети необхідно реалізувати такі завдання: здійснити порівняльний аналіз методів класифікації; розробити математичну модель для рішення задач класифікації за допомогою методу логістичної регресії; розробити програмно-алгоритмічну модель класифікації системи; провести тестування і аналіз отриманих результатів.

Класифікація - один з розділів машинного навчання, присвячений вирішення наступного завдання: є безліч об'єктів (ситуацій), розділених деяким чином на класи [2]. Основне завдання класифікації полягає в розбитті безлічі елементів даних на категорії або класи так, щоб всі елементи всередині кожного класу мали достатню кількість загальних ознак, що дозволяє знехтувати їх індивідуальними відмінностями [3].

Дана задача може бути вирішена різними методами. До таких методів належать [4]: логістична регресія, метод Баєса, дерева рішень, метод опорних векторів і т.д.

Для реалізації заданої цілі було обрано метод логістичної регресії, завдяки деяким перевагам: даний метод демонструє короткий час навчання, є досить потужним для бінарної класифікації, використовує замість прямої лінії криву, що спрощує розподіл даних на групи.

Логістична регресія – це статистична модель, що використовується для передбачення ймовірності виникнення деякої події шляхом підгонки даних до логістичної кривої. Логістична регресія застосовується для передбачення ймовірності виникнення деякої події за значеннями безлічі ознак. Для цього вводиться так залежна змінна у, приймаюча лише одне з двох значень. Як правило, це числа 0 (подія не відбулося) і 1 (подія відбулася), і безліч незалежних змінних (також званих ознаками, предикторами або регресорами) - речових X1, X2, ..., Xn, на основі значень яких потрібно обчислити ймовірність прийняття того чи іншого значення залежної змінної [5].

Іншими словами, основною ідеєю логістичної регресії є те, що простір вихідних значень може бути розділений лінійною межею на дві області, що відповідають різним класам. У разі двох вимірів лінійною межею виступає пряма лінія без вигинів. У разі трьох – площина, і так далі. Межа задається в залежності від наявних вихідних даних і навчального алгоритму. Щоб все працювало, точки вихідних даних повинні розділятися лінійною межею на дві вищезазначених області. Якщо точки вихідних даних задовольняють цій вимозі, то їх можна назвати точками, що лінійно розділяються [6].

Математичне рівняння, що оцінює лінію простої лінійної регресії:

$$Y = a + bx, \quad (1)$$

де x - незалежна змінна або предиктор;

Y – залежна змінна або змінна відгуку;

a – вільний член лінії оцінки;

b – кутовий коефіцієнт або градієнт оціненої лінії.

Коефіцієнти a і b називають коефіцієнтами регресії оціненої лінії [7].

Для виконання поставленого завдання будемо працювати з вибіркою даних на основі лабораторних досліджень стану пацієнтів. Залежна змінна, в нашому випадку – змінна, що

показує стан пацієнта, а саме – «Пацієнт вижив», «Пацієнт помер» (0 чи 1). До незалежних речових змінних відносяться: вік, стать, чи приймав пацієнт стероїди, чи приймав пацієнт антивірусні засоби, чи є ознаки нездужання, втоми, анорексія, збільшення та ущільнення печінки, чи прощупується селезінка, чи є в пацієнта асцит, опис показників крові (білірубін, альбумін, фермент АСТ, протромбін, фосфатаза), чи робилась гістологія пацієнту.

**Висновки**

На даний момент, метод логістичної регресії активно вводиться в світову медицину. Це пояснюється наявністю таких основних можливостей даного методу:

1. Визначення для конкретної групуючої ознаки Y, набору ознак-предикторів $X_i$, що пояснюють наборами своїх значень ймовірності віднесення певного спостереження до групи порівняння.
2. Впорядкування відібраних ознак-предикторів $X_i$ за рівнем впливу на залежну ознаку Y.
3. Оцінка надійності пояснення приналежності спостережень (пацієнтів) до конкретного класу залежної ознаки Y, за допомогою певної комбінації відібраних ознак-предикторів $X_i$.
4. Можливість оцінки не одного, а багатьох рівнянь логістичної регресії, шляхом використання різних алгоритмів оцінки цих рівнянь.
5. Вибір різних наборів потенційних предикторів, виходячи з яких алгоритми, що використовуються оцінюють різні рівняння логістичної регресії [8].

ЛІТЕРАТУРА

1. Артамонова Н. Сучасні тенденції розвитку наукової медичної інформації. Вісник Книжкової палати. 2009. Вип. 8. С. 27-30. URL: http://nbuv.gov.ua/UJRN/vkp_2009_8_12
2. statistica.ru - Основы линейной регрессии [Електронний ресурс] Режим доступа: http://statistica.ru/theory/osnovy-lineynoy-regressii/
3. О.І. Черняк, П.В. Захарченко Інтелектуальний аналіз даних: підручник. К. : Знання, 2014. 599 с.
4. Ким О. Дж., Мьюллер Ч.У., Клекка У.Р., и др. Факторный, дискриминантный и кластерный анализ / Под ред. И. С. Енюкова. М.: Финансы и статистика, 1989. 215 с.
5. biometrica.tomsk.ru - Логистическая регрессия в медицине и биологии [Електронний ресурс] Режим доступа: http://www.biometrica.tomsk.ru/logit_0.htm
6. Witten, Ian H. Data Mining: Practical Machine Learning Tools and Techniques. Elsevier Inc., 2011. 664 p.
7. machinelearning.ru – Классификация [Електронний ресурс] Режим доступа: http://www.machinelearning.ru/wiki/index.php?title=%D0%9A%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F
8. biometrica.tomsk.ru - Логистическая регрессия в медицине и биологии [Електронний ресурс] Режим доступа: http://www.biometrica.tomsk.ru/logit_3.htm

**МАКСИМУК Анастасія Родіонівна** – студентка 4 курсу факультету комп'ютерних наук спеціальності «Автоматизація та комп'ютерно-інтегровані технології»; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: nastia20maksimuk@gmail.com; ORCID: 0000-0001-8185-6297.

Наукові інтереси:

*– впровадження методів машинного навчання в системну інженерію.*

**БАКУМЕНКО Ніна Станіславівна** – к. т. н., доцент; доцент кафедри теоретичної та прикладної системотехніки; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: n.bakumenko@karazin.ua; ORCID: 0000-0003-3496-7167.

Наукові інтереси:

*– інтелектуальний аналіз даних.*

**УДК 004.896**

МАЛАХОВА М.О., СЕРДЮК С.А.

# РОЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМЛЕКСА ДЛЯ УПРАВЛЕНИЯ РОБОТОМ С ЭЛЕМЕНТАМИ МАШИННОГО ОБУЧЕНИЯ

**Вступление**

В современном мире все больше и больше возрастает потребность в использовании роботов в различных сферах жизни. Использование машинного обучения в робототехнике значительно расширяет их использование, облегчая человеческий труд и имеет большую эффективность из-за возможности оперирования большими количествами данных, что является сложным, а иногда и невозможным для человека. Все это способствует перспективному развитию этого направления в дальнейшем.

В данной работе рассматривается планирование маршрута для мобильного колесного робота, а также применение машинного обучения с учителем для решения задачи классификации с использованием метода опорных векторов и наивного байесовского классификатора.

**Мобильные колесные роботы**

В настоящее время широко используются колесные роботы с дифференциальным приводом и роботы с реечно-зубчатым приводом. Применение того или иного вида робота зависит от места применения и назначения [1]. Отличительной чертой колесных роботов с дифференциальным приводом является простота их конструкции и хорошая маневренность [2] за счет малого количества колес по сравнению с роботами, где используется реечно-зубчатый привод, у которых большее количество колес увеличивает проходимость [3].

Роботы с дифференциальным приводом имеют два колеса и один поддерживающий ролик, а также два мотора, каждый из который отвечает за управление соответствующим колесом. Включение моторов с разной мощностью обеспечивает изменение направления движения робота [1]. Для прямолинейного движения необходимо вращение колес с одинаковыми скоростями. В случае необходимости разворота на месте требуется установка скоростей равных по модулю, но разных по направлению. Другие комбинации скоростей приводят движение робота к движению по дуге.

Как и роботы с дифференциальным приводом, роботы с реечно-зубчатым приводом имеют два мотора, но один из нужен для вращения колес, а другой для совершения поворотов. Такие роботы не могут развернуться на месте, а при постоянной скорости и угле поворота колес происходит движение по дуге окружности. Следует отметить, что для роботов, использующий данный вид привода, необходим задний дифференциал и поворот колес на разный угол [4], поскольку при движении по окружности путь, пройденный колесами со стороны центра окружности будет меньшим, чем путь, пройденный колесами с внешней стороны, что решается с использованием принципа Аккермана (рис. 1).

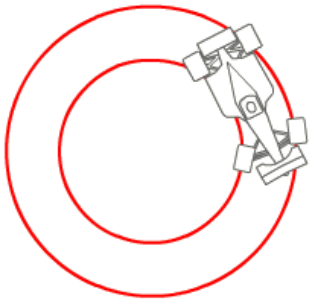


*Рис. 1 Принцип Аккермана*

**Движение и состояние робота на плоскости**

В самом простом случае местоположение робота можно описать вектором $X$ [5]

$$X = \begin{cases} x \\ y \\ \theta \end{cases} \quad (1)$$

где $x$, $y$ – координаты точки в мировой системе координат, являющейся серединой оси вращения колес;

$\theta$ - угол поворота между мировой системой координат и системой координат робота.

**Интегральное движение на плоскости**

При перемещении робота на плоскости можно найти весь его путь, просуммировав пройденные отрезки. Чтобы определить дальнейшее перемещение до следующей точки при сохранении угла поворота, необходимы произвести расчеты

$$\begin{pmatrix} x_{new} \\ y_{new} \\ \theta_{new} \end{pmatrix} = \begin{pmatrix} x + D\cos\theta \\ y + D\sin\theta \\ \theta \end{pmatrix} \quad (2)$$

где $D$ – пройденное расстояние.

В случае поворота на определенный угол необходимо произвести следующие расчеты

$$\begin{pmatrix} x_{new} \\ y_{new} \\ \theta_{new} \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta + \alpha \end{pmatrix} \quad (3)$$

где $\alpha$ – новый угол.

**Планирование маршрута**

В случае, когда роботу известно местоположение точки и ее отношение к мировой системе координат, это дает возможность прокладывать путь вдоль заранее определенных точек. Производится планирование различных маршрутов и в итоге выбирается тот, который обеспечивает минимальные затраты как со стороны затрачиваемого времени, так и со стороны потребления энергии. Поскольку мобильный робот стремится сократить минимальное общее пройденное расстояние, то он будет сразу поворачивать к следующей точке и идти прямо к ней. Этого можно достигнуть с помощью использования прогнозирующего управления, в функциях и методах которого могут применяться полиномы Ньютона, Лагранжа, а также метод наименьших квадратов и тригонометрических функций.

**Использование сенсоров**

В зависимости от условий использования и места применения используются различные сенсоры для приема информации из внешней среды. К примеру, для колесного робота используют ультразвуковые датчики, лидары, камеры с широким углом обзора [6], спектрометры [7]. Основной задачей является их правильное расположение на мобильном роботе для эффективного сбора данных и ориентирования робота в пространстве.

**Использование наборов данных**

Кроме использования сенсоров так же можно работать с уже готовыми наборами данных. К примеру, наборы данных KITTI [8] или ApolloScape [9] содержат данные с камер и лидаров и могут использоваться для мобильных роботов, преследующих цель реализации автономного вождения. Для различных задач классификации возможно использование наборов данных, содержащих в себе большое количество фотографий или видеозаписей. Необходимость использования готовых наборов данных может возникать при решении задачи, требующей идентификации тех или иных объектов, к примеру, препятствий на пути робота. С помощью распознавания объектов становится возможным улучшение планирования маршрута

робота или же его облегчение его управления. Так, при распознавании жестов с помощью готового набора данных можно жестикулируя управлять движением робота, считывающего движения с помощью камеры на мобильном устройстве.

**Применение машинного обучения**

Машинное обучение используется в том случае, когда нет возможности для конкретного и точного описания решения некоторой задачи. Если же такое решение существует, то его необходимо лишь запрограммировать. Так, на практике применяется машинное обучение с учителем и без учителя, а также с подкреплением.

Довольно часто применяется машинное обучение с учителем, где в роли учителя выступают готовые наборы данных и собранные данные с различных датчиков. Машинное обучение с учителем включает в себя решения задач классификации и регрессии. Задача классификации состоит в получении категориального ответа на основе набора признаков, а регрессии – в прогнозе на основе выборки объектов с различными признаками. Для решения задач обучения с учителем используются такие методы, как наивный байесовский классификатор и метод опорных векторов. Рассмотрим каждый из них на применении к мобильному роботу, осуществляющему передвижение исходя из распознавания жестов для его управления на основе имеющегося готового набора данных и получения изображений жестов с камеры мобильного устройства, что решается с помощью решения задачи классификации. Так, модель обучается на готовом наборе данных и получает изображения жестов с камеры, которые и управляют движением робота.

Метод опорных векторов состоит в построении гиперплоскости [11], разделяющей объекты выборки оптимальным способом, чтобы расстояние между разделяющей гиперплоскостью и объектами разделяемых классов был как можно больше, что позволяет уменьшить среднюю ошибку классификатора. К примеру, если X – входящее множество объектов (получаемые изображения с камеры), а У – готовый набор данных, то задача состоит в построении такой гиперплоскости, которая бы аппроксимировала целевую функцию на всем множестве X. К достоинствам данного метода следует отнести его простоту в применении и понятность алгоритма, а к недостаткам – склонность к шумам, что проявляется во влиянии выбросов в исходных данных на получение распределяющей гиперплоскости, а следовательно, и к возможным ошибкам при классификации того или иного объекта.

Суть метода наивного байесовского классификатора заключается в независимом рассмотрении входящих данных (признаков, получаемых с изображения камеры) и в случае, если те или иные признаки встречаются чаще при той или иной категории, то изображение считается принадлежим соответствующей категории [12]. Следует отметить простоту данного алгоритма классификации и эффективную работу даже при небольшом готов наборе данных, но недостатком данного алгоритма является низкое качество классификации в задачах, в которых признаки зависимы.

Так, при небольшом количестве входных данных и слабой зависимостью между признаками лучше использовать наивный байесовский классификатор, в обратном случае, предпочтительнее является применение метода опорных векторов.

К примеру, для решения задачи классификации [7] были использованы данные со спектрометра и камеры, а также набор данных с большим количеством изображений, в результате одновременное использование данных с камеры и набора данных позволило достичь наилучших результатов. То есть, можно сделать вывод, что обучение модели на основе собранных данных с датчиков и использование готовых наборов данных наиболее эффективно.

**Выводы**

В результате данной работы было выяснено, что мобильные роботы с дифференциальным приводом более просты в конструкции по сравнению с мобильными роботами с реечно-зубчатым приводом, однако за счет меньшего количества колес их применение на труднопроходимых плоскостях является неэффективным. Использование сенсоров дает возможность роботу получать информацию о внешней среде, однако их совместное использование с готовыми наборами данных обеспечивает наилучший результат. Для нахождения оптимального маршрута используется прогнозирующее управление, в

функциях и методах которого применяются полиномы Ньютона, Лагранжа, наименьших квадратов и тригонометрических функций. При решении задачи классификации машинного обучения с учителем используются методы оптимальных векторов и наивный байесовский классификатор. Наивный байесовский классификатор лучше использовать для небольших наборов готовых данных и слабой зависимости между признаками, а метод оптимальных векторов – в противном случае.

ЛИТЕРАТУРА

1. Денегин В.В. Виды мобильных роботов. Центр научного сотрудничества «Интерактив плюс». 2018. С. 2.
2. Давыдов О.И., Платонов А.К. Алгоритм управления дифференциальным приводом мобильного робота РБ-2 // Препринты ИПМ им. М.В.Келдыша. 2015. № 25. С. 2.
3. Dominic Baril, Vincent Grondin, Simon-Pierre Deschłnes, Johann Laconte, Maxime Vaidis, Vladim´ır Kubelka, Andr Gallant, Philippe Gigure, Franois Pomerleau. Evaluation of Skid-Steering Kinematic Models for Subarctic Environments. 17th Conference on Computer and Robot Vision (CRV), Ottawa, Canada. 2020. С. 1.
4. Mitchell, W., Staniforth, A., and Scott, I. Analysis of Ackermann Steering Geometry. Motorsports Engineering Conference & Exposition. 2006. С.2
5. Андрей А. Описание движения мобильного робота. 2014. URL: http://robotosha.ru/robotics/robot-motion.html (Дата доступу: 02.04.2020).
6. Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou. A2D2: Audi Autonomous Driving Dataset. 2020. С.
7. Zackory Erickson, Eliot Xing, Bharat Srirangam, Sonia Chernova, Charles C. Kemp. Multimodal Material Classification for Robots using Spectroscopy and High Resolution Texture Imaging. Georgia Institute of Technology. 2020. С. 5 - 8.
8. Andreas Geiger, Philip Lenz, Christoph Stiller, Raquel Urtasun. Vision meets robotics: The KITTI dataset. Karlsruhe Institute of Technology and Max Planck Institute for Intelligent Systems Tübingen, Germany. 2013. С. 1.
9. Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, Ruigang Yang. The ApolloScape Dataset for Autonomous Driving. 2019. С. 2, 14.
10. В.Н. Шашок, С.И. Филиппов, Д.В. Багаев, А.Н. Малышев, А.А. Кобзев, В.А. Соловьев, Ю.Е. Мишулин, В.А. Немонтов. Планирование маршрута движения наземным роботом в недетерминированной местности. АО «ВНИИ «Сигнал». 2015. С. 149-159.
11. Д.В. Федотов. О решении задачи классификации методом опорных векторов. Сибирский государственный аэрокосмический университет имени академика М.Ф. Решетнева. 2013. С. 77-79.
12. Е.В. Михайлов, С.В. Сай. Выделение леса на космических снимках с помощью методов машинного обучения. 2017. ТУСУРа. С. 4.

**СЕРДЮК Софья Андреевна** – студентка кафедры электроники и управляющих систем, Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 4, Харьков-22, Украина, 61022; e-mail: sf.srd.31@gmail.com; ORCID: 0000-0003-1512-0374.

**МАЛАХОВА Марина Олеговна** – к.т.н., ст. преподаватель кафедры электроники и управляющих систем, Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 4, Харьков-22, Украина, 61022; e-mail: maryna.malakhova@karazin.ua; ORCID: 0000-0001-5082-5279.

**УДК 004.4**

МАЛЫГА И.Е.

# ПРОГРАММНАЯ СТАНДАРТИЗАЦИЯ ОБРАБОТКИ GRAPHQL ЗАПРОСОВ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ GRAPHENE

**Принцип GraphQL**

Прежде чем рассматривать GraphQL, необходимо уделить внимание базе, на которой создан этот инструмент. Речь идёт об SQL — structured query language или структурированном языке запросовю. [1,4,5]

SQL — язык программирования, который применяется для создания, изменения и управления данными в базах данных. Этот язык поддерживает четыре оператора запросов: SELECT, INSERT, UPDATE и DELETE. С помощью этих операторов вы можете получить из базы данных (БД) необходимые сведения.

Например, когда необходимо «достать» из БД всех пользователей с именем Maria, это можно сделать с помощью запроса:

SELECT * FROM USERS WHERE FirstName = “Maria”

Чтобы решить эту задачу с помощью REST, потребуются такие действия:

1. Определить endpoint на сервере, который необходим для получения из базы данных пользователей с fname Maria.
2. Определить общий endpoint для получения всех пользователей с последующей фильтрацией на стороне клиента.

Фильтрация выглядит так:
users.filter(user => user.fname === “Maria”);

В этом подходе есть проблема. Несмотря на удобство SQL и простой способ взаимодействия клиента с SQL с помощью REST, операция получается сложной. Первый подход нельзя масштабировать, так как нет возможности создать endpoint для каждого пользователя. Второй подход слишком дорогой, так как требует дополнительных действий на стороне клиента.

А теперь представьте инструмент, который объединяет возможности SQL и REST на стороне клиента. Этим инструментом является GraphQL. GraphQL берёт идеи, разработанные для манипуляции данными в БД, и использует их в вебе. Поэтому с помощью одного запроса GraphQL можно получить сразу все необходимые данные.

**Виды запросов GraphQL**

Работа GraphQL базируется на трех видах запросов [1,4,5]:

1. Query — с помощью запросов GraphQL получает необходимые данные с сервера. Query в GraphQL — аналог GET в REST. Запросы — строки, которые отправляются в теле HTTP POST.
2. Mutation — ещё один корневой тип. С его помощью можно добавлять данные в БД. Mutation — аналог POST и PUT в REST.
3. Subscription — третий тип операций в GraphQL. С его помощью клиент слушает изменения в БД в режиме реального времени. Работа подписок основывается на вебсокетах.

Далее рассмотрим стандартизацию обработки запросов-мутаций.
Разберем пример кода на рис. 1:

```
class CreateDocumentMutation(graphene.Mutation):
    class Arguments:
        name = graphene.String(required=True)
        description = graphene.String()
        upload = Upload(required=True)
        file_type = graphene.String()
        page_id = graphene.ID(required=True)
        system_file_name = graphene.String(required=True)

    success = graphene.Boolean()
    errors = graphene.List(graphene.String)

    def mutate(self, info, name, description, upload, file_type, page_id, system_file_name):
        errors = list()
        success = False
        if Document.objects.filter(name=name).count() > 0:
            errors.append('Document with specified name already exists.')

        if not description or not upload:
            errors.append('Description and upload fields are required.')

        if not info.context.user:
            errors.append('You have to be logged in to perform this action.')

        if not errors:
            Document.objects.create(name=name, description=description, upload=upload, file_type=file_type,
                                    page_id=page_id, system_file_name=system_file_name, uploaded_by=info.context.user)
            success = True
        return CreateDocumentMutation(errors=errors, success=success)
```

*Рис. 1 Фрагмент кода мутации*

CreateDocumentMutation создает объект модели Document, используя переданные данные и возвращает ошибки, если какие-то условия не соблюдены. [2,3]

Стоит лишь взглянуть на этот код, как становится видна его нерасширяемость и неподдерживаемость из-за следующих причин:

1. Слишком большое количество аргументов у функции mutate, число которых может увеличиться еще, если разработчику необходимо добавить еще поля, подлежащие редактированию.
2. Чтобы мутации выглядели одинаково со стороны клиента, они должны возвращать errors и success, чтобы всегда можно было понять их статус и чем он обусловлен.
3. Проверка прав доступа в мутации. Мутация не должна происходить, если пользователь не имеет прав на это.
4. Непредсказуемый набор ошибок: если у вас нет исходного кода или документации, то вы не узнаете, какие ошибки может вернуть эта мутация, так как они не отражены в схеме.
5. Слишком много шаблонных проверок ошибок, которые проводятся непосредственно в методе mutate, который предполагает *создание* данных, а не разнообразные проверки. Идеальный mutate должен состоять из одной строки – вызова функции создания документа.

**Выводы**

Мы рассмотрели только один из трех видов запросов GraphQL – Mutation, однако даже в этом примере мы можем увидеть большое количество проблемных моментов, которых можно избежать с помощью приемов ООП и создания кодовой базы, которая поможет переложить выполнение рутинных задач в отдельные вспомогательные функции, облегчив при этом масштабирование основного кода мутации и его поддержку.

Шаги, которые можно предпринять для оптимизации мутаций:

- использование классов для группирования входящих аргументов функций, что позволит легко управлять возрастающим количеством аргументов, а также повторно использовать эти классы в качестве аргументов других мутаций;
- создание базового класса мутации, который смогут унаследовать все мутации приложения;
- создание функции в базовом классе мутации, которая будет содержать в себе список возможных ошибок во время выполнения обработки запроса, при возникновении одной из описанных ошибок обработка запроса будет прерываться автоматически и клиенту будет отправлен ответ с описанием или кодом ошибки.

ЛИТЕРАТУРА

1. Бэнкс А., Порселло А. GraphQL: язык запросов для современных веб-приложений. Питер, 2019. 240 с.
2. Документация библиотеки graphene.
3. URL: https://docs.graphene-python.org/projects/django/en/latest/
4. Документация по языку программирования python. URL: https://docs.python.org/3/
5. Официальная документация по GraphQL. URL: https://graphql.org/
6. GraphQL – новый взгляд на API. Ч1. URL: https://habr.com/ru/post/343872/

**Малыга Игорь Евгениевич** — бакалавр, студент кафедры теоретической и прикладной системотехники, Харьковский национальный университет имени В.Н. Каразина, площадь Свободы, 4, Харьков-22, Украина, 61022; e-mail: igormalyga@gmail.com;
ORCID: 0000-0002-5708-7739.

Научные интересы:

- *Разработка веб-приложений.*

**УДК 539.534.9:523.23**

МАРЧЕНКО И.Г., ПАВЛЕНКО В.И.

# МОДЕЛИРОВАНИЕ ПОВЕДЕНИЯ ПРОФИЛЕЙ ДЕФЕКТООБРАЗОВАНИЯ ОТ УГЛА ПАДЕНИЯ ИОНОВ $AL^+$, ОБЛУЧАЮЩИХ НАНОСТРУКТУРНУЮ ПЛЕНКУ CU

## ВВЕДЕНИЕ

В работах [1–2] было показано, что бомбардировка поверхности медных подложек ионами металлов меняет структурно-фазовые состояния подложек и может приводить к заметному улучшению механических свойств покрытий. Перспективным направлением повышения адгезионной прочности пленок является предварительное наноструктурирование их [2]. Кроме того известно, что осаждаемые вакуумные пленки обладают высокой пористостью [3]. Вместе с тем присутствие ионов, осаждаемых на пленки, приводит к повышению плотности осаждаемых пленок [4].

Важным фактором для формирования структуры пленки под воздействием низкоэнергетических (E≤1–3 кэВ) потоков ионов являются зависимости профилей залегания точечных дефектов от энергии падающих ионов [5]. В работах [5–6] было исследовано изменение профилей при нормальном падении падающих ионов на поверхность пленки. В то же время, влияние угла падения при ионном облучении поверхности пленки на эволюцию профилей дефектообразования точечных дефектов не изучались.

В связи с развитием технологий получения наноструктурных пленок особую актуальность приобретает компьютерное моделирование процессов, происходящих в пленках под облучением. Моделирование дает конкретные количественные оценки процессов или явлений, которые протекают в материале при его облучении, позволяет прогнозировать новые свойства, помогает существенно уменьшить затраты на физические эксперименты.

Целью данной работы являлось изучение изменения профилей залегания точечных дефектов в облучаемой ионами $Al^+$ пленке Cu, в зависимости от угла падения ионов на пленку.

## МЕТОДИКА МОДЕЛИРОВАНИЯ

Моделирование проводилось с помощью компьютерного комплекса SPURT.CRIS [7]. SPURT.CRIS, это программы, объединенные в большой комплекс, которые созданы для моделирования процессов первичного дефектообразования в сложных неравновесных системах в процессе облучения наноструктурной пленки низкоэнергетичными (E≤1–3 кэВ) потоками ионов в широком интервале углов облучения ($\alpha=0^0$–$80^0$).

SPURT.CRIS – это программа, реализующая методы Монте-Карло и основана на приближении метода парных столкновений. В методе Монте-Карло взаимодействие движущейся частицы с атомами материала пленки представляют как случайная последовательность парных столкновений. В программе SPURT.CRIS, объединены алгоритмы аморфности, слоистости и поликристалличности мишени. Т.е. реализована следующая модель наноструктурной пленки: пленка состоит из нанокристаллитов, произвольно ориентированных друг по отношению к другу и расположенных в аморфной матрице. Общая теория моделирования физических процессов в наноструктурной пленке по программе SPURT.CRIS детально описана в работе [7].

## РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ И ОБСУЖДЕНИЯ

Программа SPURT.CRIS позволяет получать и исследовать (в зависимости от энергии ионов и угла их падения $\alpha$ на пленку) такие важные параметры, развивающегося в процессе облучения, каскада столкновений, как: профили пространственного распределения вакансий и междоузельных атомов пленки. А также их разностный спектр; определять области максимальных залеганий (концентраций) точечных дефектов; получать количественные характеристики обедненных зон и т.д.

Результаты моделирования системы $Al^+$ –> Cu представлены на рис. 1–2. На рис. 1 показаны профили пространственного распределения вакансий $G_{vac}(x)$, а на рис. 2 – профили пространственного распределения междоузельных атомов $G_{int}(x)$ Cu при различных углах падения α ионов $Al^+$, бомбардирующих пленку из модельной наноструктурной меди с энергией 1 кэВ.

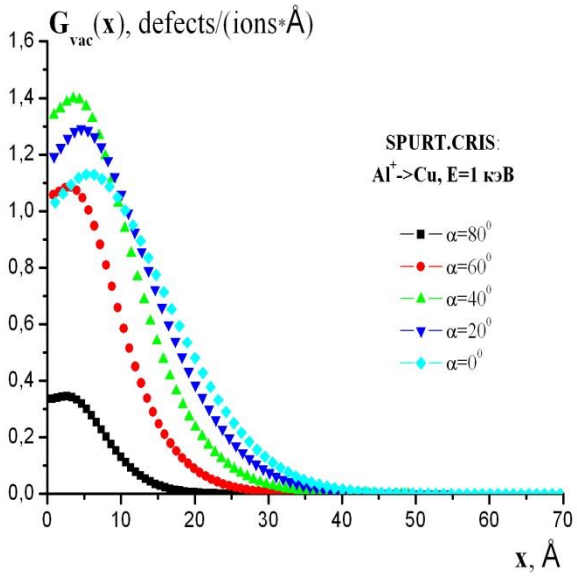



*Рис. 1. Профили пространственного распределения $G_{vac}(x)$ вакансий при облучении наноструктурной меди ионами $Al^+$ с энергией E=1,0 кэВ при углах падения ионов $\alpha=0^0$ - $80^0$*

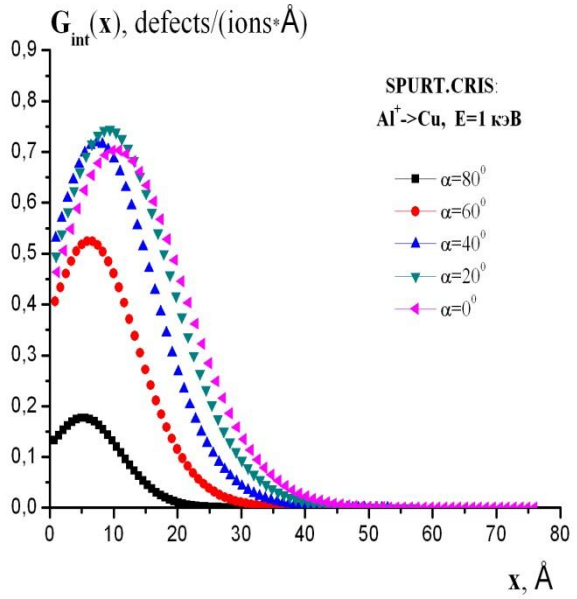



*Рис. 2. Профили пространственного распределения $G_{int}(x)$ междоузельных атомов при облучении наноструктурной меди ионами $Al^+$ с энергией E=1,0 кэВ при углах падения ионов $\alpha=0^0$ - $80^0$*

Установлено, что с увеличением угла облучения α пленки наблюдается тенденция к уменьшению значений профилей пространственного распределения, как по концентрации дефектов на заданной глубине, так и по глубине проникновения точечных дефектов вглубь мишени.

Максимальные значения концентраций вакансий, при исследованных углах падения ($\alpha=0^0$–$80^0$), достигаются в интервале углов $\alpha=20^0$–$40^0$ ($G_{vac}^{max}(x)$=1,29 vac/(ions x Å) при $\alpha=20^0$ на глубине l ~ 5,5 Å и $G_{vac}^{max}(x)$=1,4 vac/(ions x Å) при $\alpha=40^0$ на глубине l ~ 3,5–4,3Å).

Максимальное значение концентрации собственных междоузельных атомов Cu, при тех же углах падения, $G_{int}^{max}(x)$=0,75 int/(ions x Å) достигается при $\alpha=20^0$ на глубинах l~11,0Å. от поверхности пленки.

Максимальная глубина дефектообразования $\delta_{max}$, т.е. глубина модифицированного слоя, образуется при нормальном ($\alpha=0^0$) падении ионов на пленку. В рамках данных исследований, в случае $\alpha=0^0$ величина $\delta_{max}$ составляет ~ 77Å – для собственных междоузельных атомов и ~ 70Å – для вакансий.

Графики «разностного спектра» вакансий и междоузлий, т.е. профили функции $G_{vi} = G_{vac}(x) - G_{int}^{max}(x)$ демонстрируются на рис. 3.

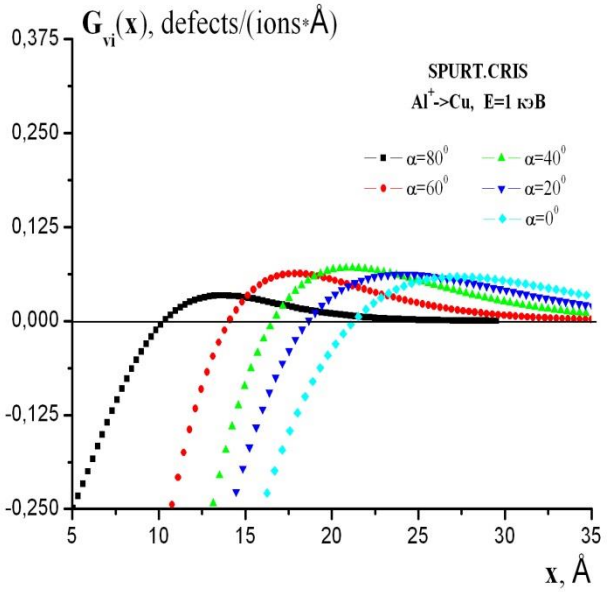



*Рис. 3. Фрагмент профиля функции $G_{vi}(x)$ при облучении наноструктурной меди ионами $Al^+$ с энергией E=1,0 кэВ при углах падения ионов $\alpha=0^0$-$80^0$*

Установлено, что начальные разностные профили пространственного распределения вакансий и собственных междоузельных атомов $G_{vi}(x)$ обнаруживают две выраженные области. Вблизи поверхности мишени $G_{vi}<0$, что связано с обогащением поверхностных слоев вакансиями за счет активно идущих процессов распыления, а также прямого выбивания атомов в более глубокие слои мишени. В более глубоких слоях мишени ($x_1=\lambda_{max}$ ~ 22 ангстрем для $\alpha=0^0$ и $x_5=\lambda_{max}$ ~ 11 ангстрем для $\alpha=80^0$) $G_{vi}>0$, что указывает на преимущественное залегание в этих слоях собственных междоузельных атомов и, следовательно, на обогащение этой области веществом из междоузельных атомов Cu.

Из рис. 3 видно, что с увеличением угла $\alpha$ (от $\alpha_1=0^0$ до $\alpha_5=80^0$) уменьшается величина обедненной зоны $\lambda$. $\lambda_{max}$ соответствует глубине от поверхности мишени, после прохождения которой, в поврежденной области преобладают собственные междоузельные атомы меди в сравнении с вакансиями (Табл. 1).

*Табл. 1 Зависимость $\lambda_{max}$ от угла падения ионов $Al^+$ на мишень Cu*

| $\alpha$ | $0^0$ | $20^0$ | $40^0$ | $60^0$ | $80^0$ |
|---|---|---|---|---|---|
| $\lambda_{max}$, анг. | 21,53 | 18,66 | 16,79 | 14,13 | 10,89 |

ВЫВОДЫ

С использованием программного комплекса SPURT.CRIS, сгенерированы и исследованы профили распределения вакансий и собственных междоузельных атомов, создаваемых в модельной наноструктурной пленке Cu при облучении ионами $Al^+$ с энергией E=1 кэВ.

Исследовано поведение полученных профилей в зависимости от угла падения $\alpha$ падающих ионов с целью определения оптимальных значений углов падения ионов, при

которых достигается максимальная концентрация первичных дефектов в пленке при облучении.

Компьютерные расчеты демонстрируют условия и параметры, при которых можно добиться требуемой концентрации дефектов (или вводимого импланта) на нужной глубине. И ее (концентрацию) можно целенаправленно регулировать путем варьирования угла падения ионов пучка.

Установлено, что первоначальные разностные профили распределения вакансий и собственных междоузельных атомов $G_{vi}(x) = G_{vac}(x) – G_{int}(x)$ имеют две ярко выраженные различные области повреждения. Вблизи поверхности мишени $G_{vi}<0$, что связано с обогащением поверхностных слоев вакансиями за счет активно идущих процессов распыления, а также прямого выбивания атомов в более глубокие слои мишени. В более глубоких слоях объема пленки для $Al^+$ –> Cu: начиная с x~11Å при $\alpha=80^0$ и заканчивая x~22Å при $\alpha=0^0$.

ЛИТЕРАТУРА

1. Калашников М.П., Нейфельд В.В, Сергеев В.П., Федорищева М.В. Изменение структурно-фазового состояния поверхностного слоя медной подложки при бомбардировке ионами титана. *Известия Российской академии наук. Серия физическая*. 2014. Том 78. №8. С. 937-939.
2. Панин А.В., Шугуров А.Р., Казаченок М.С., Сергеев В.П. Влияние наноструктурирования подложки Cu на разрушение теплозащитных покрытий Si-Al-N при одноосном растяжении. *Журнал технической физики*. 2012. Том 82. Вып. 6. С. 44-47.
3. Guglya A.G., Marchenko I.G. Ion beam-assisted deposition. Comprehensive guide for nanocoatings technology. *Nova Science Publishers.* New York. 2015. V. 1. P. 45-69.
4. Marchenko I. G., Neklyudov I.M. Film nanostructure formation during low-temperature PVD deposition using partially ionized atomic fluxes. *Journal of Physics: conference series*. 2008. V. 113. P. 2-8.
5. Bakai A. S., Sleptsov S.N., Zhukov A.I., Marchenko I.G., Sleptsov A.N. Mathematical modeling of the densification of niobium film deposited from self-ion-atomic fluxes *Met. Phys. Adv. Tech.* 1996. V. 15. P. 1329-1342.
6. Bakai A. S., Zhukov A.I., Sleptsov S.N., Marchenko I.G., Sleptsov A.N., Reznichenko A.N. Low-temperature densification of chromium films induced by bombardment with argon and chromium ions. *Met. Phys. Adv. Tech.* 1996. V. 16. P. 99-109.
7. Павленко В.И., Марченко И.Г. Компьютерное моделирование профилей имплантированных ионов $Al^+$ в наноструктурную пленку Cu. *Вопросы атомной науки и техники. Серия: «Физика радиационных повреждений и радиационное материаловедение»*. 2017. №4 (110). С. 32-38.

**МАРЧЕНКО Иван Григорьевич** – д.ф.-м.н., ведущий научный сотрудник Национального научного центра «Харьковский физико-технический институт», профессор кафедры физики нетрадиционных энерготехнологий и экологии Харьковского национального университета имени В.Н. Каразина, ул. Академическая, 1, Харьков-108, Украина, 61108; e-mail: march@kipt.kharkov.ua; ORCID: 0000-0003-1341-4950.

Научные интересы:

– *компьютерное моделирование процессов происходящих в твердых телах, стохастические процессы, физика поверхности, физика радиационных повреждений.*

**ПАВЛЕНКО Владимир Иванович** – ведущий инженер-исследователь Национального научного центра «Харьковский физико-технический институт», ул. Академическая, 1, Харьков-108, Украина, 61108; e-mail: ruslana_olirna2005@ukr.net; pavlenko@kipt.kharkov.ua; ORCID: 0000-0003-0210-3268.

Научные интересы:

– *математическое моделирование физических процессов происходящих в покрытиях (пленках) при ионном облучении, физика поверхности.*

**УДК 004.94**

МАТВИЕНКОВ А.А., ХРУСЛОВ М.М.

# РАЗРАБОТКА АВТОМАТИЧЕСКОЙ СИСТЕМЫ ИНФОРМИРОВАНИЯ СТУДЕНТОВ И АНАЛИЗА УЧЕБНОГО ПРОЦЕССА

**Введение**

В настоящее время вопрос документооборота, анализа текущего положения внутри высшего учебного заведения, а также возможностей дистанционного информирования студентов является одним из основополагающих. Благодаря развитию современных технологий, появляются новые возможности, которые помогают упростить и оптимизировать рабочий процесс, за счет упразднения бумажного документооборота и перевода его в электронный вид. В результате такого перехода открывается возможность быстрого и своевременного, обмена документацией между студентами и администрацией высшего учебного заведения, необходимость хранения бумажных документов становится не актуальной, значительно экономится рабочее время сотрудников, снижаются затраты на печать и хранение документов. Кроме того, на основе электронных документов упрощается реализация мониторинга посещаемости и успеваемости групп или отдельно взятых студентов, так же реализуется возможность быстрого и своевременного информирования студентов об изменениях в учебном плане или же в расписании занятий. [1]. Другими словами, при помощи современных технологий, получается удобный и многофункциональный планировщик задач [2]. Для этого отлично подойдет программный продукт, предложенный в данной работе, который может в зависимости от поставленной задачи, своевременно информировать студентов о изменениях в учебном процессе, быстрая и своевременная подача или получение необходимых документов, а также анализ посещаемости и успеваемости конкретной группы или студента.

**Исходные данные**

В качестве исходных данных берутся данные, получаемые при заполнении уполномоченными пользователями специальных форм после их авторизации в системе. Данный этап является наиболее важным, поскольку от полученных данных зависит дальнейший анализ, на базе которого рассчитывается успеваемость студентов и составляются соответствующие графики и таблицы. Основными данными для расчета являются: ФИО, группа, балы успеваемости, посещаемость занятий.

**Расчет и анализ результатов**

При анализе результатов, за основу оберутся такие показатели успеваемости как: абсолютная успеваемость, качественна успеваемость, средний балл, средняя успеваемость, а также показатель «человеко-пересдач». Расчеты производятся согласно формулам описанных в [3]:

Средний балл – сумма всех полученных оценок студентом в течении семестра, разделенных на их количество и определяющийся по формуле:

$$R_i = \frac{\sum_i r}{n_i}, \quad (1)$$

где $R_i$ - средний балл, $\sum_i r$ - сумма баллов, $n_r$ - количество оценок

Качественна успеваемость – процент студентов, успешно справившихся с учебным планом и процент студентов, имеющих хорошие и отличные оценки от студентов, получивших положительные оценки по дисциплинам, определяем по формуле:

$$KYc_i = \frac{n_{si}}{n_i} * 100\% \ , \tag{2}$$

где $KYc_i$ - качественная успеваемость студентов, $n_{si}$ - количество студентов, сдавших все экзамены на «хорошо» и «отлично», $n_i$ - общая численность студентов сдавших сессию.

Абсолютная успеваемость – соотношение успешно справившихся с учебным планом студентов, от общей численности учащихся, определяется формулой:

$$AYc_i = \frac{(n_{s1} + n_{s2} + n_{s3}) * 100\%}{n_i}, \tag{3}$$

где $AYc_i$ - абсолютная успеваемость, $n_{si}$ - количество студентов «отлично» , $n_{s2}$ - количество студентов «хорошо», $n_{s3}$ - количество студентов «удовлетворительно» сдавших сессию, $n_i$ - общее количество учащихся.

**Выводы**

В ходе исследования была предложена система для формирования отчетности, расчета успеваемости и упрощения коммуникации между студентами и факультетом. Данная система позволяет оперативно получать точную информацию по интересующим вопросам относительно успешности студентов и на основании полученных результатов принимать решение. Был выполнен анализ и изучение, существующих аналогов и методов, выявлены их недостатки, с учетом полученных данных была разработана данная система, учитывающая специфику образовательного процесса. Поэтому данный программный продукт является наиболее эффективным для образовательного учреждения.

ЛИТЕРАТУРА

1. Электронный документооборот [Электронный ресурс]. – Режим доступа: http://www.docflow.ru/edu/glossary/detail.php?ID=27946.
2. Автоматизация бизнес-процессов [Электронный ресурс]. – Режим доступа: https://www.terrasoft.ua/page/business-process-automation/.
3. Основные показатели успеваемости студентов [Электронный ресурс]. – Режим доступа: https://studfile.net/preview/11790357/page:4/.
4. Современные системы автоматизации производства [Электронный ресурс]. – Режим доступа: https://wiseadvice-it.ru/o-kompanii/blog/articles/sovremennye-sistemy-avtomatizacii-proizvodstva/.
5. Автоматизация бизнес-процессов [Электронный ресурс]. – Режим доступа: https://www.terrasoft.ua/page/business-process-automation/.

**МАТВИЕНКОВ Анатолий Андреевич –** студент факультета компьютерных наук Харьковского национального университета имени В.Н. Каразина.

Научные интересы:

*– экспериментальные исследования характеристик излучения антенн.*

**ХРУСЛОВ Максим Михайлович –** к.ф.-м.н., доцент кафедры электроники и управляющих систем Харьковского национального университета имени В.Н. Каразина.

Научные интересы:

*– экспериментальные исследования характеристик излучения антенн.*

**УДК 004.056.55**

МІГАЛЬ Д.О., ЄСІНА М.В.

# ЕЛЕКТРОННЕ ГОЛОСУВАННЯ НА ОСНОВІ ТЕХНОЛОГІЇ БЛОКЧЕЙН

**Вступ**

Електронне голосування являє собою подальший розвиток існуючих технологій голосування, що використовуються в Україні та інших демократичних країнах. Такий складний процес як волевиявлення потребує ретельної уваги до кожного з аспектів, які гарантує Конституція та відповідні закони. Використання електронного голосування (у подальшому ЕГ) спрямовано на покращення якості голосування у порівнянні з існуючою системою. ЕГ спрямоване в першу чергу на підвищення рівня прозорості, швидкості, анонімності, достовірності та стійкості до навмисних та ненавмисних завад, атак, тощо.

Загалом розрізняють онлайн та офлайн електронне голосування [1]. Відрізняються вони необхідністю факту фізичної наявності виборця для волевиявлення. У разі необхідності присутності виборця, ЕГ називають офлайн, якщо голосування може бути здійснено дистанційно – онлайн. В свою чергу кожний з цих типів представлений у декількох варіантах:

Офлайн: повністю електронне; з електронним підрахунком голосів.

Онлайн: за допомогою Інтернет-ресурсу; за допомогою мобільного телефону.

Перспективними, спираючись на кількість мобільних телефонів та ступінь розвитку мобільної інфраструктури, виглядають системи онлайн голосування. Легкість доступу до онлайн ресурсів та розвиток криптографії у цій сфері робить системи ЕГ з використанням Інтернет-ресурсу більш бажаними у порівнянні з системами, що використовують канали мобільного зв'язку, і тому ми докладніше розглянемо саме їх.

**Голосування онлайн**

До переваг онлайн голосування з використанням Інтернет-ресурсів відносяться [2]: можливість дистанційного голосування; забезпечення більш комфортних умов; можливість використання технології для проведення голосування в інших сферах/питаннях; прозорість виборів; вищий рівень залученості молоді; вартість проведення виборів; швидкість та достовірність підрахунку голосів.

Онлайн голосування вирішує деякі фундаментальні проблеми звичайного голосування, в свою чергу успадковуючи деякі недоліки та створюючи деякі нові вразливості. Голосування онлайн є складним для людей, які лише іноді взаємодіють з технологіями або знаходяться у більш-менш ізольованих регіонах. Існує також можливість компрометації бази даних та саботажу виборчих процесів. Через безпосередню роботу з персональними даними, забезпечення збереження їх секретності становить важливу задачу.

Венеціанська Комісія сформувала ряд рекомендації, які є бажаними до виконання у разі проведення демократичних електронних виборів. Кожна з цих рекомендації має за мету покращити систему ЕГ в одному з аспектів, а тому використання лише частини з них не може гарантувати комплексного покращення рівня безпечності голосування. До цих рекомендації відносять:

- електронне голосування може використовуватися лише за умови, що система є безпечною/захищеною і надійною;
- система електронного голосування повинна бути прозорою, тобто надавати можливість перевірки щодо її функціонування;
- виборці повинні мати нагоду одержати підтвердження свого вибору і виправити його у разі допущення помилки;
- для полегшення перерахунку голосів у разі конфліктної ситуації може передбачатися процедура роздрукування голосів.

**Блокчейн**

Блокчейн – систематизований ланцюг блоків (зв'язаний список), розподілений між усіма вузлами мережі. Кожний блок містить в собі інформацію про усі транзакції, що були оброблені в мережі. Таким чином історія мережі зберігається в кожному з вузлів, що робить фальсифікацію транзакцій значно складнішою.

Важливо відмітити інші технології, які роблять блокчейн систему більш надійним способом зберігання інформації. Геш-функції у блокчейн слугують своєрідним підписом, перевірка якого може слугувати підтвердженням відповідності. В такому разі криптостійкість системи напряму залежить від криптостійкості геш-функції.

Зрозуміло, що використання простих геш-сум не є достатнім для забезпечення необхідного рівня безпечності, тому використовують так зване дерево Мерклі. Дерево Мерклі (рис. 1) це спеціалізована структура даних яка пов'язує геші попередніх блоків з наступними блоками. Виконана ця структура у вигляді двійкового дерева. Спочатку ми беремо пару геш-значень обчислених для певних блоків, на які поділено деякий масив даних та обчислюємо їх загальну геш-суму. В тому разі, якщо пари в блоку немає, він потрапляє на наступний рівень без змін. Обчислення геш-сум пар виконується до тих пір, доки не залишиться лише одне геш-значення. Дерево Мерклі гарантує збереження цілісності та достовірності збереженої інформації.

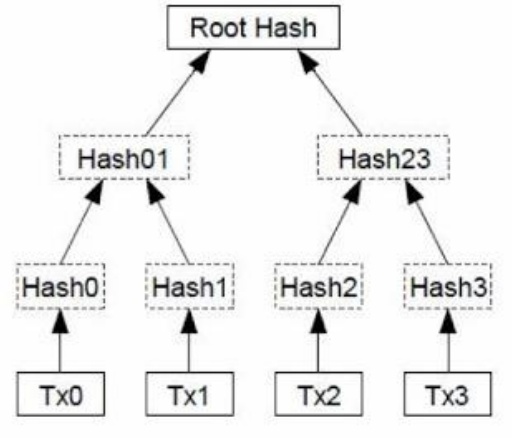



*Рис. 1 Структура дерева Мерклі*

Верхнє геш-значення (*root hash*) потрібно для безпосереднього порівняння. Перш, ніж підтверджувати трансакцію, необхідно завантажити верхній геш з надійного джерела, яке не обов'язково повинно бути звичайним вузлом мережі. Для цього підійде Інтернет-ресурс з надійною та захищеною базою даних. Далі порівнюють отримані геш-значення з перевіреним. Якщо геш-значення не співпадають, це означає що геш-значення було пошкоджено або скомпрометовано. У такому разі потрібно обчислювати геш-значення на іншому вузлі та продовжувати процедуру доки співвідношення не буде знайдене.

**Гомоморфне шифрування**

Гомоморфне шифрування – шифрування основною превагою якого є можливість виконувати певні математичні перетворення з шифротекстами, зашифрований результат яких буде відповідати виконанню відповідної операції з відкритим текстом [3].

Системи можуть бути: цілком гомоморфними; частково гомоморфними.

Частково гомоморфними називають системи, які гомоморфні лише для однієї з алгебраїчних операції (множення(1) чи додавання(2)).

Цілком гомоморфні – відповідно ті, що гомоморфні по обом з цих операції (1, 2).

$$D(E(t_1)\otimes E(t_2))=t_1\times t_2, \quad (1)$$

$$D(E(t_1)\oplus E(t_2))=t_1+t_2, \quad (2)$$

де $t$ – відкритий текст, $E(t)$ – функція шифрування, $D(E(t))$ – функція розшифрування.

Гомоморфне шифрування має великі переваги для її використання у системах електронного голосування. Можливість виконувати операції над зашифрованими даними, зберігаючи анонімними дані виборців.

Існує декілька схем електронного голосування, які використовують гомоморфне шифрування.

Розглянемо криптосистему Бенало, яка є частково гомоморфною за операцією додавання. Кожний з виборців розподіляє свій голос на складові частини, використовуючи відповідну схему гомоморфності по додаванню і відправляє отримані частини представникам обранця. Гоморфність системи гарантує, що при складанні зашифрованих частин результат буде відповідати зашифрованому голосу. Деякий суб'єкт обчислює кінцевий результат голосування, що йому передали представники та отримує дійсний результат голосування.

Саме шифрування у системі Бенало використовує формулу (3)

$$E_r(m)=(y^m u^r) mod\ n, \quad (3),$$

де $u$ – випадкове число $Z_r$ ; $m$ – повідомлення; $p$, $q$ – прості числа, такі, що $r$, $(p–1)/r$, $(q–1)$ є взаємно простими; $y$ обирається таким чином, що

$$y^{(p-1)*(q-1)/p(i)} \neq 1.$$

Стійкість системи базується на складності обчислення лишків великих ступенів.

Система Бенало базується та модифікує систему Гольдвассера-Мікалі.

**Висновки**

1. Електронне голосування є наступним кроком розвитку системи голосування в Україні. Перспективність ЕГ допоможе поліпшити діючу виборчу систему та частково усунути існуючі проблеми.

2. Введення ЕГ допоможе збільшити кількості активних виборців серед молоді, забезпечити комфортніші умови голосування, також і для людей у віддалених регіонах та людей з вадами. Система ЕГ є більш прозорою, надійною та швидкою у порівнянні з існуючою.

3. Зрозуміло, що крім переваг існують також недоліки: комп'ютерні збої, хакерські атаки, ціна, необхідна для розробки системи ЕГ, необхідність мати базову комп'ютерну грамотність та необхідні пристрої.

4. Електронна система голосування є альтернативною і для початку вона має бути опцією, що не нав'язує нові технології і не відлякує виборців. У подальшому введення її як основної поліпшить виборчу систему.

ЛІТЕРАТУРА

1. Готун А., Використання нових інформаційних технологій у виборчому процесі: світовий досвід і практика застосування в Україні, Вісник Київського НУ ім. Шевченка. Філософія і політологія. – 2008. – С. 89-90.

2. Оніпко О., Шляхи вдосконалення технології виборчого процесу, Вісн. Центр. вибор. коміс. – 2006. – Вип. № 4 (6). – С. 67–70. – URL: http://www.cvk.gov.ua/visnyk/pdf/2006_4/visnyk_st_22.pdf.

3. Homomorphic encryption – URL: https://en.wikipedia.org/wiki/Homomorphic_encryption.

**МІГАЛЬ Демид Олексійович** – студент 4 курсу кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків, Харківська область, 61000; e-mail: demidmgl@gmail.com; ORCID: 0000-0002-7400-777X.

Наукові інтереси:

– *криптографія, блокчейн-технології*.

**ЄСІНА Марина Віталіївна** – к.т.н., старший викладач кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків, Харківська область, 61000; e-mail: m.v.yesina@karazin.ua; ORCID: 0000-0002-1252-7606.

Наукові інтереси:

– *захист інформації, постквантова криптографія.*

**УДК 539.3**

МИРОНЕНКО М.Л.

# ВЛАСНІ КОЛИВАННЯ РІДИНИ В ЦИЛІНДРИЧНИХ ОБОЛОНКАХ ПРИ РІЗНИХ РІВНЯХ ГРАВІТАЦІЇ

**Загальні вимоги**

Резервуари, частково наповнені рідиною, яка має вільну поверхню, мають широку сферу промислового використання (наприклад, паливні баки ракетоносіїв або автомобілів, залізничні або автомобільні цистерни для транспортування вантажів у рідкому стані, вантажні танкери тощо). Вивчення динаміки ємностей даного типу, що призводить до коливання рідини у резервуарі, має практичне значення. Жорсткі резервуари, частково заповнені рідиною, зазнають найбільш інтенсивної дії від коливань рідини, що може призвести до втрати стійкості або навіть до руйнування конструкції [1]. Тому дослідження коливань рідини у ємностях довільної форми при частковому заповненні та наявності вільної поверхні є важливим технічним завданням.

**Аналіз останніх досліджень та публікацій**

В умовах слабкого гравітаційного поля враховуються сили поверхневого натягу рідини. Коли сили тяжіння, що утворюють поверхневі хвилі, співмірні з дією сил поверхневого натягу, виникають гравітаційно-капілярні хвилі. Такі складні гідродинамічні задачі виникають в динаміці космічних апаратів, що мають значну кількість рідкого вантажу на борту. Відомо, що капілярно-гравітаційні хвилі мають нелінійний ефект при розповсюдженні по поверхні рідини. Набула розвитку теорія капілярно-гравітаційних хвиль нескінченно малої амплітуди у працях Кельвіна [2] та Релея [3]. Вперше теорію капілярно-гравітаційних хвиль кінцевої амплітуди висвітлено у праці Вільтона [4]. Продовжили розвиток цієї теорії Пірсон та Файф у своїх працях [5]. Дослідження впливу хвиль капілярного типу на хвилі гравітаційного типу продовжилось у теоретичних [6] та експериментальних роботах [7]. Вплив поверхневого натягу на динамічну поведінку рідини у прототипі паливного баку ракетоносія досліджено у роботі [8] та досліджено такі параметри, як число Бонда, динамічне відношення капілярних сил та сил в'язкості, зміну числа Рейнольдса та критерію Вебера для поверхневого натягу, а також при незначних коливаннях сили тяжіння. Дослідженню ролі гравітації у гідродинамічній стійкості присвячено розділ у монографії [9], де також розглянуто капілярно-гравітаційні хвилі.

**Мета дослідження**

Метою дослідження є створення методики дослідження поведінки вільної поверхні при власних коливаннях рідини в циліндричних оболонках при різних рівнях гравітації.

**Виклад основного матеріалу дослідження**

У даній роботі розглядаються коливання рідких компонентів у жорсткій циліндричній оболонці висотою H та радіусом R, частково заповненій рідиною на висоту $H_1$. Змочена поверхня $S_1$ та вільна поверхня рідини $S_0$зображені на рис. 1.

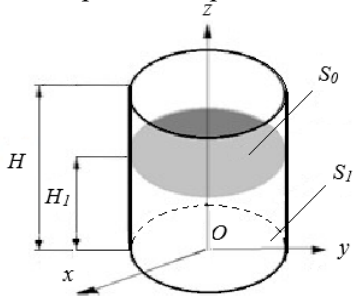



*Рис. 1 Циліндрична оболонка, заповнена рідиною*

Рідина, що заповнює резервуар є однорідною і нестисливою, знаходиться під дією гравітаційного поля, потік рідини безвихровий, а потенціал швидкості рівний $V = \nabla\varphi$. На вільній поверхні рідини $S_0$ приймаємо до уваги дію сил поверхневого натягу, а їх зв'язок з гравітаційними силами враховуємо у співвідношенні для визначення числа Бонда [9]:

$$B_0 = \left(\rho g R^2\right)/\sigma, \tag{1}$$

де $\rho$ – густина рідини;

$g$ – гравітаційна сила, що діє на рідину;

$\sigma$ – поверхневий натяг рідини;

$R$ – радіус циліндричної оболонки.

В роботі визначені частоти коливань рідини в циліндричних резервуарах за різних значень гравітаційного прискорення та поверхневого натягу:

$$\omega_k = \omega_k^g \sqrt{1+\mu_k^2 / B_0}$$

де $\omega_k^g$ - частота коливання рідини у резервуарі без врахування поверхневого натягу (відповідає лише дії гравітаційного поля);

$\mu_k$ є коренями рівняння похідної функції Бесселя $J_1(\rho)$:

$$\frac{d\,J_1(\rho)}{d\rho} = 0.$$

Результати чисельного моделювання демонструють високу точність та ефективність запропонованого методу граничних елементів та розкривають нову можливість врахування впливу поверхневого натягу на частоти вібрації рідини. Форми коливань вільної поверхні рідини мають характер, схожий з функціями Бесселя.

ЛІТЕРАТУРА

1. Еселева Е.В., Гнитько В.И., Стрельникова Е.А. Собственные колебания сосудов высокого давления при взаимодействии с жидкостью. *Пробл. машиностроения,* т. 9, №1, с.105-118, 2006.
2. Kelvin (W. Thomson), Initiation of deep-sea waves of three classes: 1) from a single displacement; 2) from a group of equal and similar displacements; 3) by a periodically varying surface pressure, Mathematical and Physical Papers, vol. 4, Cambridge, pp. 419-456, 1910.
3. Rayleigh (J.W. Strutt), *On periodic irrotationalj waves at the surface of deep water*, Philosophical Magazine, (6) 33, pp. 381–389, 1917.
4. J. R. Wilton, LXXII. On ripples, Philosophical Magazine, Series 6. 29 (173), pp. 688–700, 1915.
5. W. Pierson, P. Fife, Some nonlinear properties of long-grested periodic waves with lengths near 2,44 centimeters, Journal of Geographical Research 66**(**1),pp. 163–179, 1961.
6. Ali Hasan Nayfeh, Non linear waves in a Kelvin-Helmholtz flow, Journal of Fluid Mechanics. 55(2), pp. 311–327, 1972.
7. L. F. Mc Goldrick, *An experiment on second-order capillary gravity resonant wave interaction*, Journal of Fluid Mechanics, 40(2), pp. 251–271, 1970.
8. R.J. Hung, C.C. Lee, F.W. Leslie Similarity rules in gravity jitterrelated spacecraft liquid propellant slosh waves excitation, Journal of Fluids and Structures, pp. 493–522, 1992.
9. J. C. Legros, A. Sanfeld, M. Velarde in Fluid Sciences and Materials Science in Space: A European Perspective, Springer-Verlag, GmbH, pages 743, 1987.

**МИРОНЕНКО Марія Леонідівна** – аспірант, Інститут проблем машинобудування ім. А.М. Підгорного НАН України, м. Харків; тел.: (098) 931-85-19; e-mail: mariamyronenko87@gmail.com; ID ORCID: 0000-0002-0266-4463.

Наукові інтереси:

*– чисельні методи моделювання.*

**УДК 004.75**

МИХАЙЛОВ А.Ю., ШЕВЦОВ С.О., ЯНКО Д.Є.

# УПРАВЛІННЯ РОЗПОДІЛЕНОЮ ОБЧИСЛЮВАЛЬНОЮ ПРОЦЕСОРНОЮ СИСТЕМОЮ НА ОСНОВІ ТЕХНОЛОГІЇ HYPERLEDGER

**Вступ**

За останні кілька років відбувся істотний розвиток інструментів, призначених для розподілених обчислень. Поштовхом до розвитку слугувало активне впровадження технології BlockChain в області розподілених фінансів [1]. Ці ж технології існували і задовго раніше [2]. Але після активного впровадження в царині розподілених фінансів, істотно збільшилася кількість дослідників, які працюють в цьому напрямку, що, відповідно, призвело й до зростання кількості цікавих напрацювань. Слідом за сферою фінансів розподілені реєстри знайшли своє застосування в логістиці, ланцюгах постачання товарів, системах зберігання доказів, економіці, медицині, рітейлі, будівництві тощо. Ми ж зосередилися на застосуванні розподілених реєстрів в області розподіленого виконання математичних розрахунків та розпочали створення програмного продукту (платформи) для цього. Цей програмний продукт дозволятиме керувати (управляти) ресурсами розподіленого обчислювального середовища таким чином, щоб забезпечити усунення простоїв ресурсів та неефективності роботи проектної команди й суттєво скоротити витрати на проведення обчислень.

**Основні причини необхідності зменшення вартості обчислювальних ресурсів та зниження ступеня їх впливу**

Розподілена обробка великих масивів даних надає широкі можливості для дослідника. При цьому обмежуючим фактором є вартість необхідних обчислювальних ресурсів. Найчастіше ця вартість настільки велика, що робить проведення досліджень неможливим. Таким чином, складається структура ринку, за якої масштабні розподілені розрахунки (обчислення) доступні лише великим компаніям (організаціям). Нашою ж метою є скорочення витрат на споживані ресурси при проведенні розподілених обчислень до такого рівня, на якому розрахунки стають доступними всім бажаючим, включаючи заклади вищої освіти під час наукової, науково-дослідницької або науково-педагогічної роботи.

Розглянемо основні причини, які впливають на вартість ресурсів, необхідних для розподілених обчислень:

1. Використання технологій, при створенні яких ефективне використання ресурсів не розглядалося як мета. Багато сьогоднішніх високорівневих мов програмування й бібліотек рідко оптимізовані для високої продуктивності обробки даних. Правильний вибір технології та правильне її використання дозволяє істотно скоротити споживання ресурсів.

2. Оренда обладнання за ціною, порівнянною з вартістю придбання такого ж обладнання. Протягом тривалого часу переважаючою стала точка зору про те, що апаратне забезпечення слід брати в оренду для уникнення витрат на обслуговування власного апаратного забезпечення. При цьому вартість оренди така, що, як правило, існує альтернатива у вигляді придбання аналогічного апаратного забезпечення за вартістю, яка дорівнює вартості 1 (одного) місяця оренди. Також потрібно зауважити, що обслуговування орендованого апаратного забезпечення вимагає таких же витрат, як і обслуговування власного.

3. Повільна швидкість впровадження апаратного забезпечення, яке відрізняється від загальноприйнятого.

4. Відсутність у розробників програмного забезпечення навичок ефективного використання ресурсів.

5. Простої інженерів, пов'язані з ненадійністю програмного або апаратного забезпечення, яке бере участь в розрахунках.

Метою нашої роботи є зниження ступеня впливу причин, що підвищують вартість ресурсів, необхідних для обчислень:

1. Система сумісна з різними середовищами, застосовуваними для виконання обчислень. Проте надається методологія за вибором кращого середовища для кожного класу.

2. Мається на увазі можливість використання як орендованого обладнання, так і обладнання, що належить стороні, яка виконує обчислення.

3. Йдеться про можливість здешевлення обчислень за рахунок використання нестандартного обладнання з низького цінового сегменту [4].

4. Розроблюваний програмний продукт забезпечує управління ресурсами розподільної обчислювальної системи таким чином, щоб забезпечувалася безперервна робота обчислювального середовища без збоїв, тим самим скоротивши час простою цих ресурсів та робочої команди інженерів. Скорочення часу простою й часу неефективної роботи проектної команди є основоположним фактором, що зменшує вартість проведення обчислень.

**Архітектура**

Hyperledger - це спільний open-source проєкт, створений для просування технологій блокчейн за допомогою реалізації функцій, необхідних для відкритого міжгалузевого стандарта розподілених реєстрів. Це міжнародний проект, який охоплює провідні компанії в сфері фінансів, банківського сектора, інтернету речей, логістики, виробництва і технологій. Проєкт Hyperledger функціонує за підтримки The Linux Foundation.Практичне застосування технології блокчейн в розробці програмного продукту може суттєво скоротити витрати і час на вирішення великої кількості проблем, пов'язаних із автоматизацією процесів [10].

На даний час розроблено та успішно впроваджено такі фреймворки HyperLedger, як-от: Fabric, Iroha, Sawtooth, Indy та Burrow.

Ми пропонуємо більш детально зосередитися на фреймворці HyperLedger Fabric, оскільки наш програмний продукт використовує саме цю технологію,

Відповідно до задуму його авторів, HyperLedger Fabric є розподіленою операційною системою для управління розподіленими реєстрами ланцюжків блоків [3]. При цьому архітектура HyperLedger Fabric містить в собі велику кількість інструментів, необхідних для побудови операційної системи управління будь-якими розподіленими ресурсами. Hyperledger Fabric це open-source проєкт, який складає одну з гілок відкритого проєкту Hyperledger, консорціуму Linux Foundation. Hyperledger Fabric був започаткований Digital Assets та IBM [8]. Основною особливістю платформи Hyperledger Fabric є спрямованість на корпоративне застосування. Саме тому ця платформа розроблялася з урахуванням забезпечення високої швидкості проведення транзакцій та їх низької вартості, а також ідентифікації всіх учасників. Ці переваги досягаються за рахунок розділення служби перевірки транзакцій та формування нових блоків розподіленого реєстру, а також застосуванню центру сертифікації та авторизації учасників.

Пошарова архітектура HyperLedger Fabric дозволяє адаптувати його окремі частини для застосування у різних практичних завданнях. Компоненти certificate authority, orderer, peer залишаються без змін і налаштовуються стандартним способом [5]. Компонент chain-code розробляється і налаштовується таким чином, щоб забезпечити наступні можливості:

1. Одиницею ресурсів у розподіленій обчислювальній системі є комп'ютер, фізичний або віртуальний, під управлінням операційних систем Windows, Linux, MAC OS, Android.

2. Використовуємо chain-code (також відомий як смарт-контракт) в якості програмного забезпечення (агента) [6] для управління фізичним ресурсом.

3. Відмовляємося від запуску chain-code в Docker, оскільки у нашій системі запуск здійснюється на керованому ресурсі.

4. Розробляємо chain-code агентів для кожного виду керованих ресурсів в залежності від операційної системи. Агенту надається повний доступ до управління операційною системою ресурсу.

5. Використовуємо ledger state для незмінної фіксації будь-якого керуючого сигналу по взаємодії з агентом ресурсу.

6. Використовуємо world state для аналітичних розрахунків стану розподіленої системи.

7. Для кожного ресурсу, що знаходиться у нашому розпорядженні, реєструємо агента у мережі HyperLedger Fabric. Це забезпечує розподілений механізм управління всіма доступними ресурсами.

8. Розробляємо зовнішній механізм спостереження й управління ресурсами системи, який забезпечує безперебійну роботу обчислювального середовища.


*Рис. 1 Загальна архітектура*

**Застосування та варіанти використання**

Система управління розподіленою обчислювальною системою дозволяє встановлювати та виконувати різне програмне забезпечення, яке може використовуватися для математичних розрахунків. Безпосередній механізм обчислень є окремим агентом [6] та його опис виходить за межі нашої роботи.

Як приклад, можемо розглянути паралельні обчислення з використанням IPython [7]. Ця методологія обчислень повністю сумісна з нашим програмним продуктом, ресурси під керуванням якого можуть використовуватися для запуску обчислень IPython.

*Табл.1 Міжнародні компанії, які використовують Hyperledger Fabric [9]*

| **Company name** | **Website** | **Country** | **Top level industry** | **Sub level industry** | **Employees** |
|---|---|---|---|---|---|
| **PayPal** | paypal.com | US | Finance | General Financial Services & Insights | Above 10,000 |
| **FNZ** | fnz.com | GB | Technical | Software Development & Technical Consulting | From 1,000 to 4,999 |
| **Standard Chartered Bank** | sc.com | GB | Finance | Banking | Above 10,000 |
| **Amazon** | amazon.com | US | Business Services | All Other Professional and Technical Services | Above 10,000 |

**Висновки**

На базі програмного продукту HyperLedger Fabric розроблений проєкт платформи, що представляє собою операційну систему для управління ресурсами розподіленої обчислювальної системи.

Запропонований підхід спрямований на здешевлення розподілених обчислень, що робить такі обчислення доступними кожному без значних фінансових вкладень. Керуючий вплив в

обчислювальній системі записується у ланцюжку блоків, що дозволяє побудувати прозоре спостереження за використанням ресурсів системи.

ЛІТЕРАТУРА

1. Pavel Kravchenko, Bohdan Skriabin, Oleksandr Kurbatov. Engineer`s guide to financial Internet. Authors` test edition. Kharkiv, 2019. 224 p.
2. Ali Ghodsi. Distributed k-ary System: Algorithms for Distributed Hash Tables. A Dissertation submitted to the Royal Institute of Technology (KTH) in partial fulfillment of the requirements for the degree of Doctor of Philosophy December 2006.
3. Elli Androulaki, Artem Barger, Vita Bortnikov. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. IBM. https://arxiv.org/pdf/1801.10228.pdf.
4. Simone Atzeni, Mohammed al-Mahfoudh. How to build a Raspberry Pi cluster. http://formalverification.cs.utah.edu/pedagogy/attachments/pi_tutorial.pdf.
5. Building Your First Network. HYPERLEDGER Fabric website URL: https://hyperledger-fabric.readthedocs.io/en/release-2.0/build_network.html.
6. D. Carnì, D. Grimaldi, L. Nigro, P. F. Sciammarella, F. Cicirelli. Agent-based software architecture for distributed measurement systems and cyber-physical systems design. IEEE: website URL: https://ieeexplore.ieee.org/document/7969977.
7. Using IPython for parallel computing. IPython Interacting Computing: website URL: http://ipython.org/ipython-doc/dev/parallel/.
8. Hyperledger Fabric для Чайников. Хабр: веб-сайт. URL: https://habr.com/ru/company/ibm/blog/444874/
9. Hyperledger Fabric. HG Insights: website URL: https://discovery.hgdata.com/product/hyperledger-fabric/
10. Hyperledger. TADVISER: website URL:
11. http://www.tadviser.ru/index.php/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B0%D0%BD%D0%B8%D1%8F:Hyperledger_(Open_Ledger_Project).

**МИХАЙЛОВ Андрій Юрійович** – інженер спеціалізації «Обробка складних сигналів», дослідник, випускник ХНУРЕ, e-mail: andrew@vitche.com; ORCID: 0000-0002-6318-9467.

Наукові інтереси:

*– передача даних в комп'ютерних мережах, стискання даних, статистичний аналіз та прогнозування, математичне моделювання процесів.*

**ШЕВЦОВ Сергій Олександрович** – директор, Товариство з обмеженою відповідальністю «Бюро «ІРІС», м. Київ; член Національного технічного комітету стандартизації «Промислова автоматизація» № 185; м. Київ, а/с 48, 01054, студент заочної форми навчання Харківський національний університет імені В. Н. Каразіна, площа Свободи, 4, Харків, Україна, 61022; e-mail: shevtsov_sa@yahoo.com; ORCID: 0000-0001-5334-2109.

Наукові інтереси:

*– застосування стандартів та схем з оцінки відповідності в процесі інноваційної діяльності; діяльність національних технічних комітетів стандартизації.*

**ЯНКО Діана Євгеніївна** - студент заочної форми навчання Харківський національний університет імені В. Н. Каразіна, площа Свободи, 4, Харків-22, Україна, 61022; e-mail:yanko.diana@gmail.com ORCID*: 0000-0002-8994-546X.

Наукові інтереси:

*- Порівняльний аналіз і способи використання розподілених реєстрів на основі Hyperledger.*

**УДК 004.2**

МОРОЗ О. Ю., ТОЛСТОЛУЗЬКА О. Г.

# АНАЛІЗ ІСНУЮЧИХ ТЕХНОЛОГІЙ ВЕРИФІКАЦІЇ ПАРАЛЕЛЬНИХ ПРОГРАМ

В інформаційному суспільстві розроблення програмного забезпечення стало масовою діяльністю.

**Software Testing** розвивається як індустрія та наука. Відбуваються розподіли, напрямки, наукові течії; застосовуються відмінні техніки, методики та практики тестування програмного забезпечення. Нерідко це зумовлено різними цілями, що переслідують компанії або організації, або пов'язується із особливостями тестування різних категорій продуктів (медицина, туризм, освіта, фінанси, e-commerce та ін.)

**STLC (Software Testing life cycle)** *або Життєвий цикл тестування програмного забезпечення* – це всі дії, що виконуються під час тестування програмного продукту.

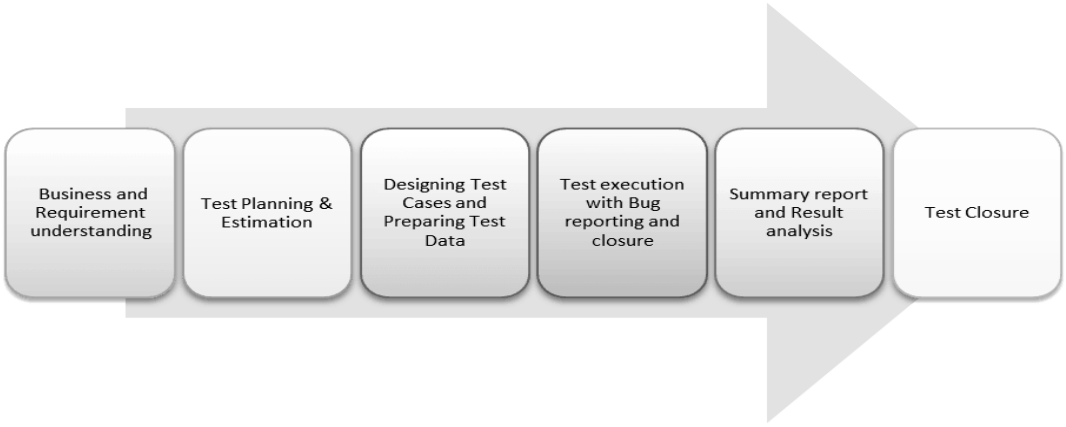



*Рис. 1 Ілюстрація STLC (Software Testing life cycle)*

На сьогодні, забезпечення високої надійності і безпомилковості програмного забезпечення сучасних систем управління є однією з головних завдань IT-індустрії. У зв'язку з цим, актуальним є впровадження в практику інженерії програмного забезпечення методів формальної верифікації, що доповнюють традиційні методи тестування і налагодження, і дозволяють підвищити безвідмовність і безпеку програм.

Верифікація програмного забезпечення – більш загальне поняття, ніж тестування. Метою верифікації є досягнення гарантії того, що верифікований об'єкт (вимоги або програмний код) відповідає вимогам, реалізований без непередбачених функцій і задовольняє проектним специфікаціям і стандартам. Загальноприйнятий поділ методів верифікації представлено у вигляді діаграми на рисунку 2.

**Експертизою ПЗ** називають всі методи верифікації, в яких оцінка артефактів життєвого циклу ПЗ виконується людьми. Перевагою даного методу є те, що при його використанні виявляються в середньому 50-90% помилок [5]. Але цей метод має і недоліки. Пошук помилок, оцінка і аналіз властивостей ПЗ людиною (зазвичай це група 2-5 осіб). Потрібні справжні експерти, програмісти з досвідом роботи не менше 10 років.

**Статичний аналіз –** аналіз без виконання програми. Методи статистичного аналізу можна розділити на два види: контроль того, що всі формалізовані правила коректності побудови цих артефактів виконані, та пошук типових помилок і дефектів в них на основі деяких шаблонів [2]. Часто інструменти статичного аналізу використовують обидва типи перевірок. Статичний аналіз можна вважати найбільш широко застосовуваним методом

верифікації. Перевірені на практиці правила коректності коду або шаблони типових помилок переносяться в середовища розробки.

Переваги статичного аналізу:

- Автоматичний аналіз багатьох шляхів виконання одночасно.
- Виявлення помилок, що проявляються лише на одиничних шляхах виконання або на незвичайних вхідних даних.
- Можливість аналізу на неповному наборі вхідних файлів.
- Відсутність накладних витрат під час виконання програми.

Недоліки даного методу:

- Велика кількість помилкових спрацювань.
- Необхідна ручна перевірка результатів роботи, що вимагає значних часових, людських та матеріальних ресурсів.

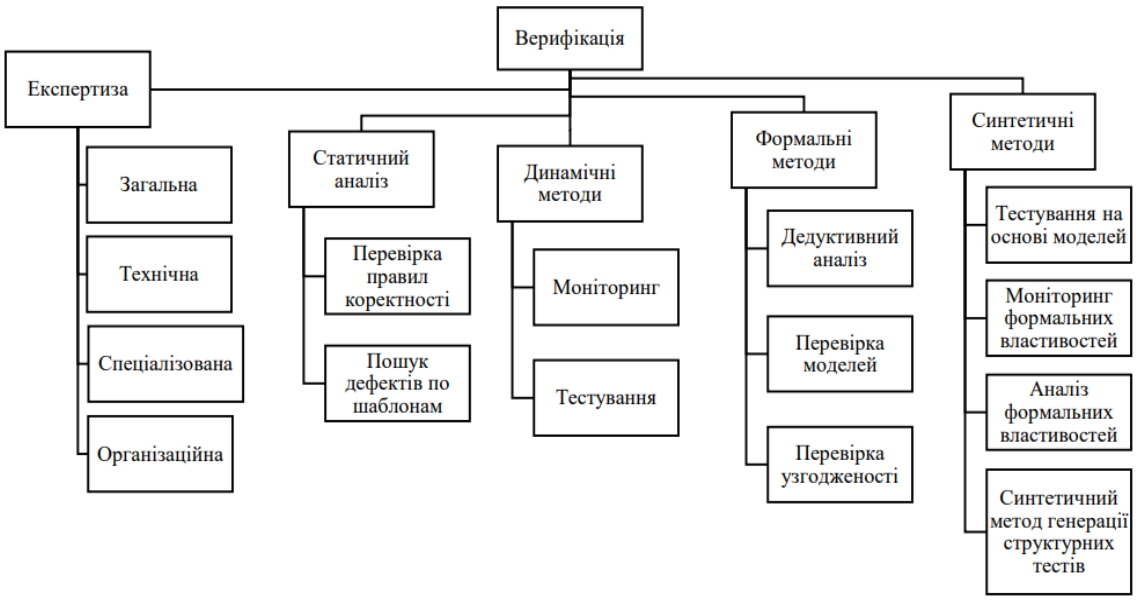



*Рис.2 – Загальноприйнятий поділ методів верифікації*

**Динамічні методи верифікації** використовують результати реальної роботи програмної системи або її прототипів, щоб перевіряти відповідність цих результатів вимогам і проектним рішенням.

Існує два основних види динамічних методів верифікації: моніторинг, в рамках якого йде тільки спостереження, запис і оцінка результатів роботи ПЗ при його звичайному використанні, і тестування, при якому ПЗ виконується в рамках заздалегідь підготовлених сценаріїв. Перевагою даного методу є висока точність виявлення помилок. А недоліками – необхідно мати набір вхідних даних та середовище виконання, а також високі вимоги до ресурсів.

**Формальні методи верифікації.** Їх відмітною особливістю є можливість проведення пошуку помилок на математичній моделі, без звернення до фізичної реалізації, що в деяких випадках досить зручно і економічно. Для проведення аналізу формальних моделей застосовуються специфічні техніки, такі як дедуктивний аналіз, перевірка моделей, перевірка узгодженості. На жаль, для побудови таких моделей завжди необхідно виходити так само з коректності та адекватності моделі ПЗ [4]. Лише після правильної побудови цієї моделі можна автоматично проаналізувати деякі з її властивостей. Проте, в більшості випадків для ефективного аналізу від фахівців будуть потрібні глибокі знання математичної логіки і алгебри і деякого набору навичок роботи з цим апаратом.

В останні роки активно розробляються інструменти автоматичної генерації тестів на основі коду, які використовують додаткові джерела інформації. В якості таких джерел виступають статичний аналіз коду, формальний аналіз, моніторинг виконання раніше побудованих тестів і т.п. Оскільки в інструментах цього типу використовується зазвичай 3-4 техніки різних типів, методи, що лежать в їх основі, винесені в окремий різновид синтетичних методів верифікації [3]. Синтетичні методи верифікації поєднують підходи декількох типів – статичний аналіз, формальний аналіз властивостей ПЗ, тестування. Деякі з таких методів породили в останні 10–15 років самостійні галузі досліджень, в першу чергу, тестування на основі моделей і моніторинг формальних властивостей. Переваги та недоліки синтетичних методів визначаються комбінацією методів верифікації, які входять до її складу.

Методи формальної верифікації програмного забезпечення комп'ютерних систем дозволяють гарантувати перевірку виконання моделлю системи верифікованих властивостей. В даний час ці методи активно розвиваються в напрямку зниження загальної вартості формальної перевірки, підтримки сучасних концепцій програмування і мінімізації «ручного» праці при переході від моделі системи до її реалізації. Кожен метод верифікації використовується в конкретному класі випадків в залежності від поставленої мети. Найактуальнішими, найбільш корисними та продуктивними можна вважати синтетичні методи верифікації ПЗ, оскільки вони так чи інакше намагаються поєднати переваги різних підходів до верифікації, зменшуючи їх недоліки. В даний час досягнуті значні успіхи в розробці таких методів і впровадженні їх у практику промислової розробки ПЗ.

ЛИТЕРАТУРА

1. IEEE 1012-2004 Standard for Software Verification and Validation. IEEE, 2005. – p.153.
2. L. Yu A light-weight static approach to analyzing UML behavioral properties / L. Yu, R. B. France, I. Ray,K. Lano.. Proc. of 12-th IEEE International Conference on Engineering Complex Computer Systems (ICECCS 2007),pp. 56-63, 2007. – p. 79.
3. M. Broy Model Based Testing of Reactive Systems / M. Broy, B. Jonsson, J.-P. Katoen, M. Leucker, A.Pretschner (eds.). LNCS 3472, Springer, 2005. – p. 273.
4. T. Ball Thorough Static Analysis of Device Drivers. In Proc. of EuroSys 2006/ T. Ball, E. Bounimova, B.Cook, V. Levin, J. Lichtenberg, C. McGarvey, B. Ondrusek, S. K. Rajamani, A. Ustuner., ACM SIGOPS OperatingSystems Review, 2006. – p. 74.
5. Y. K. Wong. Modern Software Review: Techniques and Technologies. IRM Press, 2006. – p. 368
6. Б.У. Боэм. Инженерное проектирование программного обеспечения. М.: Радо и связь, 1985. – 368 с.

**МОРОЗ Ольга Юріївна –** старший викладач кафедри теоретичної та прикладної системотехніки факультету комп'ютерних наук; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 6, Харків, Україна, 61022; e-mail: o.moroz@karazin.ua; ORCID: 0000-0002-4920-4093.

Наукові інтереси:

*– Технології автоматичного проектування паралельних програм.*

**ТОЛСТОЛУЖСЬКА Олена Геннадіївна –** д.т.н., с.н.с., професор кафедри теоретичної та прикладної системотехніки факультету комп'ютерних наук; Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 6, Харків, Україна, 61022; e-mail: elenatolstoluzka@gmail.com; ORCID: 0000-0001-2741- 180.

Научные интересы:

*– Технології автоматичного проектування паралельних програм.*

**УДК 004.51**

НАДОЛЬКО В.Ю

# МОЖЛИВОСТІ ЗАСТОСУВАННЯ ПОСТУПОВИХ ВЕБ-ЗАСТОСУНКІВ (PROGRESSIVE WEB APPLICATION) ДЛЯ РОЗРОБКИ ВЕБ-ДОДАТКІВ

**Вступ**

Сучасний світ стоїть на декількох стовпах. Одним з таких стовпів є Інтернет. Важко уявити нашу реальність без нього. З плином часу відбувався невпинний розвиток технологій, кількість регулярних користувачів Інтернету збільшувалась, та, з появою смартфонів – почав зростати веб-ринок. Для багатьох сучасних користувачів – використання мобільного інтернету залишається єдиним можливим способом отримування доступу до веб-ресурсів. Маючи на увазі таку тенденцію, власники великих інтернет медіа ресурсів отримали потреби в розробці мобільних та планшетних версій своїх порталів. Але, не кожен розробник веб-застосунків має на увазі такі фактори, як час завантаження сторінок його порталу, чи обсяг пам'яті, що займає нативний застосунок його сервісу. Веб-індустрія вже почала відчувати, що скоро можуть наближатися зміни. Поява нового прориву на веб-ринку була лише питанням часу. Таким проривом стала розробка – поступових веб-застосунків.

**Мета роботи**

Метою є дослідження і опис технології Progressive Web Application, її переваги та недоліки, вплив на сучасну веб-індустрію.

**Аналіз**

Що взагалі таке "поступовий веб-застосунок"? Поступові веб-застосунки – це веб-застосунки, які мають додаткові можливості, наприклад, такі як – робота офлайн, push сповіщення, та доступ до обладнання телефону [4]. Традиційно – такі можливості зазвичай були притаманні нативним мобільним додаткам, але зараз, рамки було роздвинуто, і поступові веб-застосунки ввібрали в себе найкраще з веб та мобільних застосунків. Саму ідею поступових веб-застосунків було запропоновано командою розробників з Google у 2015 році, оскільки вони шукали рішення у подоланні розриву між веб та нативними додатками. З моменту появи такої технології – її ціллю було уникнення встановлення застосунку через комерційні магазини типу App Store від Apple та Play Market від Google. Кожен може запитати себе як часто він встановлює додатки через магазини. Статистично підходячи до питання – за 30 денний період приблизно 50% користувачів взагалі не інсталюють додатки. Це може відбуватися з багатьох причин, по-перше – можливо, потрібно багато часу, або пам'ять телефона користувача може бути дуже обмежена, тому інсталювання – неможливе, по-друге, можливо, психологічно, користувач не хоче мати великої відданості ресурсу, який пропонує йому встановлення додатку. І тут на порятунок приходять поступові веб-застосунки – квінтесенція простоти у застосуванні і швидкості та найкращого досвіду користувача.

**Огляд**

Основними елементами для створення поступового веб-застосунку є маніфест та service worker [1]. Маніфест – це визначений за специфікацією маніфест у форматі JSON для зберігання метадати застосунку, такої як назва застосунку, посилання на іконку, посилання на головну сторінку при відкритті, та інше. Без цього маніфесту просто неможливо створити застосунок на домашньому екрані користувача. Service worker – це скрипти, написані мовою JavaScript, які працюють незалежно від головного потоку браузера, для синхронізації даних у фоновому режимі, обробки push-сповіщень, отримання запитів на ресурси, кешування. Вони забезпечують більшість переваг цієї технології, насамперед – високу продуктивність і покращений досвід користувача.

**Переваги**

Першою перевагою можна вважати легкість процесу інсталяції [2]. При користуванні веб-застосунком, який підтримує цю технологічно парадигму – вам буде запропоновано локально інсталювати поступовий застосунок на вашому смартфоні. Користувачеві навіть не потрібно перейматися про оновлення, в них завжди буде встановлено найновішу версію. Другою перевагою, яку потрібно відмітити – збільшення конверсії, оскільки застосунок знаходиться на домашньому екрані смартфону користувача, що буде спонукати його, для частішого використання, та push-сповіщення, за допомогою яких користувач буде більш зацікавлений у застосунку. Наступною перевагою є зменшення витрат на розробку та підтримку у порівнянні з звичайними веб-застосунками. Це зумовлено тим, що потрібно мати на увазі різні тонкощі підчас розробки, оскільки один і той же поступовий веб-застосунок працює однаково, як на Android девайсі, так і на iOS. Далі, потрібно зазначити швидкість, легкість та безпечність. За статистикою Google, 53 відсотки потенційних користувачів залишать ваш веб-застосунок, якщо він буде завантажуватися довше, ніж 2 секунди. Завдяки інструментам, якими користується PWA – можливо зберегти потенціальних користувачів, за рахунок швидкого завантаження. Поступовий застосунок також важить менше, ніж нативні додатки, більш ефективні та використовують менше пам’яті сховища смартфону, але у той же час – пропонують такий же досвід користувача. Також, обмін інформацію йде за допомогою HTTPS, що означає додаткові рівні захисту, та заборону на будь яке неавторизоване втручання. Однією з переваг для розробників є те, що поступові веб-застосунки не залежать від крамниць додатків, тому, розробникам не потрібно очікувати певний час модерації. Це спрощує алгоритм дистрибуції, через те, що вбирає одну ланку за ланцюга процесу. Останнім, важливо зазначити, що PWA можуть працювати офлайн, завдяки кешуванню. За допомогою своїх технології, поступові веб-застосунки завжди можуть звернутися до кешованих даних, не зважаючи на наявність інтернет з’єднання. Якщо користувач захоче відкрити якусь нову сторінку, яку ще не було прокешовано, то застосунок не закриється, а покаже юзеру спеціальне повідомлення. Ця перевага надає застосунку більш надійного досвіду користувача.

**Недоліки**

Як і будь яка інша технологія, progressive web applications мають певні незначні недоліки. Незважаючи на те, що поступові веб-застосунки встановлено локально, вони все ще веб-застосунки, по своїй сутності, тому мають певну обмежену функціональність у порівнянні з нативними додатками. Влучними прикладами можуть слугувати такі моменти, як те, що вони не мають повномірного доступу до камери чи до сканера відбитку пальця. Ще, нативні застосунки можуть пропонувати більш персоналізовані push-сповіщення, ніж PWA. Також, такі застосунки не рекламуються магазинами додатків, що призводить до не зростання аудиторії сторонніх користувачів, залучених через рекламу та чарти додатків.

**Приклад**

Можна скільки завгодно розхваляти технологію, але наглядніше буде привести приклад успішної імплементації. Таким прикладом – слугує Twitter Lite [3]. Місячний обсяг користувачів Твіттеру – 328 мільйонів, серед них – приблизно 80 % користувачі мобільних телефонів. З метою покращення досвіду користування їх платформою – було розроблено Twitter Lite, який поєднує в собі найважливіші характеристики – миттєве завантаження, зниження споживання даних та залучення користувачів. Як результат :

- Збільшення кількості надісланих твітів на 75%.
- На 65% збільшилася кількість відвіданих сторінок.
- 20% - на стільки зменшився показник відмов за допомогою push-сповіщень
- 250 000 унікальних щоденних користувачів, що запустили Twitter Lite з домашнього екрану, щонайменше – 4 рази за день.
- Вага нативного додатку – приблизно 24 мегабайти, а Twitter Lite – приблизно 600 кілобайт.

Згідно зі слів головного інженера Twitter Lite Ніколаса Галлагера Twitter Lite став найшвидшим, найнадійнішим і найергономічнішим способом використання сервіса. Є повністю конкурентоспроможним з нативним додатками, при тому, що у порівнянні має лише 3% його ваги.

**Висновки**

У цьому дослідженні було розглянуто технологію Progressive Web Application. Проаналізовано її переваги та недоліки, відмічено причини стосовно нагальності її використання, приведено приклади. Технологія, що з'явилася лише декілька років тому – вже суттєво впливає на сервіси, де була впроваджена, при тому, що це тільки перші кроки. За допомогою імплементації progressive web application до свого сервісу – користувачі отримують повний букет переваг для мобільного використання, а власники – збільшення обсягу й статистики свого веб-застосунку. Цей підхід – є дуже влучним рішенням для ситуацій, коли будь який веб-портал стикається з такими проблемами, як тривалий час завантаження або низька конверсія. Майбутні покращення – це лише питання часу, вони обов'язково будуть впроваджуватися і супроводжуватися підтримкою зі сторони гігантів-власників мобільних сервісів, для покращення свого бізнесу.

ЛІТЕРАТУРА

1. Progressive Web Application Official Documentation. [Електронний ресурс]. URL: https://codelabs.developers.google.com/codelabs/your-first-pwapp/#0 (Last accessed: 13.03.2020).
2. Progressive Web Applications, обзор: [Електронний ресурс]. URL: https://habr.com/ru/post/418923/ (Last accessed: 12.03.2020).
3. Google Twitter Lite Showcase. [Електронний ресурс]. URL: https://developers.google.com/web/showcase/2017/twitter (Last accessed: 14.03.2020).
4. Progressive Web Application Wikipedia Overview. [Електронний ресурс]. URL: https://en.wikipedia.org/wiki/Progressive_web_application (Last accessed: 14.03.2020).

**НАДОЛЬКО Владислав Юрійович** – студент кафедри штучного інтелекту та програмного забезпечення; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: vladislavnadolko@gmail.com; ORCID: 0000-0002-7087-1878.

Наукові інтереси:

*– розробка програмного забезпечення.*

**УДК 681.5.004.0**

НЕБЕСНЮК С.А., БЕРДНИКОВ А.Г.

# МОДЕЛЬ УПРАВЛЕНИЯ АСУ ТП НА ОСНОВЕ МЕССЕНДЖЕРА "Telegram"

**Введение**

В настоящее время, несмотря на развитие сложных систем управления в масштабных производственных процессах, не менее важной задачей является внедрение автоматизированных систем управления технологическими процессами (АСУ ТП) в простых производствах, требующих оперативного контроля и вмешательства оператора. К таким процессам относится технология «умного дома», которую можно использовать, например, для управления домашними приборами, несложными технологическими процессами агропромышленного комплекса (например, установка режимов экономного полива в тепличном хозяйстве и т.п.).

Вследствие большого разнообразия сравнительно несложных технологических объектов, но требующих соответствующей автоматизации, общепринятых стандартов, а также единой концепции «разумного дома» и инфраструктуры для ее поддержки пока не существует.

Для обеспечения управления такими процессами в работе, после сравнительного анализа популярных мессенджеров (табл. 1), предлагается использовать мультиплатформенный месседжер Telegram, который сможет обеспечить удаленное управление объектами как с персонального компьютера, так и с телефонного приложения. Таким образом, разработка модели системы управления на базе месседжера Telegram представляется актуальной задачей.

**Постановка задачи**

Популярный мультиплатформенный месседжер Telegram имеет удобный интерфейс программирования приложений (API – Application Programming Interface),

позволяет организовать между компонентами системы асинхронную связь, которую можно достаточно просто масштабировать и видоизменять, обеспечивать сравнительно высокую защищенности информационного обмена.

При этом система управления на базе месседжера Telegram должна оставаться безопасной, удобной в использовании, эффективной и экономически выгодной в сравнении с отечественными и зарубежными концепциями.

Разрабатываемая система управления должна соответствовать следующим основным требованиям:

-обеспечивать удобное управление объектом;

-иметь высокий уровень защищенности от неправильных действий оператора, несанкционированного доступа и вирусных атак (они не должны приводить к аварийной ситуации);

-обеспечивать мониторинг работы без привязки к месту расположения сервера.

**Суть модели**:

В данной работе была построена аналитическая АСУ ТП. Система подает запрос, получает данные с сервера и выводит статистические данные, на основании которых оператор совершает определенные действия.Например: перезапуск сервера, увеличение интенсивности вентиляторов и т.д. Структура системы управления, построенной на основе мессенджера Telegram приведена на рис.1 где:

1, 2 передача инструкций в зашифрованном виде.

3, 4 обратная связь с сервера в зашифрованном виде.

*табл.1*

| Аспекты | Особенность | Telegram | Viber | WhatsApp |
|---|---|---|---|---|
| Нефункциональные Характеристики | Скорость Отправки Сообщений | ≈ 150 миллисекунд | ≈ 900 миллисекунд | ≈ 550 миллисекунд |
| | Объём Трафика для Отправки Сообщения | Минимальный | Средний | Средний |
| Открытость | Политика Конфиденциальности | Чёткая и прозрачная | Средней сложности | Достаточно сложная |
| | Документация | Открытая и подробная | Открытая и подробная (разработка новых приложений невозможна) | Практически отсутствует |
| | Протокол Шифрования | Собственный открытый MTProto | Полузакрытый проприетарный протокол (доступна некоторая документация) | Открытый Signal (закрытый код приложения не позволяет проверить реализацию) |
| Синхронизация | Количество Устройств | Неограниченное | Одно первичное и несколько вторичных устройств | 2 |
| | Вид | Облачная и мгновенная | 1 первичное устройство (смартфон/планшет) и несколько вторичных (планшеты/компьютеры), которые синхронизируются с первичным устройством | 1 смартфон и 1 связанный с ним компьютер |
| | Содержание для Синхронизации | Все данные чатов, настройки | Данные чатов (единоразово), сообществ | Данные чатов (единоразово) |
| | Вид | Облачное и автоматизированное | Ручное, по расписанию через Google Drive/Apple iCloud, только с мобильных устройств | Ручное, по расписанию локально и через Google Drive/Apple iCloud |
| | Содержание Резервных Копий | Все данные, кроме секретных чатов, включая: чаты, сообщения, настройки, медиа, файлы | Текстовые сообщения (кроме секретных чатов) | Текстовые сообщения и медиа |
| | | В режиме клиент-сервер с распределением ключа шифрования | | |

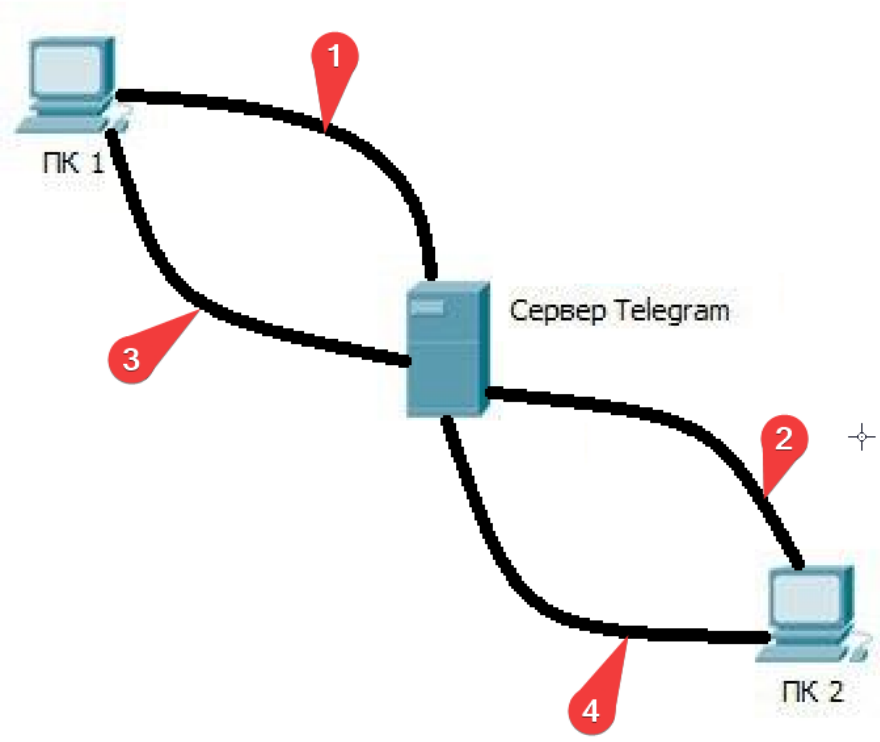



*рис.1*

Работа системы осуществляется непосредственно через программу-сервер приложения, которая обеспечивает высокое качество соединения и его стабильность.

Безопасность сообщения обеспечивается принципом E2E (end-to-end), который оценен всемирной организацией EFF (Electronic Frontier Foundation) наивысшим баллом. Суть построения системы шифрации сообщений состоит в том, что шифрование данных проходит на устройстве отправителя, а дешифровка – только на устройстве получателя. Конкуренты же в лице таких Viber и WhatsApp используют метод шифрование в котором шифрование производится на серверах мессенджеров после чего они переводятся к устройству получателя для декодирования.

Пакет Telegram имеет свой уникальный ключ доступа как к клиенту персонала, так и к самой системе с аутентификации по мобильному телефону, что позволяет защитить систему от несанкционированного доступа на высоком уровне за счет тройной аутентификации.

Преимуществом АСУ ТП на основе месседжера Telegram является то, что система обрабатывает только те команды, которые занесены в код приложения. Таким образом, пользователь в случае неправильных действий имеет возможность нанести вред системе только в том случае, если лично занесет такую функцию в программу приложения.

Вероятность поступления вирусов в программную систему достаточно низкая за счет применения защиты от воспроизведения несанкционированного кода.

**Выводы**

1) Построение системы управления на основе месседжера Telegram имеет достаточно высокий потенциал для дальнейшего развития в плане горизонтального и вертикального масштабирования, т.к. обеспечивает успешное решение задач имплементации новых узлов в системе.

2) Анализ работы модели доказали скорость работы и ее защищенность, она может быть рекомендована для проектирования автоматизированных систем управления технологическим процессом.

ЛИТЕРАТУРА

1. Денисенко В.В., научное издание. «Компьютерное управление технологическим процессом, экспериментом, оборудованием». М.: изд. «Горячая линия-Телеком», 2009, - 608 с.
2. https://automation-system.ru/spravochnik-inzhenera/34-glava7/305-7-5.html - Требование к системе в целом.
3. http://docs.cntd.ru/document/1200008639 - ГОСТ 24.104-85
4. https://ru.telegram-store.com/blog/shifrovanie-Telegramm/ - уровень надежности шофрования мессенджера telegram
5. https://ru.wikipedia.org/wiki/Автоматизированная_система_управления_технологическим_процессом

**НЕБЕСНЮК Степан Артемович** – студент группы КУ-41 факультету компьютерных наук; Харьковский национальный университет имени В.Н. Каразина, майдан Свободы, 4, Харків-22, Украина, 61022; e-mail: QuasarRasckl@gmail.com; ORCID: 0000-0001-8078-2396.

Научные интересы:

- *Программирование*
- *Управление проектами*
- *Анализ данных*

**БЕРДНИКОВ Анатолий Георгиевич** – к. т. н., доцент, доцент кафедры теоретической и прикладной системотехники; факультету компьютерных наук; Харьковский национальный университет имени В.Н. Каразина, майдан Свободы, 4, Харків-22, Украина, 61022; e-mail: tps@karazin.ua.

Научные интересы:

- *Программирование.*
- *Моделирование.*
- *Проектирование.*

**УДК 004.02**

НЄВЄЖИНА В.Ю., АРТЮХ О.А.

# МОДЕЛЬ ПРОСУВАННЯ ІНТЕРНЕТ-ПРОДУКТУ

Вступ

Об'єкт дослідження – процес просування інтернет-продукту в мережі.

Предмет дослідження – методи і моделі побудови системи просування інтернет - продукту з метою більш ефективного вирішення бізнес-завдань.

Актуальність

Інтернет в наш час є невід'ємною частиною життя всіх людей на планеті. З кожним роком з'являються все нові можливості, нові гаджети, нові технології та інструменти його використання, в тому числі і для бізнесу. З розширенням застосування Інтернету та його функцій, створюються та активно застосовуються різні методи просування товарів і послуг в мережі.

Просування – це комплекс спеціальних методів, що дозволяють власникам відповідних web-ресурсів просувати свій продукт в Інтернеті та, розкручуючи та, застосовуючи мережеві технології, отримувати додатковий прибуток, або іншу вигоду від своєї діяльності.

Моделювання просування інтернет-продукту дозволяє оцінювати різні варіанти розробки та використання продукту для досягнення максимального ефекту відносно сформульованих цілей. Дозволяє задати шкалу виміру ефективності інтернет-продукту, а також цільове значення на цій шкалі – це дає можливість розвивати інтернет-продукт тільки в напрямі, що максимально наближає його до мети. Наприклад, якщо є вибір з декількох можливих дій з інтернет-продуктом, то модель дозволить визначити пріоритетність цих дій з мірою вкладу в досягнення цілей інтернет-продукту. І це можна зробити мінімізувавши матеріальні витрати.

Мета та задачі

Метою даної роботи є аналіз теоретичних підходів до просування продукту, визначення та обгрунтування вимог до створюваної моделі, створення моделі та оцінка отриманих результатів.

Основні завдання, виконані під час роботи:

1. Аналіз інтернет-продукту, вивчення його сутт'євих характеристик для моделювання.
2. Визначення вимог до моделі. Створення структури моделі просування інтернеті-продукту.
3. Опис функціональних можливостей моделі.
4. Оцінювання працездатності та ефективності моделі.

Аналіз наявних моделей

Щоб забезпечити потрапляння в "топ" існують такі моделі просування:

Пошукова оптимізація – комплекс заходів щодо зовнішньої та внутрішньої оптимізації, для підняття позицій інтернет продукту, в результатах видачі пошукових систем по певних запитах користувачів, з метою збільшення трафіку і потенційних клієнтів і подальшої монетизації цього трафіку.

Внутрішня оптимізація – різні поліпшення і роботи проводяться спеціальною командою на самому сайті, в його внутрішній структурі. Одним з таких дій можуть бути поліпшення адрес, прописування заголовків, описів, загальна оптимізація сторінок за допомогою прописування в текстах ключових фраз, які визначаються аналізом запитів в пошукових системах того чи іншого слова.

Зовнішня оптимізація – сюди можуть входити посилання на себе на інших сайтах, реакції користувачів на ці посилання, оцінки в соціальних мережах, згадки на сторонніх ресурсах. На деякі з цих параметрів також може впливати SEO команда.

Контекстна реклама – ефективна модель просування. Розміщується даний вид реклами на сторінках пошукових систем і сайтах. Вона є адаптивної, націленої на зацікавлених користувачів. Відбувається це за допомогою аналізу пошукових запитів людей, і виведення на цій основі в рекламі схожих товарів або послуг.

Таргетована реклама – процес залучення трафіку або уваги до бренду чи продукту через соціальні платформи. Це комплекс заходів щодо використання соціальних медіа в якості
каналів для просування компаній і вирішення інших бізнес завдань.

Просування в соціальних мережах на даний момент це новий підхід, який набирає популярність. При цьому дуже ефективний: користувачі збираються за інтересами, можливі дискусії і відстеження думок клієнтів, на які можна негайно відреагувати та підлаштуватися під споживача.

Структурна схема моделі просування інтернет-продукту представлена на рисунку 1.

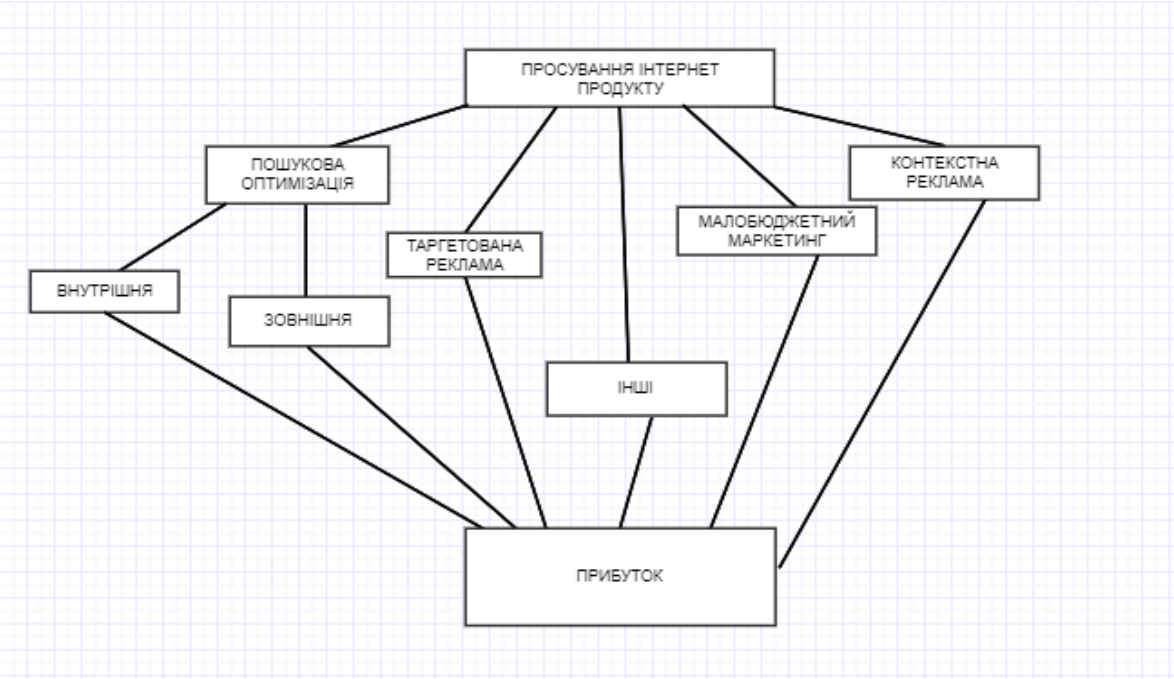



*Рисунок 1. Структурна схема моделі просування інтернет-продукту.*

Модель просування інтернет-продукту складається з наступних методів:
1.пошукова оптимізація;
2.таргетована реклама;
3.малобюджетний маркетинг;
4.контекстна реклама;
5.інші методи, які менш ефективні.

Не можна спрямувати кошти та зусилля тільки на одину з розглянутих моделей, жорстка конкуренція вимагає застосування нових методів і підходів, пошукові системи вдосконалюють свої способи перевірки інтернет продуктів на якісність. Для того, щоб стати лідером необхідно комплексно підходити до питання, планувати та розподіляти кошти.

Висновки

Модель просування інтернет-продукту потрібна передусім власникові інтернет-продукту, оскільки вона дозволяє зробити інтернет-продукт ефективнішим з точки зору рішення бізнес-завдань, а сам розвиток інтернет-продукту прозорішим і осмисленим. З іншого боку модель потрібна менеджерові продукту і усім керівникам підрозділів, що беруть участь в роботі з цим ресурсом, оскільки дозволяє їм пропонувати до реалізації найбільш ефективні ідеї та шляхи вдосконалення, а також оцінювати результат роботи.

ЛІТЕРАТУРА

1. Сумских И.А. Инновационные методы продвижения товара. //Территория науки, 2012. №3.
2. Фаустова К.И. Значение SEO для эффективных продаж в интернете. // Территория науки. 2015. №3.
3. Підприємництво. Менеджмент. [Электронный ресурс]. – Режим доступа: https://cyberleninka.ru/article/n/internet-marketing-kak-instrument-dlya-prodvizheniya-sayta/viewer.
4. Соверемнные методы продвижения сайта в интернете. [Электронный ресурс]. – Режим доступа: https://cyberleninka.ru/article/n/sovremennye-metody-prodvizheniya-sayta-v-internete/viewer.

**АРТЮХ Олексій Анатолійович** – старший викладач кафедри теоретичної та прикладної систематехніки; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail:oleksiy.artiuh@karazin.ua ORCID: 0000-0003-3916-8778.

Наукові інтереси:

*– моделювання інформаційних процесів у складних і розподілених системах.*

**НЄВЄЖИНА Вероніка Юріївна** – студентка кафедри теоретичної та прикладної систематехніки ; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: nevezhina511@gmail.com; ORCID: 0000-0003-3915-8959.

Наукові інтереси:

*– моделювання інформаційних процесів у складних і розподілених системах.*

**УДК 004.7**

НОВИКОВ В.Э., МОРОЗ О.Ю.

# РАЗРАБОТКА КОМПЬЮТЕРНОЙ МОДЕЛИ WEB-САЙТА ПРИ РАБОТЕ С БАЗОЙ ДАННЫХ СКЛАДА

**Введение**

Web-сайт – одна или несколько логически связанных между собой веб страниц; также место расположения контента сервера. Обычно сайт в Интернете представляет собой массив связанных данных, имеющий уникальный адрес и воспринимаемый пользователями как единое целое. Само веб-приложение может выступать в качестве клиента других служб, например, базы данных или другого веб-приложения, расположенного на другом сервере. Базы данных используются с целью хранения различной информации и, упрощенно, представляют собой некоторый набор взаимосвязанных таблиц. Размеры таблиц в БД различны, а их количество произвольно. Именно в базах данных хранится на сервере требуемая для работы сайта информация, например, каталог товаров или статистические данных.

**Постановка задачи**

Посредством внедрения web-сайта для внутреннего продукта планируется устранить существующие недостатки в компании. Основная цель рассматривается в разрезе двух направлений: 1)сокращение времени обработки заказа и повышение достоверности информации (достижение прямого эффекта); 2) увеличение числа обслуживаемых клиентов, увеличение продаж, повышение имиджа предприятия в целом (получения косвенного эффекта)

Наиболее подходящим видом сайта для решения поставленных задач является web-сайт, основываясь на уже существующей базе данных склада.

Целью данной работы является разработка компьютерной модели Web-сайта для обеспечения эффективной работы между складскими помещениями внутри компании с несовместимой между собой функциональностью. На этот проект выделяется определенные человеческие, временные и финансовые ресурсы, а также предъявляются определенные требования к качеству. Для создания оптимального Web-сайта необходимо количественно и качественно оценить возможные варианты использованных технологий.

**База данных** представлена в объективной форме совокупностью самостоятельных материалов (статей, расчётов, нормативных актов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

Программирование сайтов, взаимодействующих различным образом с базами данных, включает несколько основных этапов работы с БД: построение запросов к БД с помощью языка SQL, программирование сценариев для обработки этих запросов и программирование модулей для отображения результатов обработки запросов.

MySQL является популярной системой управления базами данных с открытым кодом, которая обычно используется в веб-приложениях благодаря своей скорости, гибкости и надежности. MySQL использует SQL (язык структурированных запросов) для доступа к данным в базе данных и их обработки. К главным свойствам MySQL относится :высокая скорость, надежность и универсальность в работе. Обычно сервера, работающие на PHP, автоматично включают в поддержку этой СУБД.

**Суть компьютерной модели Web-сайта**

Для создание компьютерной модели Web-сайта лучше всего подойдёт динамическая структура сайта (рис. 1). Динамический веб-сайт - это когда часть содержимого ответа генерируется динамически только при необходимости. На динамическом веб-сайте HTML-страницы обычно создаются путем вставки данных из базы данных в заполнители в HTML-шаблонах (это гораздо более эффективный способ хранения большого количества контента, чем использование

статических сайтов). Динамический сайт может возвращать разные данные для URL-адреса на основе информации, предоставленной пользователем или сохраненными настройками, и может выполнять другие операции.

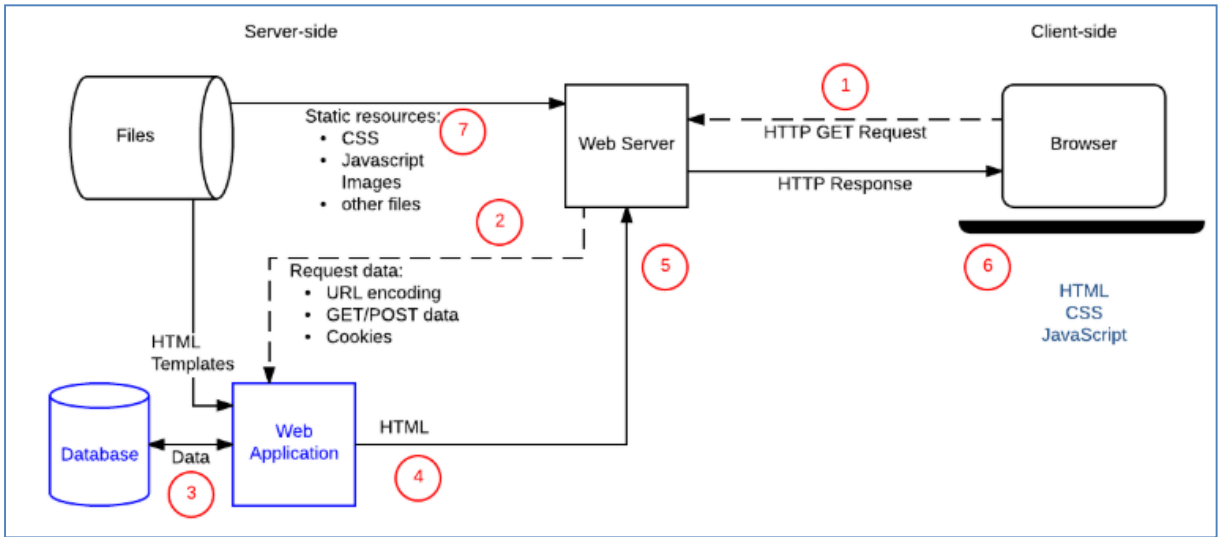



*Рис. 1 Архитектура динамического сайта*

В работе было проведено исследование для определения конкретных элементов, которые используют в эффективном дизайне веб-сайтов [4]. Элементами дизайна, наиболее часто встречающихся в литературе, были навигация, полезность контента, графическое представление, цель, простота, организация и удобочитаемость. Предложенный интерфейс приятен визуально, интуитивно понятен и удобен для пользователя, что подтверждается данными, полученными в процессе тестирования. Расположение и размеры блоков, активных элементов и меню, разработаны с соблюдением требований к web-продуктам. Цветовая гамма web-сайта, яркость и контрастность, размеры изображений отвечают стандартам.

ЛІТЕРАТУРА

1. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика [Текст] / Т. Коннолли, К. Бегг ; [пер. с англ. Г. Баркулов] – М.: Вильямс, 2015. – 546 с
2. Хабибуллин И. Разработка Web-служб средствами Java. – СПб.: БХВ-Петербург, 2003.
3. Обзор современных Web - технологий – [Электронный ресурс] - http://www.sciteclibrary.ru/rus/catalog/pages/6643.html.
4. W3schools.com. Browser Statistics and Trends. Retrieved 1/15, 2015, [Электронный ресурс] - http://www.w3schools.com/browsers/browsers_stats.asp.

**НОВИКОВ Владимир Евгеньевич** – студент группы КУ-41 факультета компьютерных наук; Харьковский национальный университет имени В. Н. Каразина, площадь Свободы 6, Харьков, Украина, 61022; e-mail: vladimirnovikov105@gmail.com ORCID: 0000-0002-7628-7808.

Научные интересы: *Программирование. Управление проектами. Алгоритмы и структуры данных.*

**МОРОЗ Ольга Юрьевна** – старший преподаватель кафедры теоретической и прикладной системотехники факультета компьютерных наук; Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 6, Харьков, Украина, 61022; e-mail: o.moroz@karazin.ua; ORCID: 0000-0002-4920-4093.

Научные интересы: .

**УДК 539.534.9:523.23**

ПАВЛЕНКО В.И., МАРЧЕНКО И.Г., ЖУКОВ А.И.

# МНОГОУРОВНЕВОЕ МОДЕЛИРОВАНИЕ ОСАЖДЕНИЯ ПЛЕНОК NB ИЗ ИОННО-АТОМНЫХ ПОТОКОВ

## ВВЕДЕНИЕ

Известно, что физические свойства пленок тугоплавких металлов, получаемые методом вакуумного осаждения, связаны с микроструктурой материала пленки, состояние которой в значительной степени определяется параметрами процесса осаждения, такими как, энергия и тип падающих ионов, плотность ионного тока и др. При таком способе осаждении пленок, когда температура на подложке $T \le 0{,}3T_{пл.}$ (где $T_{пл.}$ – температура плавления конденсируемого материала) образуются, как правило, пористые пленки тугоплавких металлов [1–4]. Полученные конденсаты имеют довольно низкие физико-механические свойства. Так, например, плотность таких пленок может отличаться от объемной плотности на величину ~ 15% [4].

Для управления структурой пленки, формируемой из ионно-атомных потоков, необходимо понимать, как она зависит от энергии и вида падающих потоков ионов ($E_{ion}$), степени ионизации облучаемого потока частиц $\gamma_{ion}$ ($\gamma_{ion} = J^{ион}/J_{общ}$, где $J^{ион}$ – плотность только ионного потока частиц, $J_{общ} = J^{атом} + J^{ион}$ – общая плотность атомно-ионного потока, $J^{атом}$ – плотность только атомного потока), температуры подложки (Т) и других параметров.

Несмотря на большое количество экспериментальных данных, до конца механизм формирования микроструктуры пленок не изучен.

Перспективным способом изучения сложных процессов в неравновесных системах является метод многоуровневого моделирования. В данной работе мы используем комплекс программ с различными характерными масштабами времени. Данные о первичном дефектообразовании в пленках, с характерной длительностью процесса $\sim 10^{-12}$с, получались с помощью программы SPURT.CRIS [5]. Диффузионные процессы, с характерными временами порядка секунд описывались системой дифференциальных уравнений в частных производных программой RADI-DIFUS.

Целью данной работы являлось исследование поведения плотности формируемой в процессе осаждения пленки, при изменении энергии атомно-ионного потока частиц и степени ионизации этого потока с использованием метода многоуровневого моделирования.

## КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ И ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Методика моделирования включала в себя два этапа. На первом этапе, с помощью программного комплекса SPURT.CRIS, созданного для моделирования процессов первичного дефектообразования в сложных неравновесных системах в процессе облучения наноструктурной пленки низкоэнергетичными ($E \le 1$–3 кэВ) потоками ионов в широком интервале углов облучения ($\alpha = 0^0$–$80^0$), получали все величины и параметры, которые необходимы для работы программы RADI-DIFUS.

Эти данные в дальнейшем использовались на втором этапе компьютерных исследований в программе RADI-DIFUS как исходные данные в процессе расчета плотности пленок, формируемых при низкоэнергетическом осаждении в вакууме из собственных атомно-ионных потоков. Программа RADI-DIFUS непосредственно реализует процесс уплотнения пористых пленок, осаждаемых из атомно-ионных потоков, с одновременной бомбардировкой пленки ионами. Программа RADI-DIFUS (в основе которой лежит решение уравнений диффузии) создана для реализации, разработанной авторами [6-7], теоретической модели радиационно-диффузионного уплотнения пленок (РД-модель) и позволяет получать количественные и качественные характеристики, описывающие поведение плотности формируемой при осаждении пленки, при изменении параметров ионного облучения. РД-модель описывает поведение

плотности пленок при низкотемпературном и низкоэнергетическом вакуумном осаждении из паровой фазы.

Были рассчитаны плотности наноструктурной пленки Nb, осаждаемой из собственных ионно-атомных потоков $Nb^+$–Nb. Начальная плотность ниобиевой пленки, сформированной без ионного облучения, принималась равной $\rho_{film}^0$=0,85 ($\rho_0$ – плотность ниобиевой подложки). Это соответствует экспериментальному значению, полученному в работе [8] при электронно-лучевом испарении Nb с последующим осаждением металлического пара на подложку с температурой T = (373±20K) при потоке частиц $J_{общ} = 3{,}3\times10^{16}см^{-2}с^{-1}$.

Энергия ионов в осаждаемых потоках изменялась в интервале 0,050 кэВ ≤ E ≤0,750 кэВ, а степени ионизации – в интервале $\gamma_{ion}$ = 0 – 0,35.

На рис. 1 представлены результаты моделирования по программе RADI-DIFUS плотности пленки Nb в зависимости от энергии $E_{ion}$ ионов при осаждении собственных атомно-ионных потоков ниобия на подложку со степенью ионной компоненты $\gamma_{ion}$=0,1. Результаты расчета даны в сравнении с экспериментальными данными из работы [8]. При этом следует иметь в виду, что при компьютерном расчете в диффузионные уравнения RADI-DIFUS подставлялись профили пространственного распределения, полученные при облучении модельной наноструктурной пленки (программа SPURT.CRIS). В экспериментах [8] исследовались поликристаллические конденсаты ниобия, полученные осаждением ионно-плазменных $Nb^+$–Nb потоков методом атомно-ионного распыления.

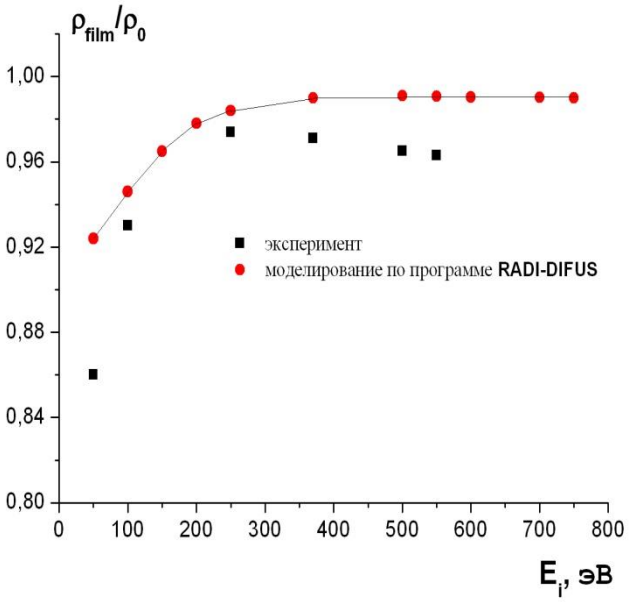



*Рис. 1. Рассчитанные по программе RADI-DIFUS и полученные из эксперимента [8] значения плотности Nb-пленки, осаждаемой при различных энергиях облучения $E_i \le 750$ эВ (степень ионизации падающего потока частиц $\gamma_i$=0,1; температура подложки 373±20K).*

Из рис. 1 видно, что наличие в осаждаемом потоке частиц ионов приводит к увеличению плотности осаждаемой пленки. Так при низко интенсивной степени ионизации, составляющей в осаждаемом потоке $\gamma_{ion}$=0,1 с повышением энергии падающих ионов $E_i$ вплоть до ~ 350 эВ, плотность пленки монотонно растет и практически достигает значения плотности $\rho_0$ в объеме материала. Дальнейшее повышение энергии $E_i$ не приводит к увеличению плотности пленки. Таким образом, при заданных начальных условиях, наибольшее ионное уплотнение в формируемой пленке достигается при энергии осаждаемого потока частиц $E_i$ ~350 эВ. Т.е. в интервале энергий 50 эВ ≤ E ≤350 эВ происходит активное замещение вакансионных пор собственными междоузельными атомами ниобия в формируемой пленке с выходом на «насыщение» при энергиях ~ 350 эВ.

Изменение плотности $\rho_{film}$ от $E_i$ с достаточной степенью точности может быть описано зависимостью типа $E_i^n$, где $n$~0.44. Полученная зависимость подобна зависимости сечения ядерного торможения ионов в металлах $S_n(E) \sim E^{1/2}$ при энергии $E \leq 1$ кэВ [9]. Это означает, что величина уплотнения пленки контролируется кинетикой точечных дефектов, а их концентрация определяется эффективностью процессов упругого ионно-атомного и атомно-атомного взаимодействий. При облучении ионами с энергией ~ 350 эВ на подложке формируются плотные Nb конденсаты. Дальнейшее повышение энергии, как видно из рисунка, не оказывает заметного влияния на изменение плотности. Следовательно, $E_i \cong 350$ эВ является оптимальной энергией для получения плотных наноструктурных Nb пленок при ионной бомбардировке с интенсивностью потока $\gamma_{ion}$=0,1.

При этом заметим, что программа RADI-DIFUS дает стабильно более высокие значения плотности пленки, чем эксперимент. Также отметим, что в эксперименте [8] плотность пленки достигает плотности $\rho_0$ в объеме материала при энергии ~ 250 эВ.

Влияние степени ионизации падающего потока частиц $\gamma_{ion}$ на изменение плотности конденсирующейся пленки Nb демонстрирует рис. 2. На рисунке показана, рассчитанная по программе RADI-DIFUS, в сравнении с экспериментальными данными [8], зависимость плотности Nb–пленок от степени ионизации потока. Энергия ионов $E_i$=250 эВ, температура подложки T=(373±20)К. Рис. 2 демонстрирует, что при значениях параметра $\gamma_{ion} \geq 0{,}1$ плотность формируемой ниобиевой пленки достигает максимума и не зависит от дальнейшего увеличения ионного потока. В интервале значений $0 < \gamma_{ion} < 0{,}1$ обе кривые (рассчитанная по программе RADI-DIFUS и экспериментальная) показывают практически линейный рост плотности с возрастанием степени ионизации осаждаемого потока частиц.

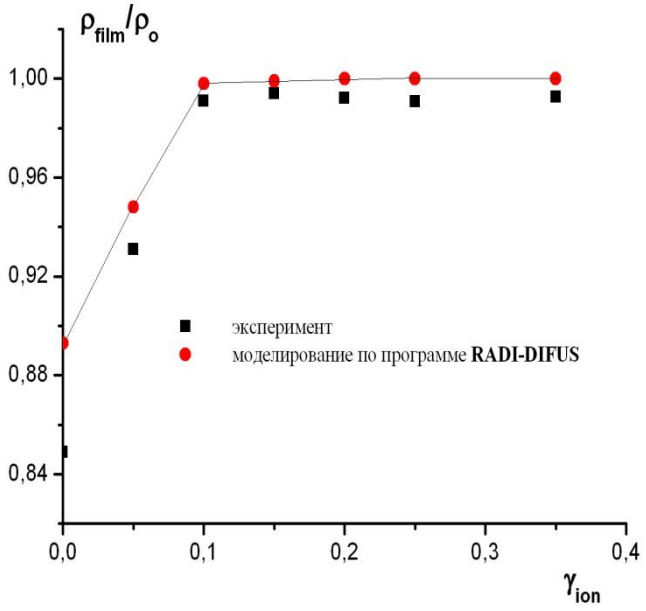



*Рис. 2 Рассчитанные по программе RADI-DIFUS и полученные из эксперимента [8] значения плотности Nb-пленки, осаждаемой при различных значениях степени ионизации падающего потока частиц Nb (энергия ионов $E_i$=250 эВ; температура подложки 373±20К).*

## ВЫВОДЫ

С помощью метода многоуровневого моделирования изучено изменение плотности пленок ниобия в зависимости от энергии ионов и степени ионизации атомно-ионного потока. Установлено, что при степени ионизации потока $\gamma_i$=0,1 процесс уплотнения формируемой пленки происходит в интервале энергий 50 – 350 эВ, что объясняется активно идущими процессами вбивания собственных междоузельных атомов в вакансионные поры.

Показано, что при осаждении ниобиевой пленки потоком частиц с энергией $E_i < 200$ эВ в интервале $0 < \gamma_{ion} < 0{,}1$ плотность пленки прямо пропорциональна степени ионизации потока. При увеличении степени ионизации потока частиц ($\gamma_{ion} \geq 0{,}1$) уплотнение пленки выходит на стационар и дальнейшее увеличение $\gamma_i$ его не меняет.

ЛИТЕРАТУРА

1. Guglya A.G., Marchenko I.G. Ion beam-assisted deposition. Comprehensive guide for nanocoatings technology. *Nova Science Publishers.* New York. 2015. V. 1. P. 45-69.
2. Marchenko I. G. Computer simulation of the formation of niobium film nanostructure by low-temperature deposition. *Vacuum.* 2007. V. 81. P. 700-707.
3. Marchenko I. G., Neklyudov I.M. Film nanostructure formation during low-temperature PVD deposition using partially ionized atomic fluxes. *Journal of Physics: conference series*. 2008. V. 113. P. 2-8.
4. Марченко И.Г., Марченко И.И., Неклюдов И.М. Компьютерное моделирование вакуумного осаждения пленок ниобия. *Вестник Харьковского университета*, 2004. №628, С. 93-98.
5. Павленко В.И., Марченко И.Г. Компьютерное моделирование профилей имплантированных ионов $Al^+$ в наноструктурную пленку Cu. *Вопросы атомной науки и техники. Серия: «Физика радиационных повреждений и радиационное материаловедение»*. 2017. №4 (110). С. 32-38.
6. Bakai A. S., Sleptsov S.N., Zhukov A.I., Marchenko I.G., Sleptsov A.N. Mathematical modeling of the densification of niobium film deposited from self-ion-atomic fluxes *Met. Phys. Adv. Tech.* 1996. V. 15. P. 1329-1342.
7. Bakai A. S., Zhukov A.I., Sleptsov S.N., Marchenko I.G., Sleptsov A.N., Reznichenko A.N. Low-temperature densification of chromium films induced by bombardment with argon and chromium ions. *Met. Phys. Adv. Tech.* 1996. V. 16. P. 99-109.
8. Слепцов С.Н., Марченко И.Г., Булатова Л.В., Слепцов А.Н., Поляков Ю.И. Структурное состояние толстых конденсатов ниобия, осаждаемых из собственных атомно-ионных потоков. *Вопросы атомной науки и техники. Серия: «Физика радиационных повреждений и радиационное материаловедение».* 1993. Вып. 1(60). С. 62-69.
9. Sigmund P. Theory of sputtering, 1. Sputtering yield of amorphous and polycrystalle targets. *Phys. Rev.* 1969. V. 124. P. 383-416.

**МАРЧЕНКО Иван Григорьевич** – д.ф.-м.н., ведущий научный сотрудник Национального научного центра «Харьковский физико-технический институт», профессор кафедры физики нетрадиционных энерготехнологий и экологии Харьковского национального университета имени В.Н. Каразина, ул. Академическая, 1, Харьков-108, Украина, 61108; e-mail: march@kipt.kharkov.ua; ORCID: 0000-0003-1341-4950.

Научные интересы:

*компьютерное моделирование процессов происходящих в твердых телах, стохастические процессы, физика поверхности, физика радиационных повреждений.*

**ПАВЛЕНКО Владимир Иванович** – ведущий инженер-исследователь Национального научного центра «Харьковский физико-технический институт», ул. Академическая, 1, Харьков-108, Украина, 61108; e-mail: ruslana_olirna2005@ukr.net; pavlenko@kipt.kharkov.ua; ORCID: 0000-0003-0210-3268.

Научные интересы:

*Математическое моделирование физических процессов происходящих в покрытиях (пленках) при ионном облучении, физика поверхности.*

**ЖУКОВ Александр Иванович** – к.ф.-м.н., старший научный сотрудник Национального научного центра «Харьковский физико-технический институт», ул. Академическая, 1, Харьков-108, Украина, 61108; e-mail: azhukov@kipt.kharkov.ua.

Научные интересы:

*Физика радиационных повреждений, компьютерное моделирование процессов, происходящих в твердых тела при облучении.*

**УДК 004.056.55**

ПАЗУШКО М.А, БОБУХ В.А.

# ЗАГАЛЬНА СУТНІСТЬ MQ-ПЕРЕТВОРЕНЬ

## Вступ

На сьогоднішній день проходить конкурс на створення нових стандартів постквантової асиметричної криптографії – NIST США. Квантові комп'ютери ще недостатньо потужні, але вони швидко еволюціонують, що робить можливим криптоаналіз існуючих асиметричних криптоперетворень.

На відміну від класичних комп'ютерів, що використовують біти та здатні приймати лише значення 1 або 0, квантові машини використовують кубіти, здатні одночасно представляти різні можливі стани, проміжні між 0 і 1 (суперпозиція). Вони також можуть впливати один на одного на відстані завдяки такому явищу, як заплутаність.

Алгоритм Шора загрожує таким системам з відкритим ключем, як RSA, чий математичний захист, залежить від того, наскільки складно провести зворотний інжиніринг результату перемноження дуже великих простих чисел (розкладання на множники). У звіті з квантових обчислень, опублікованому в 2017 році національною академією наук, інженерної справи і медицини США, передбачається, що потужний квантовий комп'ютер, на якому працює алгоритм Шора, зможе зламувати 1024-бітні варіанти RSA менш, ніж за день.

Без криптографічного захисту, що враховує квантові обчислення, усім сервісам – від військового обладнання до фінансових транзакцій і комунікацій – загрожуватимуть атаки з боку хакерів, які отримали доступ до квантових комп'ютерів.

Одним з типів постквантових алгоритмів, які можуть зайняти місце теоретико-числових завдань (таких як факторизація, дискретний логарифм) є MQ-перетворення, тобто схеми розроблені на базі використання багатовимірних квадратичних перетворень (Multivariate Quadratic Transformations). На конкурсі NIST представлено такі алгоритми на MQ-перетвореннях: MQDSS, LUOV, Rainbow, GeMSS. Ці схеми, побудовані на вирішенні системи рівнянь, заснованих на багатовимірних поліномах над кінцевим полем F.

## Побудова MQ-схеми

Сутність побудови такої схеми полягає у виборі системи на скінченому $\mathbb{F}_q$ полі з $q$ елементами (центральне відображення) на полі з $m$ багатовимірних квадратичних багаточленів від $n$ змінних [1]. На наступному етапі обираються два випадкових афінних лінійних перетворення $S: \mathbb{F}^m \to \mathbb{F}^m$ та $\mathcal{T}: \mathbb{F}^n \to \mathbb{F}^m$ для приховування центрального відображення у відкритому ключі, що в свою чергу визначається як квадратичне перетворення виду $P = S \cdot F \cdot \mathcal{T}$:

$$P(x_1, \dots, x_n) = \begin{pmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_m(x_1, \dots, x_n) \end{pmatrix}, \quad (1)$$

де $f_i \in \mathbb{F}[x_1,\dots,x_n]$ – багаточлен другого порядку [2, 3].

Особистий ключ складається з трійки $(S, P', \mathcal{T})$, де $S \in AGL_n(\mathbb{F})$, $\mathcal{T} \in AGL_m(\mathbb{F})$ афінні перетворення і $P' \in MQ_m(\mathbb{F}^n)$– поліном-вектор з $P' = (p_1', \dots, p_m')$ $m$ компонентами, кожен з яких є багаточленом з $n$ змінних $x_1', \dots, x_n'$. На відміну від $P' \in MQ_m(\mathbb{F}^n)$, особистий поліном-вектор $P'$ дозволяє ефективно обчислити $x_1', \dots, x_n'$ для даного $y_1', \dots, y_m'$.

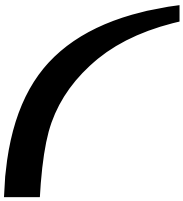
MQ-схеми розрізняються в побудові центральних відображень і, отже, вхідного апарата, який вони вбудовують у конкретний клас MQ-проблеми.

## MQ-проблема

Безпека багатовимірної криптографії ґрунтується на припущенні, що рішення системи квадратичних багаточленів над кінцевим полем F, в загальному випадку, є NP-повною.

Проблема багатовимірного квадратичного поліному (MQ) є основою безпеки для потенційно постквантових криптосистем. Складність вирішення визначається рядом параметрів: кількістю змінних та рівнянь, розміру базового поля тощо.

Криптографічна стійкість (безпека) залежить від складності рішення системи багатовимірних поліноміальних багаточленів з коефіцієнтами у кінцевому полі :



(2)

Рішення має вигляд у полі даної системи [4].

Наявність великого особистого (і, отже, відкритого) ключового простору є бажаною властивістю для будь-якої схеми відкритого ключа. Багато схем на основі багатовимірних квадратичних поліноміальних рівнянь мають велику кількість "еквівалентних" особистих ключів. Отже, у них багато зайвих особистих ключів і, відповідно, менший простір особистого та відкритого ключа, ніж очікувалося. Для цього обираються "стійкі перетворення" [5].

Два особисті ключі є "еквівалентними", якщо вони призводять до одного і того ж відкритого ключа, тобто:

$$\mathcal{T} \cdot P' \cdot S = P = \tilde{\mathcal{T}} \cdot \tilde{P}` \cdot \tilde{S} \tag{3}$$

Нехай , $(S, P', \mathcal{T}) \in AGL_m(\mathbb{F}) \times MQ_m(\mathbb{F}^n) \times AGL_n(\mathbb{F})$ де і $\sigma, \sigma^{-1} \in AGL_n(\mathbb{F})$ $\tau, \tau^{-1} \in AGL_m(\mathbb{F})$. Також $P = \mathcal{T} \cdot \tau^{-1} \cdot \tau \cdot P' \cdot \sigma \cdot \sigma^{-1} \cdot S = (\sigma, \tau) \cdot (S, P', \mathcal{T})$.

Пара $(\sigma, \tau) \in AGL_n(\mathbb{F}) \times AGL_m(\mathbb{F})$ називається "стійким перетворенням" для MQ-системи, якщо "форма" $P'$ інваріантна до перетворення $\sigma$ і $\tau$, і позначається $(\sigma, \tau)$.

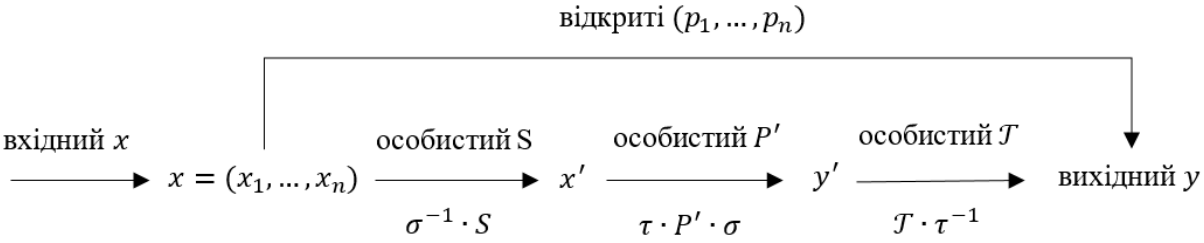



*Рис. 1 Еквівалентні особисті ключі з використанням афінних перетворень* σ, τ

Нехай (σ, τ) – стійке перетворення. Якщо $G := (\sigma, \circ)$ і $H := (\tau, \circ)$ утворюють підгрупу афінних перетворень, вони створюють співвідношення еквівалентності в просторі особистого ключа (бо маємо властивості рефлексивності, симетрії і транзитивності).

**Афінні перетворення**

Нехай – кінцеве поле з $q = |\mathbb{F}|$ елементів. Тоді $\prod_{i=0}^{n-1} (q^n - q^i)$ – обернена $(n \times n)$ матриця над полем .

Матричне зображення афінного перетворення *S*: нехай є $M_s \in \mathbb{F}^{n \times n}$ оберненою $(n \times n)$ матрицею і $v_s \in \mathbb{F}^n$ – вектор, і нехай $S(x) = M_s x + v_s$.

Багатовимірне зображення афінного перетворення *S*: нехай $s_1,\dots,s_n$ – n поліномів першого порядку, тобто $s_i(x_1,\dots,x_n)=\beta_i x_1+\dots+\beta_i x_n+\alpha_i$, при $1\le i,j\le n$ та $\beta_{i,j}\in\mathbb{F}$ .Тоді $S(x)=(s_1(x),\dots,s_n(x))$ при $x=(x_1,\dots,x_n)$ - вектор над полем $\mathbb{F}^n$.

Одномірне зображення над розширеним полем $\mathbb{E}$ афінного перетворення : $S(x)$ нехай $0\le i\le n$ та $A,B_i\in\mathbb{E}$, тоді $S(x)=\sum_{i=0}^{n-1}B_i x^{q^i}+A$ .

Афінне перетворення в одномірному зображенні може бути ефективно перенесене у багатовимірне і навпаки [5].

**Висновки**

1. Можливо ефективно перенести афінне перетворення в одномірному зображенні у багатовимірне і навпаки.

2. Багатовимірне та матричне зображення афінного перетворення *S* взаємозамінні. Перше більш доречно при роботі з матричними рівняннями, друге з афінними перетвореннями в контексті заміщення.

3. Багатовимірні квадратичні системи дозволяють отримати багато еквівалентних особистих ключів і, отже, мати велику надмірність у своїх ключових просторах. Зменшення розмірів представляє нижню, а не верхню межу: додаткові стійкі перетворення можуть ще більше зменшити ключовий простір схем.

ЛІТЕРАТУРА

1. Shuaiting Qiao Construction of Extended Multivariate Public Key Cryptosystems / Shuaiting Qiao, Wenbao Han, Yifa Li, Luyao Jiao // International Journal of Network Security. – Vol. 18. – No.1. P. 60-67. – Режим доступу: https://pdfs.semanticscholar.org/317e/3f52d4666ceb667151fdb4295f2972eaf33d.pdf?_ga=2.246085124.1075338183.1584345818-1588415786.1584345818.
2. Lih-Chung Wang A Medium-Field Multivariate Public-Key Encryption Scheme / Lih-Chung Wang, Bo-Yin Yang, Yu-Hua Hu, and Feipei Lai // – Режим доступу: https://link.springer.com/chapter/10.1007%2F11605805_9.
3. Горбенко Ю.І. Порівняння кандидатів електронного підпису на постквантовий стандарт NIST PQC на базі MQ-перетворень та функцій гешування / Ю.І. Горбенко, І.С. Кудряшов, Д.С. Науменко, В.В. Онопрієнко // Прикладна радіоелектроніка. – Х. : Харківський національний університет радіоелектроніки, 2018. – Том 17, № 3, 4. – С. 138-146.
4. Fukuoka MQ Challege. News. – Електронний ресурс. – Режим доступу: https://www.mqchallenge.org.
5. Christopher Wolf Superfluous Keys in Multivariate Quadratic Asymmetric Systems Christopher / Christopher Wolf, Bart Preneel // – Режим доступу: https://eprint.iacr.org/2004/361.pdf.

**ПАЗУШКО Марина Андріївна** – студентка кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків, Харківська область, 61000; e-mail: pazushkomr@ukr.net; ORCID: 0000-0001-7279-002X.

Наукові інтереси:

*– криптографічний захист інформації, MQ-перетворення.*

**БОБУХ Всеволод Анатолійович** – к.т.н., начальник відділу апаратних засобів АТ «ІІТ»; АТ «ІІТ», вул. Бакуліна, 12, Харків, Харківська область, 61166; e-mail: bobukhv@iit.kharkov.ua.

Наукові інтереси:

*– криптографічний захист інформації, постквантові криптографічні примітиви.*

**УДК 004.7**

ПЕЛЫХ Д.А., ПАВЛОВ А.Н.

# МОДЕЛЬ ИНФОРМАЦИОННО-СЕРВИСНОЙ СЛУЖБЫ

**Общее определение информационно-сервисной службы**

Информационно-сервисной службой называется совокупность действий, связанных с обработкой, хранением и использованием данных. Модель информационно сервисной службы разрабатывается для того, чтобы заблаговременно продумать ее характеристики с учетом возможных изменений, которые могут возникнуть в ходе ее эксплуатации

**Требования к информационно-сервисной службе**

Информационно-сервисная служба доставки еды должна иметь четыре действующих лица, а именно: клиент, менеджер, предоставтель услуг и доставщик. Каждому из них отведены роли и набор действий, которые должны или могу выполняться.

Предлагаю рассмотреть функциональные требования к информационно-сервисной службе с помощью User story и Use case диаграммы. С помощью use case диаграммы будут показаны действия, которые смогут совершить клиент (Рис. 1), менеджер (Рис. 2), предоставитель услуг (Рис. 3) и доставщик (Рис. 4)

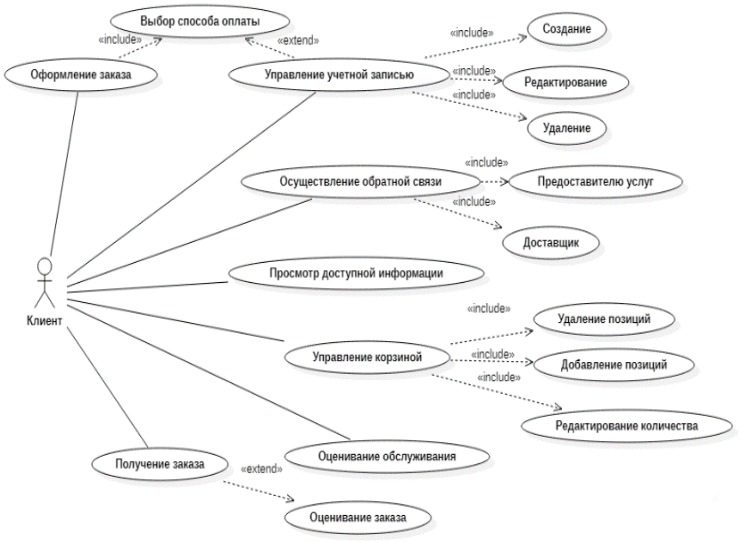



*Рис. 1 Диаграмма прецедентов для сущности "Клиент"*

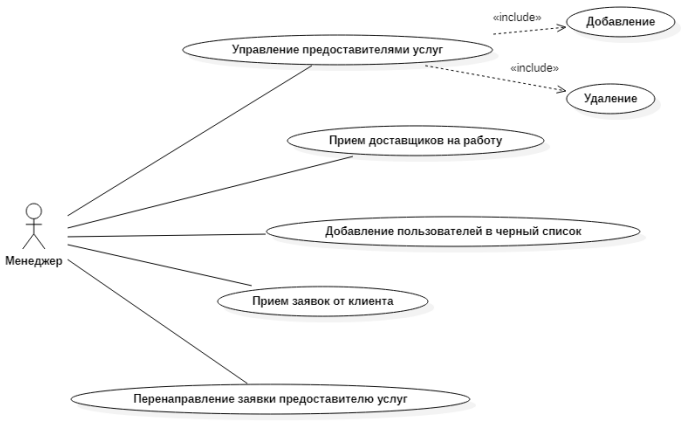



*Рис. 2 Диаграмма прецедентов для сущности "Менеджер"*

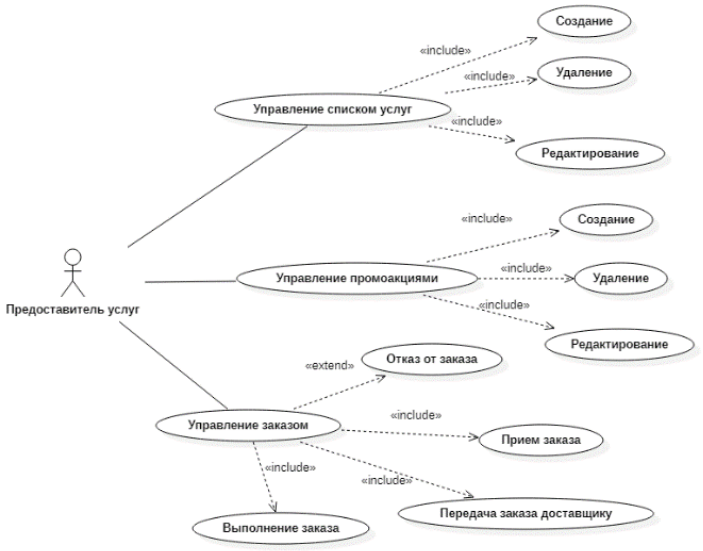



*Рис. 3 Диаграмма прецедентов для сущности "Предоставитель услуг"*


*Рис. 4 Диаграмма прецедентов для сущности "Доставщик"*

User Story диаграмма нужна для полного понимания всех возможных действий каждого элемента информационно-сервисной службы. Ниже приведены возможности для клиента, менеджера, предоставителя услуг и доставщика.

**User Story для клиента:**

Как клиент я могу управлять учетной записью для того, чтобы создавать редактировать или удалять информацию из аккаунта

Как клиент я могу просматривать доступную информацию для того, чтобы иметь возможность сделать выбор

Как клиент я могу получить обратную связь для того, чтобы узнать нужную информацию у доставщика или предоставителя услуг

Как клиент я могу управлять корзиной для того, чтобы удалять, добавлять или редактировать ее содержимое

Как клиент я могу оценить обслуживание для того, чтобы улучшить качество обслуживания

Как клиент я могу выбрать способ оплаты в учетной записи для того, чтобы не вводить эти данные при каждом заказе

Как клиент я могу получить заказ для того, чтобы воспользоваться им

**User Story для менеджера:**

Как менеджер я могу управлять карточками представителей услуг для того, чтобы удалять или добавлять их для удобства клиентов

Как менеджер я могу принимать доставщиков на работу для того, чтобы доставлять заказы клиентам

Как менеджер я могу добавлять пользователей в черный список для того, чтобы улучшить качество сервисной службы

Как менеджер я могу принимать заявки от клиентов для того, чтобы перенаправить их предоставителю услуг

**User Story для предоставителя услуг:**

Как предоставитель услуг я могу управлять списком услуг для того, чтобы создавать, удалять или редактировать его

Как предоставитель услуг я могу управлять промакциями для того, чтобы создавать, удалять или редактировать их

Как предоставитель услуг я могу управлять заказом для того, чтобы отказаться, выполнить, принять или передать его доставщику

**User Story для доставщика:**

Как доставщик я могу зарегистрироваться сервисной службе для того, чтобы иметь возможность брать заказы и доставлять их

Как доставщик я могу принимать заявку от клиента для того, чтобы забрать нужный заказ у предоставителя услуг

Как доставщик я могу принять заказ у предоставителя услуг для того, чтобы передать его клиенту

Как доставщик я могу выдать заказ клиенту для того, чтобы получить оплату и завершить его

**Структура информационно-сервисной службы**

Выражение структуры информационно-сервисной службы представлено с помощью Data Flow диаграммы.

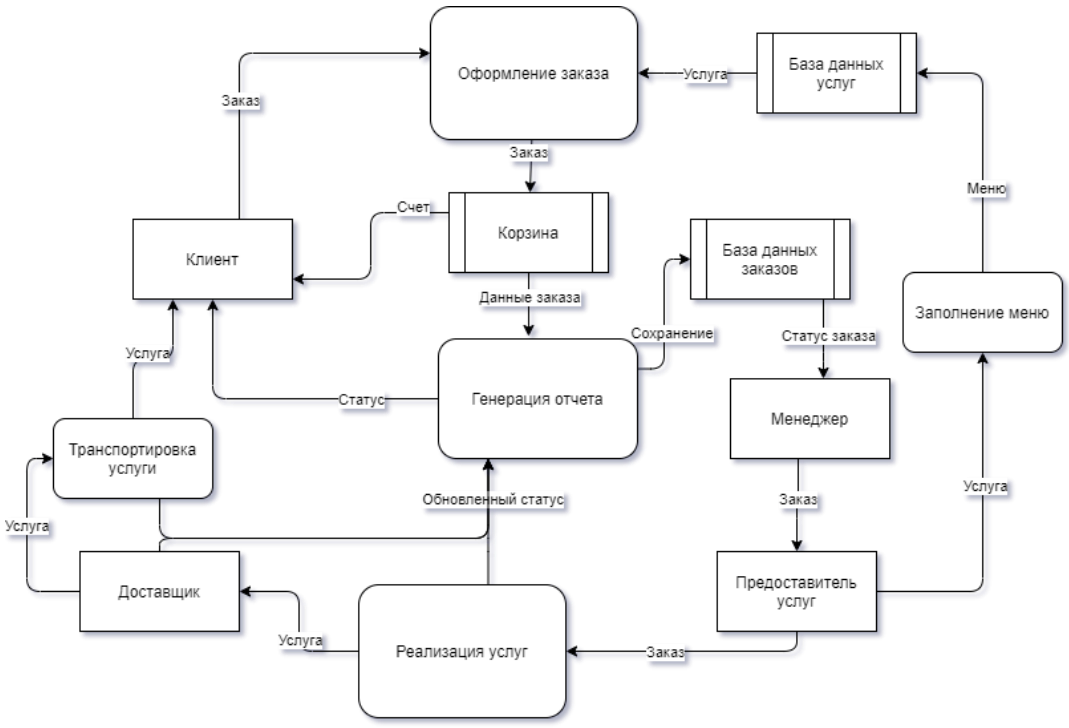



*Рис. 5 Структура информационно-сервисной службы*

**Алгоритм работы (модель функционирования) информационно-сервисной службы**

Алгоритм работы информационно сервисной службы изображен с помощью диаграммы последовательности в нотации UML (Рис. 6)

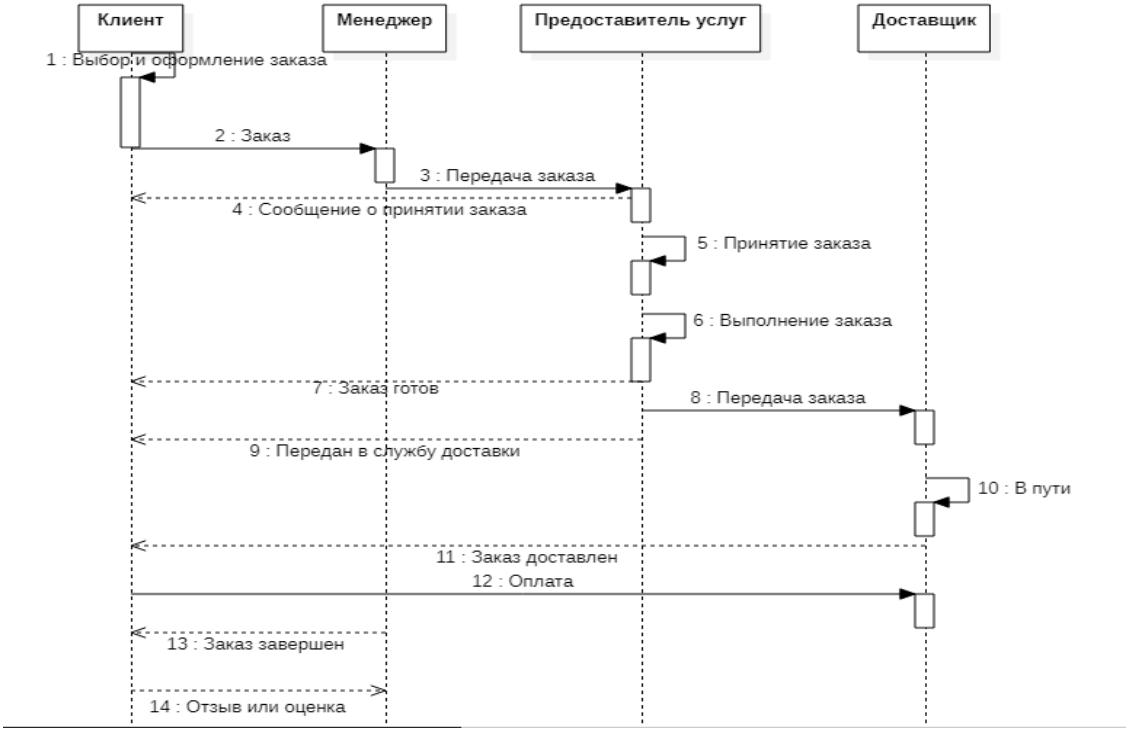



*Рис. 6 Диаграмма последовательности информационно-сервисной службы*

ЛИТЕРАТУРА

1. Лосев Ю. І., Руккас К. М., Шматков С. І. Комп'ютерні мережі: навч. посіб. / за редакцією Ю. І. Лосева. Харків : ХНУ імені В. Н. Каразіна, 2013. 157 с.
2. Крег Ларман Применение UML 2.0 и шаблонов проектирования. 3-е издание, 2019. 37с.
3. А. Н. Калашян, Г. Н. Калянов Структурные модели бизнеса: DFD-технологии, 2009. 105с.

**ПЕЛЫХ Дария Александровна** – студентка кафедры теоретической и прикладной системотехники, Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 4, Харьков-22, Украина, 61022; e-mail: dariyapelykh@gmail.com; ORCID: 0000-0001-9086-4926.

Научные интересы:

*– разработка веб-сайтов, моделирование информационных систем.*

**ПАВЛОВ Анатолий Николаевич** – старший преподаватель кафедры теоретической и прикладной системотехники; Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 4, Харьков-22, Украина, 61022; e-mail:tps@karazin.ua.

Научные интересы:

– математическое моделирование системы расчета основных параметров серверных вычислительных систем и, моделирование информационных систем.

**УДК 004.056.55**

ПИСАРЕНКО Н. В, ГОРБЕНКО І. Д.

# АНАЛІЗ АЛГОРИТМУ ЦИФРОВОГО ПІДПИСУ CRYSTALS-DILITHIUM ТА УМОВ ЙОГО ЗАСТОСУВАННЯ

**Вступ**

На сьогодні вирішується серйозна задача у створенні постквантових криптографічних методів цифрового підпису (ЦП). Дані розробки пов'язані зі створенням 20, 53 та 72-кубітних квантових комп'ютерів.

Саме тому виникла необхідність для розгляду та аналізу існуючих на сьогоднішній день криптографічних алгоритмів, заміни їх параметрів або збільшення розміру цих параметрів, а також необхідність створення нових стандартизованих криптографічних алгоритмів з урахуванням можливостей квантового комп'ютера.

Для вирішення цієї проблеми на світовому рівні проводиться міжнародний конкурс NIST США, основною метою якого є відбір нових стандартів для ЕП у постквантовий період. На 2-му етапі залишилось 9 алгоритмів ЦП із 17, що брали участь у 1-му етапі: CRYSTALS-DILITHIUM, FALCON, qTESLA, GeMSS, LUOV, MQDSS, Picnic, Rainbow та SPHINCS+.

На наш погляд, одним із перспективних ЦП є підпис на основі алгебраїчних решіток, який називають CRYSTALS-Dilithium.

**Загальні положення ЕП на алгебраїчних решітках CRYSTALS-Dilithium**

Dilithium – це схема ЦП, яка має надійну безпеку від атак на вибране повідомлення, що базуються на складності проблем решітки над модульними решітками. Поняття безпеки означає, що порушник, який має доступ до оракула підпису, не може виробити підпис повідомлення, підпис якого він ще не бачив, а також не може створити інший підпис повідомлення, яке він вже бачив підписаним. Dilithium – один із алгоритмів-кандидатів, що був представлений на конкурсі NIST PQC та досліджується на 2-му етапі конкурсу.

Конструкція Dilithium базується на методі «Fiat-Shamir з перериваннями» Любашевського, який використовує вибірку відхилення, для того, щоб зробити схеми Fiat-Shamir на основі решіток компактними та безпечними.

Схема Dilithium була розроблена з урахуванням наступних критеріїв:

1. Простий для безпечної реалізації.
2. Консервативний за параметрами.
3. Зменшення розміру відкритого ключа та підпису.
4. Відносно легко змінювати безпеку.

Тому дуже ефективні реалізації алгоритму повинні оптимізувати ці операції та виконуватись за постійний час. Для всіх рівнів безпеки ця схема використовує одне і те ж кільце $q=2^{23}-2^{13}+1$ та $n=256$. Зміна безпеки полягає в тому, щоб робити більше/менше операцій над цим кільцем і робити більше/менше розширення XOF.

**Сутність основних відмінностей ЦП CRYSTALS-Dilithium 2-го раунда конкурсу NIST США**

Під час аналізу ЦП CRYSTALS-Dilithium, що був поданий на 1-й та 2-й раунди конкурсу, було виявлено, що найголовнішою відмінністю підпису у 2-му раунді є пропозиція до реалізації детермінованої версії ЦП, що закладена в основу версії 1-го раунду і додати додатково рандомізовану версію.

Різниця між цими версіями механізму полягає у тому, що початкова ентропія удосконаленого ЦП, тобто у рандомізованій версії, може встановлюватись випадковим чином, тоді як у першій, детермінованій версії, вона визначалась тільки у вигляді геш-значення ключа та безпосередньо повідомлення M.

Інші зміни були зроблені в реалізації. Вони стосуються оптимізації в алгоритмі підписання. Найважливіша оптимізація стосується обчислення умови відхилення на основі нижньої частини вектора w і вектора підказки.

Також у версії 2-го раунду, для розгортання матриці A та генерування секретних векторів $s_1$, $s_2$ і вектора маскування y, замість алгоритму гешування SHAKE може використовуватись стандарт симетричного шифрування AES. Це зроблено для підвищення ефективності та уніфікації ЦП Crystals-Dilithium, наприклад, засобом використання апаратної підтримки для реалізації SHAKE та AES.

Проведений аналіз показав, що з урахуванням 1 та 2 версій, механізм ЦП Dilithium був розроблений та удосконалений з урахуванням, в тому числі наведених вище вимог.

**Умови застосування алгоритму CRYSTALS-Dilithium**

Нижче наведені постійні параметри Crystals-Dilithium.

N=256 – степінь поліному

q=223−213+1=8380417– більший модуль перетворення коефіцієнтів поліномів

QBITS=23 – значення $\log_2 q$

D=14 – параметр, який використовується при виділенні старшої і молодшої частини 32 бітного слова

$\gamma_1$ =((q–1)/16)=523776

$\gamma_2$ =(GAMMA1/2)=261888

ALPHA=(2* $\gamma_2$ )–523776 (19 бітів) параметри, які використовуються при обчисленні норми

Параметри, які можуть змінюватися у залежності від вимог, що висуваються до стійкості та складності криптографічних перетворень при виконанні ЕП, наведено в таблиці 1.

*Табл.1 Параметри, що можуть змінюватись*

| Модель | K | L | η | ζ | β | ω |
|---|---|---|---|---|---|---|
| 0 | 3 | 2 | 7 | 4 | 375 | 64 |
| 1 | 4 | 3 | 6 | 4 | 325 | 80 |
| 2 | 5 | 4 | 5 | 4 | 275 | 96 |
| 3 | 6 | 5 | 3 | 3 | 175 | 120 |

В таблиці 1 параметри K та L визначають розмір матриці A, η – основа алфавіту ключових даних, β, ω та ζ – додаткові параметри забезпечення стійкості та оптимізації.

Далі наводимо узагальнений алгоритм генерації ключів.

В процесі генерування ключів послідовно виконуються такі кроки:

1. Генерація випадкових послідовностей у вигляді відкритого початкового ключа ρ (32 байти) та початкових секретних ключів ($\rho_1$ та Key) (32*2=64 байти), що визначають початкову ентропію системи генерування відкритих та особистих (секретних) ключів ЦП Dilithium.

2. Генерація відкритої матриці поліномів A розміром [K][L][256] з використанням відкритого ключа ρ (32 байти) для значень K та L, що наведені в таблиці 1.

3. Генерація особистих (секретних) ключів у вигляді масиву поліномів $S_1$[L][256] та $S_2$[K][256] з використанням секретного початкового ключа $\rho_1$ (32 байти).

4. Обчислення масиву t із K поліномів, використовуючи матрицю поліномів A та секретні ключі $S_1$ і $S_2$ згідно формули $t=A*S_1+S_2$.

5. Представлення кожного коефіцієнту поліному масиву поліномів (векторів) t у вигляді старших (high) та молодших (low) бітів $t[i][j]=\{t[i][j]_{high}, t[i][j]_{low}\}$ згідно формули

$$t[i][j]=t[i][j]_{high}*2D+t[i][j]_{low} \ (D=14). \quad (1)$$

6. Формування відкритого ключа в складі відкритих даних

$$\{\rho, t_{high}\}. \quad (2)$$

7. Перетворення відкритого ключа у масив октетів та обчислення геш-значення від даних ключа (2).

$$tr = H(\rho IIthigh), \quad (3)$$

причому в якості функції гешування в оригіналі Dilithium використовується геш-функція shake256.

8. Формування особистого ключа у складі відповідних відкритих та особистих ключових даних:

$$\{\rho_1, key, tr, S1, S2, t_{low}\}. \quad (4)$$

Генерація матриці поліномів A у Dilithium.

При генерації (обчислення) матриці поліномів A використовується відкритий ключ ρ (32 байти). При генерації матриці A (коефіцієнтів поліномів матриці A) попередньо необхідно сформувати з використанням функції гешування чи симетричного шифру псевдовипадкову послідовність з довжиною не менше K*L*256*32 бітів. Але, оскільки кожний коефіцієнт визначається значенням параметру QBITS=23 (значення $\log_2 q$) і займає 23 біта, то число необхідних бітів можна скоротити до K*L*256*23 бітів.

Для формування коефіцієнтів поліномів в проекті стандарту ЦП Dilithium пропонується використовувати функцію гешування Keccak чи алгоритм симетричного шифрування AES. У випадку використання функції гешування послідовно гешуються спочатку відкритий ключ ρ+0, а потім для вироблення необхідного числа бітів коефіцієнтів матриці A значення ρ доповнюється кожен раз як ρ+1, ρ+2, …, ρ+32, ρ+33, … тощо для необхідного числа бітів.

Таким чином, наприклад, якщо K=5, а L=4, то необхідно згенерувати, як мінімум, 5*4*256*23=117760 бітів. У загальному випадку для побудування матриці A необхідно згенерувати 23*32*K*L байтів. Процес завершується після отримання усіх елементів матриці A.

Матриця A використовується в операції множення. Для використання швидкого NTT алгоритму множення поліномів попередньо потрібно перетворити елементи даних, для яких виконується множення, в NTT форму. Але, оскільки матриця A випадкова, то для скорочень обчислень можна вважати, що вона вже і є матрицею в NTT формі.

Генерація секретних ключів $S_1$ та $S_2$ ЦП Dilithium

Секретні ключі $S_1$[L][256] та $S_2$[K][256] генеруються у вигляді множини відповідно K та L поліномів з використанням $\rho_1$. Значення коефіцієнтів поліномів, які утворюють вектори $S_1$ та $S_2$, не можуть перевищувати значення ±η. З урахуванням усіх параметрів значення η не перевищує 7, тобто потребує 4 або 3 байти (тобто η=3). В якості початкового випадкового елементу при генерації секретних векторів $S_1$ та $S_2$ використовується секретний початковий ключ $\rho_1$ (32 байти). При необхідності ентропія $\rho_1$ може бути змінена, як зменшена так і збільшена. Безпосередньо генерування необхідного числа псевдовипадкових бітів – коефіцієнтів векторів поліномів $S_1$ та $S_2$ генерується аналогічно як при обчисленні матриці A.

Важливо відмітити, що після генерації усіх поліномів для векторів $S_1$ та $S_2$ виконується їх перетворення в NTT форму.

Формування старшої та молодшої частин коефіцієнтів поліному $t_{high}$ та $t_{low}$.

При заданому значені D t[i][j] коефіцієнти записується в формі $t=t_{high}*2D+t_{low}$. При чому, $t_{low}$ – це значення $t\%2^D$ відносно середини, тобто відхилення від відповідного значення D/2, яке може бути як позитивним, так і негативним.

Але безпосередньо секретними компонентами є $\{\rho_1, key, S_1, S_2, t_{low}\}$. Значення ρ є відкритим ключем і використовується для генерування матриці A. Значення tr (48 байтів) є геш-значення відкритого ключа – байтового рядка для відкритого ключа (використовується shake256) для формування геш-значення заданої довжини.

Значення $\rho_1$ використовується як секретне при генеруванні секретних ключів $S_1$ та $S_2$. Значення початкового ключа key (32 байти) використовується при генеруванні секретного ключа маскування Y. Секретний ключ $S_1$ є вектор довжиною L поліномів. Секретний ключ $S_2$ є вектор довжиною K поліномів. Кожний коефіцієнт поліномів $S_1$ та $S_2$ ключів обмежується значеннями [–η, +η].

Якщо критичним параметром є час обчислення цифрового підпису, то замість $S_1$ можна записати NTT перетворення для $S_1$. Це збільшить довжину особистого ключа, тому що для запису одного елементу потрібно не 3 , а 23 біта.

$S_2$ – вектор довжиною K поліномів. Кожний компонент – поліном з елементами, значення яких обмежується [–η, +η]. Секретна складова ключа $t_{low}$ – обчислюється згідно формули $t=t_{high}*2^D+t_{low}$ (D=14), причому $t_{low}$ – це значення $t\%2^D$ відносно середини, тобто відхилення від D/2, яке може бути як позитивним, так і негативним, елемент потребує D бітів.

Склад відкритого ключа ЦП Dilithium визначається параметром ρ. Значення ρ є по суті початковою ентропією, що визначає матрицю поліномів B ЦП Dilithium це початкове значення

має 32 байти, кожний елемент є поліном з коефіцієнтами, що обмежені значенням $q$. Відкрите значення $t_{high}$ обчислюється згідно формули (1), тобто у вигляді $t=t_{high}*2^D+t_{low}$, причому D=14.

**Висновок**

1. На першому етапі досліджень були виявлені певні проблемні питання щодо ЦП CRYSTALS-Dilithium, стосовно яких були запропоновані певні удосконалення механізму ЦП.

2. Найважливішою зміною для 2-го раунду щодо ЦП CRYSTALS-Dilithium є пропозиція реалізувати як детерміновану, що закладена в версії 1-го раунду, так і рандомізовану версію.

3. Різниця між цими версіями механізму полягає у тому, що початкова ентропія удосконаленого ЕП, тобто у рандомізованій версії, може встановлюватись випадковим чином, тоді як у 1-й, детермінованій версії, вона визначалась тільки у вигляді геш-значення ключа та безпосередньо повідомлення M, що підписується.

4. Інші зміни зроблені в реалізації ЦП CRYSTALS-Dilithium. Вони стосуються оптимізації в алгоритмів підписання. Найважливіша оптимізація стосується обчислення умови відхилення на основі молодшої частини вектора w і вектора підказки. Так як механізм CRYSTALS-Dilithium розроблено з вимогою забезпечення довгострокової безпеки, то під час його розроблення розглядається модель порушника з найспритнішими можливостями зловмисника.

5. Зокрема, вважаються застосування ним квантових алгоритмів криптоаналізу. Хоча подібні квантові алгоритми в даний час можливо нереалістичні, але в механізмі Dilithium передбачено можливість використання криптоаналітичних систем.

6. В механізмі Crystals-Dilithium була зроблена спроба мінімізувати суму довжин відкритого ключа та ЕП. Внаслідок цього механізм Dilithium має, у порівнянні з іншими механізмами на алгебраїчних решітках, найменше поєднання розміру підпису та розмірів відкритих ключів, з однаковими рівнями безпеки (приблизно у 2,5 рази).

ЛІТЕРАТУРА

1. CRYSTALS-Dilithium Algorithm Specifcations and Supporting Documentation. / Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler and Damien Stehlé – Publishing Nov/30/2017 – P.2-30.
2. Post-Quantum Cryptography. Round 2 Submissions. URL: https://csrc.nist.gov/Projects/Post-Quantum- ryptography/Round-2-Submissions.
3. Vadim Lyubashevsky Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. ASIACRYPT, 2009. – pp. 598–616.
4. Lyubachevsky V., Ducas L., Kiltz E. [et all]. CRYSTALS–Dilithium. Techn. rep. NIST (2019). URL: https://crc.nist.gov./projects/post-quantum-cryptogtraphy/ round-2-submissions

**ПИСАРЕНКО Надія Володимирівна** – студентка, кафедра безпеки інформаційних систем і технологій, факультет комп'ютерних наук, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків, Харківська область, 61000; e-mail: pisarenko108@gmail.com; ORCID: 0000-0002-3122-9129.

Наукові інтереси:

*– дослідження алгоритмів цифрового підпису в постквантовому періоді.*

**ГОРБЕНКО Іван Дмитрович** – д.т.н, професор, професор кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків, Харківська область, 61000; e-mail: gorbenkoi@iit.kharkov.ua.

Наукові інтереси:

*– криптографія, криптоаналіз, постквантова криптографія, захист інформації.*

**УДК 004.458**

ПРАВОТОРОВА І.І., ЛАЗУРИК В.М.

# ВИБІР ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ДЛЯ РЕІНЖИНІРИНГУ ТЕСТОВОГО ПАКЕТУ TSHELL

**Вступ**

Педагогічне тестування − це форма вимірювання знань, заснована на застосуванні педагогічних тестів. В даний час більшість дослідників вітчизняної і зарубіжної тестології визнають дедалі більшу роль комп'ютерного тестування як форми педагогічної діагностики. При проведенні педагогічного комп'ютерного тестування важливим завданням є забезпечення кожного, хто проходить випробування, набором тестових завдань, відібраних на основі наукових прийомів, що використовуються для контролю знань, умінь і навичок тестуємого [1]. Педагогічний тест, що правильно сконструйований, містить тестові завдання, оцінка валідності яких була проведена в кілька етапів на різних вибірках випробовуваних [2].

Для комп'ютерного моніторингу якості освіти та визначення рейтингів студентів в навчальному процесі в 2001 році на комп'ютерному факультеті Харківського національного університету імені В.Н. Каразіна був розроблений програмний пакет TShell [3]. З 2004 року пакет не тільки використовується для автоматизації контролю знань студентів в ХНУ, але й постійно розвивається [4, 5].

В роботі приділено увагу деяким аспектам подальшого розвитку TShell. Щоб реалізувати можливості, необхідність застосування яких виникла за останні роки, треба провести реінжиніринг деяких робочих модулів пакету. При цьому треба враховувати те, що пакет постійно використовується на факультетах університету. Важливим моментом є коректний вибір програмних засобів для реалізації нових можливостей пакету. Роботу присвячено аналізу та вибору сучасного програмного забезпечення (ПЗ), яке б дозволило виконати реінжиніринг TShell без суперечностей з тим ПЗ, яке вже застосоване для розробки програмного пакета.

**Програмний пакет TShell**

Пакет TShell має багато рис, які вигідно відрізняють його від багатьох програмних засобів, що використовуються для комп'ютерного тестування. Перш за все передбачена можливість портативного його застосування, коли це програмне забезпечення запускається з флеш носія. В пакеті реалізовані такі алгоритми, що забезпечують варіативну функціональність. Передбачено побудову завдань всіх типів, що використовуються в педагогічних вимірюваннях. При проведенні тестування із загальної бази тестових завдань динамічно формуються індивідуальні набори тестових завдань для кожного, хто проходить випробування [4]. Для розширення бази тестових завдань використовуються методи симетричної та асиметричної синонімізації [5]. Окрім контролю знань TShell дозволяє проводити тренінг і має режими, що допомагають формувати практичні навички. Реалізована можливість перегляду як інтегральних результатів тестування, так і повного протоколу проходження тесту кожним студентом. Вище перелічені лише деякі можливості TShell. На протязі багатьох років проведено багато досліджень як результатів тестування, одержаних за допомогою пакету, так і можливостей самого TShell. Розроблені допоміжні пакети, що забезпечують статистичну обробку результатів тестування, написано багато статей, захищено багато дипломних робіт.

**Визначення реінжинірингу**

На даний момент технології безперервно розвиваються, удосконалюються з метою оптимізації, розширення можливостей і якості програмного забезпечення. Настає момент в життєвому циклі будь-якої повноцінної функціонуючої системи або програмного продукту, коли виникає необхідність переосмислення підходу, поліпшення програмних засобів (прикладом може служити навіть оновлення версії бібліотек). Якщо використовувати загальноприйняті терміни, то даний процес у багатьох сферах діяльності називають реінжиніринг. Цей процес вважається досить трудомістким тому, що під час реінжинірингу системи зачіпаються часто не конкретні

модулі, а здебільшого абсолютно всі модулі системи. Більш того, при реінжинірингу треба зберігати стиль і принципи кодування, а саме головне – це збереження поточних функціональних особливостей системи.

**Постановка проблеми**

При проведенні тестування за допомогою TShell в рамках модульного оцінювання в студентських групах було виявлене вузьке місце пакету. Індивідуальна тестова батарея завдань, над якими буде працювати кожний з тих, хто тестується, складається з випадково обраних завдань необхідної кількості, з однаковою інтегральною складністю кожного такого набору. Цей досить складний алгоритм реалізовано на мові програмування JavaScript (JS), і формування тестової батареї відбувається на клієнтському комп'ютері. З одного боку це дає можливість швидкого формування тестового набору, але з іншого – файл з тестовими завданнями знаходиться на клієнті. Кожен браузер має інструменти розробника, які включають в себе можливість подивитися ресурси на вкладці Sources. Ресурси включають в себе те, що було завантажено разом з поточною сторінкою − це HTML, JavaScript. На теперішній час студенти, що становлять цільову аудиторію TShell, мають комп'ютерну грамотність, що є метою їх навчання. Але у випадку тестування за допомогою TShell їм достатньо знати про можливості браузера і скористатися цим, щоб проглянути файл тестових завдань, як ресурс клієнта. При цьому вже не йде мова про перевірку знань з дисципліни, кращім становиться той, хто потихеньку та досить швидко зможе переглянути JS файл.

Тому виникла необхідність реінжинірингу тієї частини пакету, що відповідає за формування індивідуальної тестової батареї, а саме перенесення логіки її створення на сервер. При цьому програмні засоби для розробки нових модулів повинні бути такими, що відповідають принципам конструювання TShell, нові програмні рішення повинні не зашкодити функціональним особливостям пакету, і все це повинно відбуватися в реальному часі. Важливою задачею стає необхідність розгляду можливих альтернативних засобів, які підходять для реалізації серверної частини, вибору конкретних з них та обґрунтуванню зробленого вибору.

**Особливості програмної реалізації TShell**

Кодова база програмного пакета TShell складається з HTML + JavaScript модулів, в яких реалізовано основний функціонал і алгоритми пакету, та PHP скриптів, які забезпечують взаємодію з MySQL базою даних (БД). В БД зберігається вся інформація, що стосується проходження тесту кожним студентом. PHP модулі не складні, логіка роботи з БД та запити до неї прозорі, немає багатьох функцій, що вжиті в коді модулів. На відміну від PHP, код JS модулів складний, з багатьма вкладеними функціями, і хоча кожна з них непогано задокументована, код не простий для розбору та розуміння. Тому, на сам перед, виникає бажання не чіпати, по можливості, JS реалізацію, а якимось чином перенести її на серверну частину. Також скриптова мова зручна у випадках, якщо потрібно забезпечити програмування без ризику дестабілізувати систему. Так як, на відміну від плагінів, скрипти інтерпретуються, а не компілюються, неправильно написаний скрипт виведе діагностичне повідомлення, а не приведе до системного краху [6].

**Пошук альтернативних програмних засобів для реалізації**

Під час аналізу програмного забезпечення, що може бути застосоване для реінжинірингу TShell, розглядалися PHP і Node.js. Проблема вибору між PHP і Node.js не нова. З нею досить часто стикаються backend-програмісти. Раніше мова JavaScript (та, що лежить в основі Node.js) не перетиналася з PHP. JavaScript використовували для створення інтерфейсних програм, а PHP для розробки серверних додатків. Працюючи разом, ці дві мови створювали дивовижні сайти. Але з часом JS був представлений абсолютно новим ПЗ Node.js і почав входити в сферу розробки на стороні сервера, чим і відвернув увагу backend-програмістів від традиційного PHP. Якщо звести усі переваги застосування PHP [7], то можна сказати, що він:

- забезпечує просте підключенням до SQL бази даних;
- не має обмежень для розміщення хостингу;
- не потрібен компілятор, або будь-який з файлів JAR;

- легко встановити і використовувати на стороні сервера;
- працює на движку Zend;
- PHP з початку створення синхронний, проте існують API, які надають можливості асинхронного виконання;
- підтримується в популярних системах управління контентом (Drupal, Joomla, WordPress);
- додаткові бібліотеки − пакетний менеджер Composer.
- Стосовно Node.js підсумок особливостей застосування [8] може бути таким:
- спосіб та швидкість роботи з базами даних не відрізняється від PHP;
- складний, розгортання програм та фреймворків вимагає більш складної підготовки інфраструктури та сервера;
- код Node.js виконується швидко, зменшує навантаження на сервер, отримуючи доступ до зворотних викликів, що витрачають менше часу на роботу з кількома різними потоками;
- це середовище виконання для JavaScript на стороні сервера, створене на основі движка Google V8 JavaScript (який вміє компілювати JavaScript код в машинний код);
- асинхронний (це означає, що механізм JavaScript проходить весь код за один раз і не чекає повернення функції);
- використовує керовану подіями модель, що не блокує потік вводу / виводу;
- можливо масштабувати на системах з кількома ядрами так, щоб задіяти всі можливості ядра;
- вимагає багато часу на кодування і компіляцію, але кінцевий результат більш оптимізований.

PHP дуже популярний для створення блогів та веб-додатків електронної комерції, а Node.js ефективно служить інструментом для створення масштабованих динамічних рішень, які стосуються чисельних операцій вводу / виводу. З точки зору доступу до різних бібліотек PHP та Node.js знаходяться на абсолютно одному рівні [9].

На даний момент, щоб запустити для роботи програмний пакет TShell використовується кросплатформна збірка веб-серверу XAMPP [10]. З використанням PHP буде можливість залишити цей веб-сервер, бо PHP працює здебільшого на Apache сервері, але з використанням Node.js з'являється необхідність змінити його. Адже Node.js працює на власному середовищі виконання. Також разом з цим постає проблема разом з Node.js мати налаштованим також MySQL Server, щоб сервер мав змогу комунікації з ним. У випадку з XAMPP, MySQL входить в його збірку. Проте це лише збірка програмних інструментів. Якщо йде мова про складність встановлення кожного з компонентів, то є можливість написання скрипту під різні операційні системи задля їх встановлення. У випадку з сімейством Linux написання скрипту не є трудомістким завданням завдяки відкритим репозиторіям.

TShell – пакет реального часу. Тож при виборі стеку технологій для реалізації реінжинірингу потрібно враховувати невпинний потік вводу – виводу, та необхідність забезпечення безперебійної роботи сервера.

**Висновки**

Проведений аналіз можливих альтернативних програмних засобів для реалізації нового функціоналу TShell показує, що розглянуте ПЗ цілком підходить для розробки і досягнення поставленої мети. Але, зважаючи на те, що складний код формування індивідуальної тестової батареї реалізований на JS, кращим для реінжинірингу вважається застосування Node.js. В такому випадку можливо перенести на серверну частину JS код багатьох функцій, не чіпаючи та не переписуючи їх. Важливо також враховувати досвід розробника, який реалізує нові можливості пакету, бо від цього залежить час виконання роботи. За цим критерієм застосування Node.js теж вбачається кращім.

ЛІТЕРАТУРА

1. Аванесов В.С. Форма тестовых заданий : учеб. пособие / Аванесов В.С. – М.: Центр тестирования, 2005. 156 с.
2. Челышкова М.Б. Теория и практика конструирования педагогических тестов: Учебное пособие. М.: Логос, 2002. с. 268 – 278
3. Васильєва Л.В., Лазурик В.Т., Лазурик В.М., Рудичев Д.В. Комплексна комп'ютерна система моніторінгу та визначення рейтингів у навчальному процесі // П'ята міжнарожна науково-практична конференція. Херсон: Херсонський держ. університет, 2009. С. 36-37. 7.
4. Лазурик В.В., Лазурик В.М. Построение моделей педагогических измерений в программе компьютерного тестирования Tshell. Вестник Херсонского Технического Университета. №2(45). 2012. - с.185-189
5. Лазурик В.В., Лазурик В.М., Ю.А. Шаптала Симетрична та асиметрична синонімізація завдань при КОМП'ЮТЕРНОМУ ТЕСТУВАННІ З ВИКОРИСТАННЯМ ПАКЕТУ Tshell. Вісник Херсонського національного технічного університету. - 2015. - № 3. - С. 262-265.
6. Сценарный язык. [Электронный ресурс] Режим доступа: https://ru.wikipedia.org/wiki/Сценарный_язык (Last accessed: 01.03.2020).
7. Руководство по PHP. [Электронный ресурс] Режим доступа: https://www.php.net/manual/ru/index.php
8. Jorge Ramon. Everything you need to know about Node.js. URL: https://dev.to/jorge_rockr/everything-you-need-to-know-about-node-js-lnc#theproblemwithcpuintensivetasks (Last accessed: 01.03.2020).
9. **Sophia Martin**. Node.js vs PHP: Which is better for web development? URL: https://hackernoon.com/nodejs-vs-php-which-is-better-for-your-web-development-he7oa24wp (Last accessed: 07.03.2020).
10. XAMPP. [Электронный ресурс] Режим доступа: https://ru.wikipedia.org/wiki/XAMPP.

**ПРАВОТОРОВА Ірина Ігорівна** – студентка факультету комп'ютерних наук кафедри штучного інтелекту та програмного забезпечення Харківського національного університету імені В.Н. Каразіна, площа Свободи, 4, Харків-22, Україна, 61022; e-mail: pravotorovairina12@gmail.com; ORCID: 0000-0003-2432-8732

Наукові інтереси:

- *мікросервісна архітектура; проектування інформаційних систем.*

**ЛАЗУРИК Валентина Михайлівна** – старший викладач кафедри штучного інтелекту та програмного забезпечення факультету комп'ютерних наук Харківського національного університету імені В.Н. Каразіна, площа Свободи, 4, Харків-22, Україна, 61022; e-mail: lazurik@hotmail.com; ORCID: 0000-0002-3340-9780.

Наукові інтереси:

- *розробка програмного забезпечення області радіаційних технологій, комп'ютерне моделювання педагогічних вимірювань.*

**УДК 004.415.5**

ПУДОВКІНА Л.Ф.

# ЗАСТОСУВАННЯ ЕМПІРИЧНИХ ТА АНАЛІТИЧНИХ МОДЕЛЕЙ ДЛЯ ОЦІНЮВАННЯ ЯКОСТІ ПРОГРАМ

Якість – сукупність властивостей продукції, що обумовлюють її придатність задовольняти певні потреби відповідно до її призначення . Якість програмного забезпечення[1] завжди обов'язково взаємопов'язана з надійністю його функціонування.

Надійність функціонування програмного забезпечення – це ймовірність безвідмовної роботи програмного забезпечення протягом певного періоду часу в заданому середовищі[2]. Висока складність програмного забезпечення є основним фактором, що впливає на проблеми надійності функціонування програмного забезпечення. Ця проблема має два аспекти: забезпечення і оцінка (вимір) надійності [2].

Моделі надійності функціонування програмних засобів підрозділяються на аналітичні та емпіричні [3]. Аналітичні моделі дають можливість розрахувати кількісні показники надійності, ґрунтуючись на даних про поведінку додатка в процесі тестування. Емпіричні моделі базуються на аналізі структурних особливостей програм.

Аналітичні моделі представлені двома групами: динамічні і статичні. У динамічних моделях поведінку ПЗ (поява відмов) розглядається в часі. У статичних моделях поява відмов не пов'язують з часом, а враховують залежність кількості помилок або від числа тестових прогонів (моделі по області помилок), або від характеристики вхідних даних (моделі по області даних).

Деякі емпіричні моделі обчислюють складність програми, спираючись на властивості її тексту, наприклад, метрика Джілбі [4] визначає логічну складність програми як насиченість її операторами типу if-then-else або операторами циклів. Також може враховуватися співвідношення кількості зв'язків між модулями до загальної кількості модулів. Модель, заснована на метриках Чепіна [7], оцінює надійність окремо взятого модуля за результатами аналізу характеру використання змінних.

Емпірична модель – різновид моделей, основу якої складають результати аналізу деякого об'єму даних (інформації), отриманих в результаті експерименту, вимірювань або прогнозування.

Емпіричні моделі базуються на аналізі структурних особливостей програмних засобів. Емпіричні моделі часто не дають кінцевих результатів показників надійності функціонування, проте їх використання на етапі проектування ПЗ корисно для прогнозування потрібних ресурсів тестування, дозволяє виявляти взаємозв'язок між складністю програмних засобів і його надійністю, уточнення планових термінів завершення проекту.

Ці моделі можна використовувати на етапі проектування програмних засобів, коли здійснена розбивка на модулі і відома його структура. Найбільш проста емпірична модель пов'язує число помилок в програмному забезпеченні з його об'ємом [7].

Емпіричні моделі якості створюються у формі «чорного ящика», оскільки ніякі знання про саму систему не включені в моделі. Емпіричні моделі часто не дають кінцевих результатів показників надійності, однак їх використання на етапі проектування ПЗ корисно для прогнозування необхідних ресурсів тестування, уточнення планових термінів завершення проекту . При цьому верифікація та тестування вимагає додаткових витрат на складання імітаційної моделі, і приблизний характер одержуваних показників.

При проведенні тестування відома структура додатка, що імітує дії основної, але ніхто не знає конкретний шлях, який буде виконуватися при введенні певного тестового входу. Ці умови повинні цілком відповідати реальним умовам тестування великих програм.

Емпіричну і аналітичну моделі можна використовувати для прогнозування оптимальних наборів даних для тестів продуктів будь-якого призначення

Більшість моделей надійності функціонування програмного забезпечення визначають надійність на початкових стадіях життєвого циклу. Тому для визначення надійності функціонування програмного забезпечення на всіх стадіях його життєвого циклу доцільно застосовувати, як мінімум, дві моделі надійності функціонування програмного забезпечення.

Емпіричні моделі доцільно використовувати на етапах проектування і збірки, так як вони дозволяють приймати оптимальні проектні рішення.

Динамічні моделі надійності ПЗ [5] можливо використовувати на етапі складання і налагодження ПЗ, так як вони дають можливість вносити виправлення в програму по мірі тестування. А статистичні моделі [6] зручно використовувати після налагодження програми, а також після впровадження на етапі супроводу, так як в таких моделях тестування ґрунтується на введенні тестових наборів і зборі статистичної інформації про проведені тести. Для визначення надійності функціонування програмного забезпечення на завершальних стадіях найбільш ефективно застосовувати моделі надійності з системно-незалежним аргументом.

Перш ніж забезпечувати надійність, слід навчитися її вимірювати. Але для цього потрібно мати практично прийнятну одиницю виміру надійності функціонування ПЗ і модель її розрахунку. Доведено доцільність використання моделі Джелінскі-Моранді [3] для кількісної оцінки показників надійності функціонування програмного забезпечення.

Якість програмного забезпечення залежить від складності, тому потрібно завжди оцінювати складність програмних продуктів. Інтегральна система оцінювання не тільки складності, але і якості програм в цілому є система метрик, запропонована Холстедом [7]. програмування Будь-яка метрика – це лише показник, який сильно залежить від мови та стилю.

Статичні складові складності, характерні для етапів розробки програмного продукту і визначають їх трудомісткість. Динамічні складові складності програмного продукту проявляються в процесі його виконання. При вивченні кожної складної програмної системи необхідно, враховує особливості функціонування і своєрідність конкретної системи, тому треба виконувати оцінювання надійності програм, використовуючи емпіричну модель метрик Холстеду.

Перевага емпіричних моделей в тому, що вони не містять складних формул і обчислення по ним прості. Недоліком же є їх грубість і приблизність. Крім того, вони не відображають динаміки обчислювального процесу при експлуатації програм.

Висновок. Емпіричну і аналітичну моделі треба використовувати для прогнозування оптимальних необхідних наборів даних для тестування продуктів будь-якого призначення. Ці дані – масиви, що представляють собою еталони, з якими треба порівнювати дані, які отримані в процесі тестування.

Результатом використання емпіричних і аналітичних моделій є вивід (створення) нових формул, рівнянь закономірностей, кореляційних залежностей, що описують зв'язок між розглянутими метриками.

Підвищення надійності функцінування і якості програмного модуля здійснюється не тільки за рахунок аналізу прогнозування складності, а також через аналіз і прогнозування правильності обраних метрик тестування на всіх етапах життєвого циклу.

ЛІТЕРАТУРА

1. Липаев В. В. Качество программных средств / В. В. Липаев. – Москва: Синтег, 2002. – 216 с.
2. Романюк С. Г. Оценка надежности программного обеспечения / С. Г. Романюк. // Открытые системы. – 1994. – №4. – С. 24.
3. Модель Джелінскі-Моранді [Електронний ресурс] – Режим доступу до ресурсу: https://studopedia.su/10_153579_model-dzhelinskogo-morandi.html
4. Метрика Джілбі [Електронний ресурс] – Режим доступу до ресурсу: https://studopedia.org/11-18411.html
5. Динамічні моделі [Електронний ресурс] – Режим доступу до ресурсу: http://info-tehnologii.ru/kac_sr/Mod_nad/DiN/index.html
6. Статичні моделі [Електронний ресурс] – Режим доступу до ресурсу: http://info-tehnologii.ru/kac_sr/Mod_nad/Stat/index.html
7. Метрика Холстеда [Електронний ресурс] – Режим доступу до ресурсу: https://studfiles.net/preview/6196491/page:5/

**ПУДОВКІНА Лариса Федорівна** – к.т.н, доцент кафедри інженерії програмного забезпечення Національного аерокосмічного університету імені М. Є. Жуковського «Харківський авіаційний інститут» » Україна, 61070, Харків-70, вул. Чкалова, 17, e-mail l.pudovkina@khai.edu.

Наукові інтереси: – *моделі надійності функціонування ПЗ і моделі оцінки якості ПЗ.*

**УДК 004.85**

РУЖАНСЬКА А. В., ВАСИЛЬЄВА Л. В.

# ВИКОРИСТАННЯ ТЕХНОЛОГІЇ TRANSFER LEARNING ДЛЯ РОЗПІЗНАВАННЯ І КЛАСИФІКАЦІЇ ОБ'ЄКТІВ

Розв'язання задачі розпізнавання об'єктів і класифікації є одним з найважливіших застосувань машинного навчання. Рішенням такого завдання вважають виділення істотних ознак, що характеризують вихідні дані, із загальної маси несуттєвих даних та віднесення цих даних до певного класу.

Розпізнавання образів є однією з найбільш фундаментальних проблем теорії інтелектуальних систем, тому вона має величезне практичне значення. Проблема розпізнавання образів легко вирішується людським мозком, причому робиться це, як правило, підсвідомо. Спроби ж побудувати штучні системи розпізнавання не дуже переконливі через те, що часто неможливо адекватно визначити ознаки, на основі яких слід здійснювати розпізнавання. У завданнях, для яких такі ознаки вдається виділити, штучні системи розпізнавання набули значного поширення і широко використовуються.

У даній роботі розглянуто принцип створення комп'ютерної системи для розпізнавання і класифікації маркувань на пластмасі для подальшої його переробки з використанням методики transfer learning. Ідея створення такої системи виникла через те, що з кожним роком виробництво виробів з пластмаси збільшується з кожним роком, тому й відповідно збільшується і кількість пластмасових відходів. Пластмаси належать до матеріалів, які практично не розкладаються з часом, а при спалюванні виділяють вкрай токсичні речовини, які неможливо вивести з організму.

З кожним роком все більше досліджень говорять про те, що деякі види пластмаси можуть бути небезпечні. Одноразовий пластмасовий посуд зручний при використанні, але має безпосереднє згубний вплив на здоров'я споживачів. До пластмасового посуду необхідно ставитися вкрай обережно. Для правильного використання важливо навчитися розуміти коди переробки на пластмасах. Для розв'язання цієї проблеми виникла ідея створити систему, яка по фотографії (або відео) коду переробки на пластмасах зможе розпізнати тип пластмаси та пояснити інформацію про нього.

У даній роботі використовувались 7 типів пластмаси - PET (1), PEHD / HDPE (2), PVC / V (3), PELD / LDPE (4), PP (5), PS (6), OTHER / O (7 ). На пластмасовому пакуванні кожен з типів позначений у вигляді трикутника з відповідною цифрою всередині і буквеною абревіатурою під трикутником.

На початку роботи був створений набір зображень вручну та містив 900 зображень, які розділені на 8 класів (7 видів пластмаси + клас непластмаси). Весь наш набір розділено на train (той, що будемо навчати) і test (тестовий) набори даних. Далі були використані методи обробки зображень (зміна розміру зображень, збільшення яскравості і контрастності, згладжування) і аугментації (створення додаткових навчальних даних з наявних даних).

Для вирішення даного завдання було використано метод трансферного навчання (transfer learning). Трансферним навчанням називають техніку машинного навчання, при якій модель, навчена по одному завданню, перевизначається по іншому пов'язаному завданню. Основна ідея такого навчання так на тому, що навчивши нейронну мережу на великому наборі даних ми можемо застосувати отриману модель і для набору даних, який раніше ця модель ще не зустрічала. Саме тому методика називається transfer learning - передача процесу навчання з одного набору даних на інший.

Трансферне навчання дозволяє комбінувати наш набір зображень з попередньо навченою моделлю для створення власного класифікатора зображень. Такий метод дозволяє заощадити величезні ресурси часу, необхідні для навчання моделі глибокого навчання та отримання індивідуального класифікатора.

У даній роботі для найкращого результату була застосована згорткову нейромережу. Згорткова мережа на відміну від звичайних повнозв'язних нейронних мереж краща тим, що швидше навчається та має меншу кількість параметрів ваг, бо одне ядро ваг використовується

цілком для всього зображення, замість того, щоб для кожного пікселя вхідного зображення робити свої персональні вагові коефіцієнти. Це спонукає згорткову нейромережу при навчанні до узагальнення демонстрованої інформації, а не до попіксельного запам'ятовування кожної показаної картинки в міріадах вагових коефіцієнтів, як це робить перцептрон. Також згорткова нейромережа більш стійка до повороту і зсуву розпізнавання зображення.

Для навчання використано згорткову нейромережу ResNet-50 (скорочена назва для Residual Network, дослівно - «залишкова мережа»). ResNet-50 є згортковою нейронною мережею, яка навчена більш ніж на мільйоні зображень бази даних ImageNet. Мережа складається з 50 шарів та може класифікувати зображення більш ніж 1 000 категорій об'єктів. Задача нейронної мережі ResNet-50 - дати нам якісний класифікатор коду переробки для визначення типу пластмаси.

Також використано бібліотеку Fast.ai для оптимізації розробки класифікатора, а саме модуль *vision.learner*, який визначав метод *cnn_learner* для швидкого отримання моделі, придатної для трансферного навчання.

У роботі було застосовлено два методи використання попередньо навчених мереж: виділення ознак (feature extraction) і донавчання (fine-tuning). Перший метод полягав в використанні уявлень, вивчених попередньої мережею, для виділення ознак з нових зразків, які потім пропускаються через новий класифікатор, який навчається з нуля.

Для того, щоб застосувати трансферне навчання для класифікації зображень, необхідно було змінити останній шар нашої згорткової нейронної мережі так, щоб він містив таку кількість виходів, яка б відповідала кількості класів в новому наборі. Також заздалегідь необхідно було переконатися, що під час процесу тренування наша навчена модель не змінилася. Для цього було відключено змінні попередньо навченої моделі - тобто заборонено алгоритму, який оновлює значення при прямому і зворотному поширенні, їх міняти. Цей процес має назву "заморожування моделі" (freezing the model). "Заморожуючи" параметри попередньо навченої моделі, навчався тільки останній шар нейронної мережі класифікації, при чому значення змінних попередньо навченої моделі залишилися незмінними.

Ще одна незаперечна перевага попередньо навчених моделей полягала в тому, що було скорочено час навчання моделі завдяки тренуванню тільки останнього шару зі значно меншою кількістю змінних, а не всієї моделі.

Якщо не "заморозити" змінні попередньо навченої моделі, то в процесі навчання на новому наборі даних значення змінних будуть змінюватися. Це відбувається тому, що значення змінних на останньому шарі класифікації будуть заповнені випадковими значеннями. Через випадкові значення на останньому шарі модель буде допускати великі помилки в класифікації, що може спричинити серйозні зміни вихідних ваг в попередньо навченій моделі, а вкрай небажано. Саме з цієї причини було використано метод "заморожування" і відключення необхідності навчання попередньо навченої моделі.

Після навчання нейронної мережі на валідаційній вибірці було отримано результати прогнозів, де точність моделі становила 89%.

Інший широко використовуваний метод трансферного навчання називається донавчанням (fine-tuning). Донавчання полягає в розморожуванні кількох верхніх шарів замороженої моделі, яка використовувалася для виділення ознак, й спільному навчанні знову доданої частини моделі (в цьому випадку класифікатора) та її верхніх шарів. Цей метод має саме таку назву, оскільки трохи корегує найбільш абстрактні уявлення в повторно використаній моделі, щоб зробити їх більш актуальними для даного завдання. Після використання донавчання нейронної мережі, точність збільшилася до 90,4%, виходячи з результатів.

Отже, завдяки проведеній роботі було розглянуто принцип розпізнавання і класифікації маркування на об'єктах з використанням технології передачі навчання (transfer learning) для створення власної згорткової нейронної мережі на основі ResNet-50. Також можна зробити висновок, що згорткові нейронні мережі є найкращим варіантом моделей машинного навчання для задач розпізнавання і класифікації образів. Внаслідок чого ми змогли навчити нейронну мережу з нуля на невеликому наборі даних і отримати гарний результат. Також для виділення ознак використовувався метод донавчання, що адаптував до нової задачі деякі з представлень, які були раніше отримані моделлю. Донавчання нейронної мережі ще більше підвищило якість та точність моделі.

ЛІТЕРАТУРА

1. Ф. Шолле. Глубокое обучение на Python: Пер. с англ. – СПб.: Питер, 2018, с. 148 - 190.
2. Богатырева А.А., Виноградова А.Р., Тихомирова С.А. Исследование способности к Transfer learning сверточных нейронных сетей, обученных на ImageNet // Международный журнал прикладных и фундаментальных исследований. – 2019. – № 7. – с. 106 - 111; URL: http://www.applied-research.ru/ru/article/view?id=12808 (дата звернення: 10.03.2020).
3. Л. Шапиро, Дж. Стокман. Компьютерное зрение = Computer Vision. — М. : Бином. Лаборатория знаний, 2006. — 752 с.
4. Горелик А. Л., Скрипкин В. А. Методы распознавания. — 4-е изд. — М.: Высшая школа, 1984, 2004. — 262 с.
5. Форсайт Дэвид А., Понс Джин. Компьютерное зрение. Современный подход = Computer Vision: A Modern Approach. — М.: Вильямс, 2004. — 928 с.

**РУЖАНСЬКА Анастасія В’ячеславівна** – студентка кафедри штучного інтелекту та програмного забезпечення факультету комп’ютерних наук Харківського національного університету імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: asynasty1999@gmail.com; ORCID: 0000-0002-2683-6768.

Наукові інтереси:

- *методи машинного навчання;*
- *системи комп'ютерного зору;*
- *штучний інтелект.*

**ВАСИЛЬЄВА Лариса Валентинівна** – к.б.н., доцент кафедри електроніки та управляючих систем; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: lvvasilieva@karazin.ua.

Наукові інтереси:

- *штучний інтелект.*

**УДК [004.7:004.75]:004.8**

СЕМЕНЮК Б.С.

# КОМП'ЮТЕРНА МОДЕЛЬ РОЗПОДІЛЕНОГО ПРОЦЕСУ НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ НА БАЗІ ТЕХНОЛОГІЇ TENSORFLOW

**Застосування розподілених систем для навчання штучних нейронних мереж**

У сучасному цифровому світі використання нейронних мереж знаходить застосування в різних сферах діяльності. Для вирішення деяких завдань проектуються штучні нейронні мережі, що досягають величезних розмірів — кількість нейронів в них може досягати мільйона, а іноді і більше. Розподілені обчислення відкрили нові шляхи застосункам, що вимагають великих обчислювальних потужностей. Необхідність використання розподілених обчислень для вирішення завдання росте разом із зростанням обсягів даних. Але з іншого боку застосування розподілених обчислень для навчання штучних нейронних мереж є відносно новим завданням і вимагає додаткових досліджень.

**Аналіз штучних нейронних мереж**

З точки зору реалізації штучна нейронна мережа є система з'єднаних і взаємодіючих між собою простих процесорів (штучних нейронів). Такі процесори зазвичай досить прості, особливо в порівнянні з процесорами, які використовуються в персональних комп'ютерах. Кожен процесор подібної мережі має справу тільки з сигналами, які він періодично отримує, і сигналами, які він посилає іншим процесорам.

Типовий приклад мережі прямого поширення показаний на рис. 1. Нейрони регулярним чином організовані в шари. Вхідний шар — розподільний - служить просто для введення значень вхідних змінних. Кожен з прихованих і вихідних нейронів з'єднаний з усіма елементами попереднього шару.

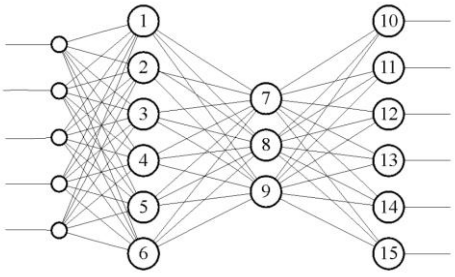



*Рис. 1 Приклад мережі прямого поширення*

Ідея розпаралелювання обчислень базується на тому, що більшість завдань може бути розділена на набір менших завдань, що працюють паралельно (одночасно) та можуть бути вирішені незалежно один від одного. Окремим випадком паралельних обчислень є розподілені обчислення, які представляють собою спосіб вирішення трудомістких обчислювальних завдань з використанням двох і більше комп'ютерів, об'єднаних в мережу.

Спільною особливістю масштабованих систем є розподіленість процесів обробки і розподіленість даних. З одного боку у розподілі обчислень і даних закладена можливість підвищення продуктивності таких систем. З іншого боку інтенсивний обмін даними суттєво завантажує комунікаційне середовище. Це обмежує можливості масштабування систем. Отже, ефективність виконання програм в масштабованих архітектурах значною мірою залежить від організації взаємодії розподілених процесів. Для організації розподілених обчислень необхідно, щоб завдання, яке вирішується було сегментовано — розділено на підзадачі, які можуть обчислюватися паралельно.

Використання дрібнозернистої паралельності нейронних мереж для проведення розподілених обчислень вважається невиправданим, оскільки число зв'язків і вузлів мережі дуже велике, а обсяг обчислень в кожному з вузлів мережі — незначний. Зокрема, для мережі,

зображеної на рис. 1, що нараховує всього 15 вузлів, число зв'язків виявляється рівним 66 і стрімко зростає зі збільшенням розмірів мережі. Для обчислення мережі, що містить всього 1000 нейронів, буде потрібно обчислювальна мережа, яка містить 1000 вузлів, і, в залежності від структури мережі, близько 250000 зв'язків. При цьому ефективність роботи такої мережі буде значно нижче, ніж при обчисленні штучної нейронної мережі на одному з вузлів розподіленої обчислювальної системи, оскільки буде потрібно чимало часу для передачі даних у мережі, очікування цих даних і синхронізацію обчислень.

### Типи паралелізму в розподіленому глибокому навчанні

Існує два основні методи реалізації, які можна використовувати для поширення навчання моделі глибокого навчання: паралелізм моделі або паралелізм даних. Іноді єдиний підхід призводить до підвищення продуктивності застосуників, в той час як в інших випадках поєднання двох підходів знижує продуктивність.

У паралелізмі даних використовується одна і та ж модель для кожного пристрою, але навчається модель, використовуючи різні навчальні зразки, як зображено на рис. 2. Це контрастує з паралелізмом моделі, який використовує одні й ті ж дані для кожного пристрою, але розділяє модель між пристроями, як показано на рис. 3. Кожен пристрій буде незалежно обчислювати помилки між своїми прогнозами для навчальних вибірок та позначеними вихідними даними. Оскільки кожен пристрій навчається на різних вибірках, він обчислює різні зміни в моделі («градієнти»). Однак алгоритм залежить від використання об'єднаних результатів всієї обробки для кожної нової ітерації, так якби алгоритм працював на одному процесорі. Тому кожен пристрій має відправляти всі свої зміни всім моделям на всіх інших пристроях. В такому випадку використовується модель з серверами параметрів та обчислень.


*Рис. 2 Модель розподіленого навчання - паралелізм даних*

Застосунки TensorFlow для розподіленого навчання використовують архітектуру кластера, який складається з декількох серверів параметрів (parameter server, PS) та багатьох серверів обчислень (worker). Оскільки сервера обчислень підраховують градієнти під час навчання, вони зазвичай поміщаються в графічний процесор. Сервери параметрів повинні збирати градієнти і обробляти широкомовні повідомлення, тому всі операції зазвичай проводяться на центральному процесорі, тому такий сервер не потребує додаткового графічного процесору.

Одним з недоліків Distributed TensorFlow, що є частиною ядра TensorFlow, є те, що існує необхідність явно керувати запуском і зупинкою серверів. Це означає відстеження IP-адресів та портів всіх серверів TensorFlow у вашій програмі, а також постійний контроль над запуском і зупинкою цих серверів. Як правило, це призводить до великої кількості операторів switch в коді, щоб визначити, які інструкції повинні виконуватися на поточному сервері.

Такий спосіб контролю схильний до помилок і тому недоцільно створювати файл специфікації кластеру з використанням кінцевих точок хоста (IP-адреса і номеру порту). В такому випадку краще використовувати диспетчер кластерів, такий як YARN, Kubernetes або Mesos, щоб зменшити складність налаштування і запуску додатків TensorFlow.

Використання подібних систем контролю та управління кластерами має велику перевагу — горизонтальне масштабування. Горизонтальне масштабування вирішує проблему додаткового навантаження, але не шляхом збільшення існуючих ресурсів (CPU та RAM) на

обчислювальних вузлах, а саме збільшенням кількості обчислювальних вузлів. Розширення кластеру за допомогою нових обчислювальних одиниць відбувається без простоїв в роботі існуючих серверів. При цьому кількість нових серверів, які можна додати до загальної системи, нічим не обмежена.

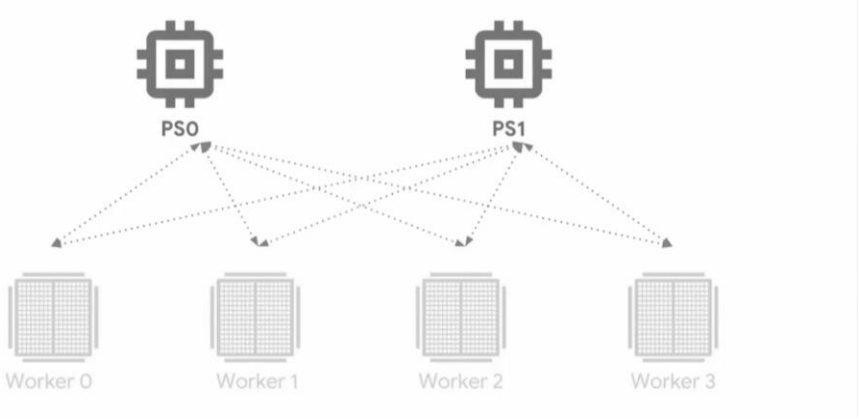



*Рис. 3 Модель розподіленого навчання - паралелізм моделі*

## Висновок

В ході роботи були проаналізовані можливості використання кластерних паралельних обчислювальних систем для прискорення обчислення штучних нейронних мереж, позитивні і негативні сторони для розподіленого підходу.

В результаті аналізу було обрано диспетчер кластерів Kubernetes для запуску додатків Tensorflow. Kubernetes є найбільш переконливим вибором між доступних диспетчерів саме завдяки ефективному інструменту управління і обмеження ресурсів, в тому числі і ресурсів GPU, між завданнями, що виконуються на кластері.

Стратегії Tensorflow MirroredStrategy та MultiWorkerMirroredStrategy є найкращими варіантами розподіленого навчання, тому що вони реалізують модель паралелізму даних і саме тому підходять для будь-якого розміру нейронних мереж. Вибір цих стратегій дозволяє масштабувати кластер як вертикально, у випадку з MirroredStrategy, так і горизонтально, у випадку з MultiWorkerMirroredStrategy.

Можливість автоматичного розподілу нейронної мережі між вузлами обчислювальної системи є безперечною перевагою систем подібного типу, оскільки, на відміну від більшості інших способів організації обчислень, не вимагає спеціальних знань від користувача.

ЛІТЕРАТУРА

1. Герасименко М. С. Вычисление искусственных нейронных сетей на вычислительных кластерах или в локальных вычислительных сетях. *Вестник ВГУ, Сер: Системный анализ и информационные технологии.* 2010. № 1. С. 120-125
2. Нгуен Занг, Краснощеков А. А. Распределенная платформа для параллельного обучения искусственных нейронных сетей DisANN. *Программные продукты и системы.* 2013. № 3. С. 99-103
3. Distributed training. [Електронний ресурс]. Tensorflow: Web site. URL: https://www.tensorflow.org/guide/distributed_training/ (дата звернення: 15.02.2020).
4. Distributed TensorFlow. [Електронний ресурс]. Oreilly: Web site. URL: https://www.oreilly.com/ideas/distributed-tensorflow (дата звернення: 12.02.2020).

**СЕМЕНЮК Борис Сергійович** – бакалавр, студент кафедри теоретичної та прикладної системотехніки; Харківський національний університет імені В.Н. Каразіна, площа Свободи, 4, Харків-22, Україна, 61022; e-mail: semenyuk.boris@gmail.com; ORCID: 0000-0002-6892-1872.

Наукові інтереси:

*– розподілені комп'ютерні системи.*

**УДК 504.43**

СЄРІКОВА О.М., СТРЕЛЬНІКОВА О.О.

# ТРИВИМІРНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ ЗМІНИ РІВНЯ ҐРУНТОВИХ ВОД МІСЬКИХ ТЕРИТОРІЙ

Для сталого розвитку міст, захисту забудови від небезпечного підйому рівня ґрунтових вод (РГВ) та підтоплення, необхідно правильно оцінювати існуючі гідрогеологічні умови та з необхідною точністю їх прогнозувати [1]. Основним завданням є визначення характеру зміни рівня ґрунтових вод під дією зовнішніх факторів. У зв'язку з тим, що проводимість в анізатропних ґрунтах в різних напрямках різна, якщо будова пористого середовища така, що має більш високу проводимість в одному напрямку ніж в інших, існує необхідність враховувати зміни РГВ у тривимірному моделюванні [1,2,3]. Для прогнозування зміни рівня ґрунтових вод розроблено математичну модель, що враховує інфільтрацію атмосферних вод, додаткове живлення в ґрунтові води, транспірацію, випаровування, евапотранспірацію і водовідбір з підземних вод.

При цьому вважається, що зміна РГВ має усталений характер, про що свідчать дані багаторічних досліджень [4], в 3-х режимних водопунктах м. Харків. На відміну від досліджень [4,5] в цій роботі розглянуто задачу прогнозування зміни РГВ в тривимірному формулюванні.

Розглянемо рівняння фільтраційного напору у вигляді

$$\frac{\partial^2 h}{\partial z^2} + \gamma_1^2 \frac{\partial^2 h}{\partial x^2} + \gamma_2^2 \frac{\partial^2 h}{\partial y^2} = 0, \qquad (1)$$

де $h$ – рівень ґрунтових вод, $x, y, z$ – координати, показані на рис. 1, $\gamma_1$ $\gamma_2$– коефіцієнти анізотропії.

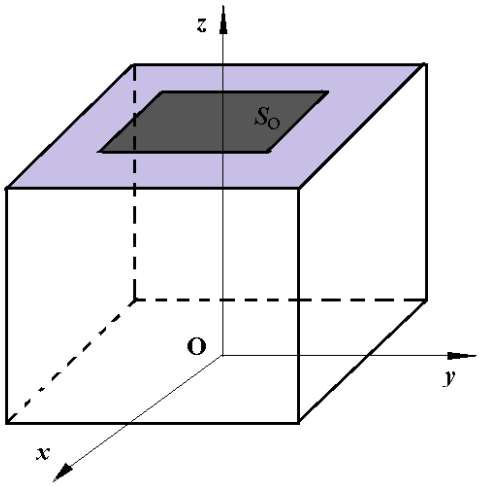



*Рис. 1 Розрахункова область для визначення РГВ*

Сформулюємо крайові умови для рівняння (1), які враховують наявність штучних покриттів, інфільтрацію, випаровування та транспірацію, а також ефект евапотранспірації. Ці умови ставимо відносно значень невідомої функції, або її нормальної похідної на межах розрахункової області. Припустимо, що розрахункова область є прямокутним паралелепіпедом. Нижня та верхні грані цього паралелепіпеда є прямокутниками $S$ зі сторонами $[2a,2b]$. Висоту паралелепіпеда позначимо як $L$. Нехай область $S_0$ є квадратом зі сторонами $[2l,2l]$, який розташовано в центрі верхньої грані.

Припустимо, що на ділянці $S_1 = S \setminus S_0$ відбувається вплив природних і техногенних факторів на зміну рівня ґрунтових вод; в той час як на ділянці $S_0$ впливу на рівень ґрунтових вод не відбувається завдяки наявності штучних покриттів. Маємо таку граничну умову, що характеризує наявність штучних покриттів:

$$\left.\frac{\partial h}{\partial z}\right|_{S_0} = 0. \qquad (2)$$

На ділянці $S \setminus S_0$ відбувається інфільтрація, водовідбір, транспірація і випаровування, тому маємо

$$\left.\frac{\partial h}{\partial z}\right|, S_1, z = L = f + s - g - d - k,$$

де $f$ – додаткове живлення ґрунтових вод (прибуткова частина балансу ґрунтових вод); $s$ – кількість опадів, яка інфільтрується в ґрунтові води (прибуткова частина балансу ґрунтових вод); $g$ – інтенсивність транспірації (видаткова частина балансу ґрунтових вод); $d$ – інтенсивність випаровування (видаткова частина балансу ґрунтових вод); $k$ – водовідбір з ґрунтових вод (видаткова частина балансу ґрунтових вод).

Переходимо до умов, які враховують епапотранспірацію. Оскільки зміна рівнів ґрунтових вод та їх розповсюдження є локальним, і моделювання проводиться для обмежених ділянок міської території (промислових об'єктів, будівель і т.ін.), з однорідними гідрогеологічними умовами, то можна прийняти, що боковий приплив і відтік рівні між собою, тому

$$\begin{cases} \left.\frac{\partial h}{\partial x}\right| x = l + a = e_1(z), & \left.\frac{\partial h}{\partial x}\right| x = -l - a = e_1(z), \\ \left.\frac{\partial h}{\partial y}\right| y = l + b = e_1(z), & \left.\frac{\partial h}{\partial y}\right| y = -l - b = e_1(z). \end{cases}$$

В цих рівняннях згідно з [ ] маємо

$$e_1(z) = \frac{2}{1 + (z / z_{50})^{\tau}}, \qquad (3)$$

де τ – відносна мінливість потенційної транспірації; $y_{50}$ – параметр, що характеризує висоту капілярного всмоктування води; $y$ – глибина, де відбувається тиск вологи, який всмоктує. У розрахунках згідно [4] прийнято значення τ=2,2. У подальших розрахунках прийнято, що $y_{50}$ =3, тобто вважалось, що $L = 6$м. Якщо еавапотранспірація не враховувалась, то значення $L$ обговорюються окремо.

Початковий рівень приймається за точку відліку, $h$=0.

$$h\Big|_{z=0} = 0.$$

Таким чином, сформулювало таку крайову задачу для визначення невідомої функції $h(x, y, z)$. Знайти розв'язок диференціального рівняння (1) при таких крайових умовах:

$$\left.\frac{\partial h}{\partial z}\right|_{S_0} = 0, \left.\frac{\partial h}{\partial z}\right|, S_1, z = L = f + s - g - d - k, \quad h\Big|_{z=0} = 0,$$

$$\begin{cases} \left.\frac{\partial h}{\partial x}\right| x = l + a = e_1(z), & \left.\frac{\partial h}{\partial x}\right| x = -l - a = e_1(z), \\ \left.\frac{\partial h}{\partial y}\right| y = l + b = e_1(z), & \left.\frac{\partial h}{\partial y}\right| y = -l - b = e_1(z). \end{cases}$$

Зауважимо, що неможливо побудувати одну систему базисних функцій для цієї крайової задачі з неоднорідними крайовими умовами на шості межах. Тому в роботі запропоновано шукати невідому функцію $h(x, y, z)$ у вигляді суми трьох доданків

$$h(x, y, z) = h_1(x, y, z) + h_2(x, y, z) + h_3(x, y, z).$$

Кожній функції $h_i(x, y), \quad i = 1,2,3$ відповідає своя крайова задача, при чому в кожній з цих задач наявні однорідні граничні умови, що дає змогу побудувати системи незалежних

базисних функцій. Такий засіб не лише дозволяє побудувати розв'язок сформульованої крайової задачі, що враховує наявність штучних покриттів, інфільтрацію, випаровування та транспірацію, а також ефект евапотранспірації, але й дослідити окремо вплив штучних покритті та ефект евапотранспірації.

Крайова задача для функції $h_1(x,y,z)$ описує наявність штучних покриттів, інфільтрацію, випаровування та транспірацію, але не враховує ефект евапотранспірації в залежності від глибини. Цю задачу сформулюємо наступним чином:

$$\frac{\partial^2 h_1}{\partial z^2}+\gamma_1^2\frac{\partial^2 h_1}{\partial x^2}+\gamma_2^2\frac{\partial^2 h_1}{\partial y^2}=0 \tag{4}$$

$$\left.\frac{\partial h_1}{\partial z}\right|_{S_0}=0,\ \left.\frac{\partial h_1}{\partial z}\right|,S_1,z=L=f+s-g-d-k,\ \left.h_1\right|z=0=0$$

$$\begin{cases}\left.\dfrac{\partial h_1}{\partial x}\right|x=l+a=0, & \left.\dfrac{\partial h_1}{\partial x}\right|x=-l-a=0,\\ \left.\dfrac{\partial h_1}{\partial y}\right|y=l+b=0, & \left.\dfrac{\partial h_1}{\partial y}\right|y=-l-b=0.\end{cases}$$

Для функції $h_2(x,y,z)$ отримаємо таку крайову задачу

$$\frac{\partial^2 h_2}{\partial z^2}+\gamma_1^2\frac{\partial^2 h_2}{\partial x^2}+\gamma_2^2\frac{\partial^2 h_2}{\partial y^2}=0 \tag{5}$$

$$\left.\frac{\partial h_2}{\partial z}\right|_{S_0\cup S_1}=0,\ \left.h_2\right|z=0=0,$$

$$\begin{cases}\left.\dfrac{\partial h_2}{\partial x}\right|x=l+a=e_1(z), & \left.\dfrac{\partial h_2}{\partial x}\right|x=-l-a=e_1(z),\\ \left.\dfrac{\partial h_2}{\partial y}\right|y=l+b=0, & \left.\dfrac{\partial h_2}{\partial y}\right|y=-l-b=0.\end{cases}$$

Аналогічно для функції $h_3(x,y,z)$ маємо

$$\frac{\partial^2 h_3}{\partial z^2}+\gamma_1^2\frac{\partial^2 h_3}{\partial x^2}+\gamma_2^2\frac{\partial^2 h_3}{\partial y^2}=0 \tag{6}$$

$$\left.\frac{\partial h_3}{\partial z}\right|_{S_0\cup S_1}=0,\ \left.h_3\right|z=0=0,$$

$$\begin{cases}\left.\dfrac{\partial h_3}{\partial x}\right|x=l+a=0, & \left.\dfrac{\partial h_3}{\partial x}\right|x=-l-a=0,\\ \left.\dfrac{\partial h_3}{\partial y}\right|y=l+b=e_1(z), & \left.\dfrac{\partial h_3}{\partial y}\right|y=-l-b=e_1(z).\end{cases}$$

Застосовуючи методику, описану в роботі [4], отримаємо такі розв'язки крайових задач (4)-(6)

$$h_1^{mn}=E^{mn}\cos\frac{\pi mx}{\gamma_1(l+a)}\cdot\cos\frac{\pi ny}{\gamma_2(l+b)}\cdot\mathrm{sh}\,\lambda_{mn}z,\quad \lambda_{mn}=\sqrt{\left(\frac{\pi m}{\gamma_1(l+a)}\right)^2+\left(\frac{\pi n}{\gamma_2(l+b)}\right)^2}\ m=1,2,...$$

$$h_2^m=F^m\sin\frac{\pi(0.5+m)z}{L}\cdot\sin\frac{\pi(0.5+n)y}{(l+b)\gamma_2}\cdot\mathrm{sh}\,\lambda_{mn}x,\quad m,n=0,1,2,...$$

$$h_2^m = F^m \sin\frac{\pi(0.5+m)z}{L}\cdot\sin\frac{\pi(0.5+n)x}{(l+a)\gamma_1}\cdot \mathrm{sh}\,\lambda_{mn}y, \quad m,n = 0,1,2,...$$

## ВИСНОВКИ

- Визначено необхідність створення тривимірних математичних моделей для описання змін РГВ та покращення прогнозів їх змін.
- Розроблено тривимірну математичну модель зміни РГВ міських територій, що враховує інфільтрацію атмосферних вод, додаткове живлення в ґрунтові води, транспірацію, випаровування, евапотранспірацію і водовідбір з підземних вод.
- Сформульовано граничні умови математичної моделі.

## ЛІТЕРАТУРА

1. Маринова И. В. Современные математические методы прогноза и планирования эксплуатации водоносного горизонта. Вестник Таганского института управления и экономики №2. 2008. С. 74–77.
2. Венгерський П. С. Чисельне моделювання руху поверхневих і грунтових потоків та їх взаємодія на території водозбору: дис. докт. фіз.-мат. наук : 01.05.02. Львів, 2017. 293 с.
3. Гавич И. К. Теория и практика применения моделирования в гидрогеологии: Москва, Недра, 1980. 358с.
4. Сєрікова, О. М. Прогнозування і управління рівнем ґрунтових вод для підвищення екологічної безпеки забудованих територій України: дисертація канд. техн. наук, спец.: 21.06.01 – екологічна безпека / О. М. Сєрікова; наук. кер. В. В. Яковлєв. Х.: Харківський нац. ун-т міськ. госп-ва ім. О. М. Бекетова, 2019. 166 с.
5. Serikova E., Strelnikova E., Yakovlev V. Mathematical Model of Dangerous Changing the Groundwater Level in Ukrainian Industrial Cities. *Journal of Environment Protection and Sustainable Development*, USA, Vol. 1, No. 2, 2015. P. 86–90.

**СЄРІКОВА Олена Миколаївна** – к. т. н., старший викладач кафедри прикладної механіки та технологій захисту навколишнього середовища; Національний університет цивільного захисту України, вул. Чернишевська, 94 м. Харків, 61023; e-mail: elena.kharkov13@gmail.com; ORCID: 0000-0003-0354-9720.

Наукові інтереси:
- *ґрунтові води;*
- *підтоплення*

**СТРЕЛЬНІКОВА Олена Олександрівна** – д. т. н., професор, провідний науковий співробітник зі спеціальності механіка деформівного твердого тіла; Інститут проблем машинобудування ім. А. М. Підгорного НАН України вул. Пожарського, 2/10, м. Харків, 61046; e-mail: elena15@gmx.com; ORCID: 0000-0003-0707-7214.

Наукові інтереси:
– *теорія і методи розв'язання сингулярних інтегральних рівнянь;*
– *динамічні задачі гідропружності елементів машинобудівних конструкцій;*
– *теорія тріщин в пружних і пружно-пластичних тілах.*

**УДК 004.7(075)**

СЛАБИШЕВ М.О.

# МОДЕЛЬ ПРОЦЕСУ УПРАВЛІННЯ ДОСТУПОМ У БЕЗДРОТОВІЙ КОМП'ЮТЕРНІЙ МЕРЕЖІ

У доповіді розглянуті особливості процесу управління бездротових комп'ютерних мереж.

Метою даної роботи є розробка моделі процесу управління доступом у бездротовій комп'ютерній мережі з запобіганням колізій. Обґрунтування вибору міжкадрових інтервалів в режимах доступу до середовища.

Об'єктом даної роботи є бездротова комп'ютерна мережа в АСУ ТП.

Предмет дослідження - модель процесу управління доступом у бездротовій комп'ютерній мережі з запобіганням колізій.

Розроблена модель процесу управління доступом у бездротовій комп'ютерній мережі, яка призначена для передачі інформації з запобіганням колізій в бездротовій комп'ютерній мережі.

Розроблена модель наочно показує роботу і особливості цього режиму. Необхідність розробки цієї моделі полягала в тому, що при вивченні режимів доступу в бездротової мережі ця модель полегшила процес навчання і розуміння передачі кадрів в бездротових мережах.

Актуальність роботи: до сьогодні існує проблема передачі інформації в бездротових мережах, бо під час передачі часто виникають колізії. Тема є актуальною, бо розроблена модель протоколу управління доступом забезпечує запобігання колізії.

**Дослідження проблем використання бездротових мереж**

АСУ ТП (Автоматизована система управління технологічним процесом) найчастіше потребує модернізації, так як технічний прогрес не стоїть на місці. При цьому, якість виробництва не повинна змінюватися і має бути на однаково високому рівні. Найважливішим елементом модернізації є застосування бездротових технологій, що приносять економію коштів і часу, в порівнянні з розгортанням дротових мереж. [1]

Характерною особливістю бездротових комп'ютерних мереж є те, що вони використовують не множинний доступ з виявленням колізій (CSMA / CD - Carrier Sense Multiple Access with Collision Detection.), а множинний доступ з запобіганням колізій (CSMA / CA - Carrier sensing multiple access with collision avoidance.). [2]

Стандарт бездротових мереж 802.11 намагається уникати колізій за рахунок множинного доступу з запобіганням колізій **CSMA/CA**. Запобігання колізій використовується для того, щоб поліпшити продуктивність передачі, віддавши мережу єдиному передавальному пристрою. Збільшення ефективності досягається за рахунок зниження ймовірності колізій і повторних спроб передачі. Уникнення колізій корисно на практиці в тих ситуаціях, коли своєчасне виявлення колізії неможливо - наприклад, при використанні радіопередавачів. [3]

В даному методі розрізняють 2 режими доступу до середовища: розподілений (DCF - **Distributed Coordination Function**) і централізований (PCF - Point Coordination Function**.**).

**Розподілений режим доступу DCF**

В цьому режимі реалізується безпосередньо метод CSMA / CA. Кожен переданий кадр повинен бути підтверджений кадром позитивної квитанції, який надсилається станцією призначення. Якщо по закінченню тайм-ауту для цього кадру квитанція не надходить, станція-відправник вважає, що сталася колізія.

Режим доступу DCF потребує синхронізації станцій. У стандарті 802.11 ця проблема вирішується так - тимчасові інтервали починають відраховуватися від моменту закінчення передачі попереднього кадру.

Станція, яка хоче передати кадр, зобов'язана попередньо перевірити зайнятість середовища. Як тільки вона підтверджує закінчення передачі попереднього кадру, вона

зобов'язана відрахувати інтервал часу, рівний міжкадровому інтервалу IFS (Inter Frame Space). [4]

Якщо після закінчення IFS середовище все ще вільне, то починається відлік слотів, визначених заздалегідь тривалості (як зазначено на рисунку 1). Кадр можна починати передавати тільки на початку слота за умови, що середовище передачі в цей самий момент вільне. Станція вибирає для передачі слот на підставі усіченого експоненціального двійкового алгоритму відстрочки (алгоритм додавання перевірочних слотів перед передачею основного кадру). Номер слота вибирається як випадкове ціле число, яке рівномірно розподілене в інтервалі [0, CW], де CW означає **Contention Window** (конкурентне вікно). [5]

Для наочності прикладу роботи режиму DCF розроблена модель, яка наочно показує роботу і особливості цього режиму. Необхідність розробки цієї моделі полягає в тому, що при вивченні режимів доступу в бездротової мережі ця модель полегшує процес навчання і розуміння передачі кадрів в бездротових мережах.

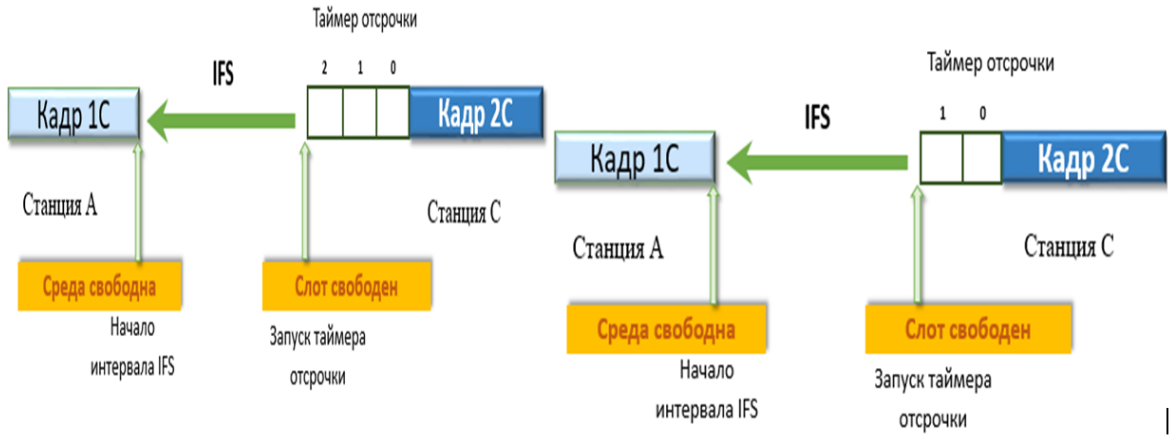



*Рисунок 1 – Відлік тимчасових інтервалів та їх зменшення при вільному середовищі*

Дана імітаційна модель застосовує розподілений режим доступу до середовища при передачі інформації.

У даній моделі підраховується час очікування передачі кадрів. Він обчислюється за формулою (1):

$$t_{ож} = 2 * \frac{D}{V_c} + t_з + t_{a1} + t_{a2} \quad , \qquad (1)$$

де D - це відстань між приймаючої і відправна станціями;

$V_c$- швидкість поширення сигналу (є сталим виразом);

$t_з$- тривалість сигналу запиту (дорівнює розміру одного слота - 1 мкс);

$t_{a1}$і $t_{a2}$- час обробки сигналу на приймаючій і передавальній станції.

Так як $t_{a1}$и і$_{a2}$ вимірюються в наносекундах, при підрахунку цими значеннями можна нехтувати (при роботі в моделі ці обчислення не враховуються). Необхідно тільки знати відстань між станціями.

Отже, цей метод буде успішно працювати і забезпечує запобігання колізій під час передачі кадрів між станціями. Під час роботи моделі можна розрахувати необхідну кількість слотів відстрочки для станції–передавача та час очікування передачі інформації. Модель запобігає колізії, що пришвидшує час передачі інформації від передавача до приймача.

Завдяки формулі (2) знаходиться мінімальна кількість кодових комбінацій, яку необхідно зберігати в запам’ятовуючому приладі(ЗП)(об’єм ЗП):

$$M = 1 + (\frac{t_{ож}}{(m+k)*T_c}), \qquad (2)$$

де $m + k$ – число інформаційних та перевірочних розрядів в кодовій комбінації;

$T_c$- час обробки сигналу;

Час обробки сигналу з формули (2) не є сталою величиною та підраховується за формулою(3):

$$T_c = \frac{1}{B} - \text{час обробки сигналу ;} \qquad (3)$$

де B – швидкість модуляції (дорівнює 2400 біт/с).

Рахується також максимальна відстань, за якою є можливим передача даних між станціями:

$$D_{max} = (M-1) * \frac{(m+k)*V_c}{2*B} - \frac{t_з + t_{a1} + t_{a2}}{2} * V_c . \qquad (4)$$

Якщо $D_{max} < D$, то можна вважати, що передача інформації неможлива, бо відстань для передачі є більшою, ніж відстань, на яку можлива ця передача. [6]

Так як $t_{a1}$ и $i_{a2}$ вимірюються в наносекундах, при підрахунку цими значеннями можна нехтувати (при роботі в моделі ці обчислення не враховуються). Необхідно тільки знати відстань між станціями.

**Результати дослідження роботи моделі**

Дослідження моделі полягає в тому, щоб розрахувати час очікування інформації при різних відстанях між станцією-передавачем і приймаючою станцією. Завдяки отриманому часу очікування знаходиться максимальна відстань між станціями. Цей результат дає зрозуміти доцільність використання саме бездротових технологій в АСУ ТП.


*Рисунок 2 – Залежність часу очікування передачі від максимальної відстані передачі інформації*

Аналіз дослідження моделі показав, що час очікування передачі інформації залежить від максимально можливої відстані передачі інформації. Чим більша буде максимально можлива відстань передачі, тим більший час буде затрачений на передачу інформації. Є доцільним використання саме бездротових технологій в АСУ ТП, бо саме такий метод передачі інформації дозволяє передавати інформацію на великі відстані, не застосовуючи дротових технологій, а час, витрачений на передачу інформації, є оптимальним.

**Висновки**

Таким чином, необхідність розробки такої моделі полягає у використанні методів передачі інформації з запобіганням колізії. Саме цей метод вважається найефективнішим, бо він потребує меншого часу, аніж в інших методах. А час, витрачений на передачу інформації, є критично важливою характеристикою при виборі технологій та методів передачі даних. Є

доцільним використання в АСУ ТП саме бездротових технологій, так як вони є більш економічно вигідними, ніж дротові технології. Метод передачі інформації з запобіганням колізії забезпечують безпомилкову передачу даних на відносно далекі відстані за оптимальний час передачі.

Ця модель може бути використана в навчальній дисципліні «Комп'ютерні мережі» для дослідження передачі інформації в бездротових комп'ютерних мережах. З її допомогою можна розробити спеціальні індивідуальні завдання для студентів для вивчення методів передачі даних в бездротовій комп'ютерному середовищі.

ЛІТЕРАТУРА

1. Електроний ресурс. Стаття про приклад використання бездротових технологій в промисловості.
   URL:http://1234g.ru/wifi/standarty-wifi(Last accessed: 02.02.2020)
2. Лосев Ю. І. Комп' ютерні мережі : навчальний посібник / Ю. І. Лосєв, До. М. Руккас,С. І. Шматков / За редакцією Ю. І. Лосева. Х. : ХНУ імені В. Н. Каразіна, 2013. – 248 с.
3. Електроний ресурс. Блок-схема алгоритму CSMA/CA при передаванні даних.
   URL:https://neerc.ifmo.ru/wiki/index.php?title=Data_link_layer_-_MAC_-_Aloha,_CSMA
   (Last accessed: 02.09.2019)
4. Оліфер В., Оліфер Н. Комп'ютерні мережі. Принципи, технології, протоколи : Підручник для внз. 5–е видавництво – СПб.: Пітер, 2016. – 992 с.: мул. – (Серія «Підручник для внз»).
5. Електроний ресурс. Стаття про підрахунок слотів в конкурентному вікні.
   URL:https://mipt.ru/drec/upload/d27/multihop-wireless.pdf (Last accessed: 28.12.2019)
6. Лосев Ю.І. Методи передачі інформації.: навчальний посібник / Ю. І. Лосев, Н. Д. Плотніков /. :Вірта ім. Л.А.Говорова, 1978р. – 187 с.

**СЛАБИШЕВ Михайло Олегович** – студент кафедри теоретичної та прикладної системотехніки; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: misha220498@gmail.com; ORCID: 0000-0003-2367-1731

Наукові інтереси:

*– використання бездротових мереж в автоматизованих системах управління технологічним процесом.*

**УДК 004.85:616.07**

СТРІЛЕЦЬ В.Є., УГРЮМОВ М.Л., АНТОНЯН И.М., ГЕГЛЮК О.М.

# МЕТОДИ КЛАСИФІКАЦІЇ В ЗАДАЧАХ МЕДИЧНОЇ ДІАГНОСТИКИ

Поява та розвиток дефектів у складних системах – складний динамічний процес. Експерти не завжди можуть спрогнозувати їх появу та поведінку системи при цьому. Не завжди вдається прийти до однієї думки на якій стадії розвитку знаходиться дефект та у якому стані система, які методи необхідно застосувати для ліквідації дефекту і нормалізації стану системи. Контроль і прогнозування динамічного процесу роботи складної системи допомагають експертам приймати рішення, які приводять до кращих показників якості роботи системи.

У якості складної системи була розглянута медико-біологічна система, до складу якої входять лікар, пацієнти й система діагностування стану пацієнтів. Кожний цикл лікування характеризується множиною кінцевих станів пацієнтів. Кількість станів, які розглядаються, визначається експертом на основі результатів кластерного аналізу.

Розгляданню задач класифікації складних систем приділяється багато уваги. На сьогодні опубліковані багато робіт, які присвячені опису методів розв'язання задачі класифікації для технічних і медико-біологічних систем [1-16]. Наведемо деякі з них.

Наївний байєсівський класифікатор – це простий ймовірнісний класифікатор, який спирається на теорему Байєса з «наївним» припущенням про незалежність. Перевагою цього класифікатора є невелика кількість даних, необхідних для навчання, оцінювання параметрів і класифікації. Наївний байєсівський метод часто використовується при класифікації текстів через просту його реалізацію й достатньо високу ефективність класифікації об'єктів, природа яких пов'язана із статистичною різницею [1].

Класифікатор k-найближчих сусідів – це метричний алгоритм для автоматичної класифікації об'єктів або регресії. Цей алгоритм зазвичай використовується для розв'язання задач розпізнавання й інтелектуального аналізу даних. Перевагами даного методу є проста реалізація, висока продуктивність і мала ймовірність помилки [2]. Серед недоліків методу виділяють високу чутливість до шумів у даних, і як наслідок низька ефективність.

Класифікатор випадкових лісів – це алгоритм машинного навчання, основа якого полягає у використанні множини дерев рішень. Алгоритм надає порівняння важливості параметрів і допомагає відокремити значущі властивості для визначення класу або стану [3]. Класифікатор використовується для визначення станів пацієнтів, розпізнавання мови і рукописного тексту, прогнозування молекулярних взаємодій.

Логістична регресія – алгоритм лінійної класифікації, в основі якого лежить ідея побудови роздільної гіперплощини між об'єктами різних класів. Алгоритм часто використовують для отримання початкового результату моделі передбачення, щоб переконатися, що існуюча вибірка даних придатна для опрацювання і побудови гіпотез. Перевагою методу є те, що вихідним результатом є ймовірність. приналежність прикладу до певного класу [4].

Класифікатор дерев рішень – це інструмент підтримки прийняття рішень, який використовується в машинному навчанні, аналізі даних і статистиці. Використовується деревовидна модель рішень і їх можливих наслідків, включаючи випадкові результати подій, витрати і корисність. Зазвичай дерева рішень застосовуються для визначення найбільшої ймовірності досягнення цілі [5].

Класифікатор AdaBoost – це алгоритм, який використовується разом із іншими класифікаторами для підвищення їх ефективності. Алгоритм підсилює класифікатори, об'єднуючи їх у «комітет». Він є адаптивним у тому сенсі, що кожний наступний комітет класифікаторів будується за об'єктами, які були невірно класифіковані попередніми комітетами. AdaBoost є чутливим до шуму у вхідних даних, але він менш піддається перенавчанню у порівнянні з іншими алгоритмами машинного навчання [6].

У роботі [7] детально описані загальні положення теорії штучних нейронних мереж, які навчаються, що широко використовуються для побудови діагностичних моделей у формі рівнянь регресії.

Розглянемо застосування перелічених методів для розв'язання задачі класифікації стану медико-біологічної системи.

**Постановка задачі дослідження**

На основі системного аналізу процесу діагностування стану пацієнта була визначена ієрархія етапів діагностування: лабораторне діагностування (аналізи крові та ін.), візуальне діагностування (УЗД, МРТ тощо) і огляд лікаря. На кожному етапі реєструються відповідні змінні стану пацієнта. Була сформована експериментальна вибірка змінних, які реєструються, що характеризує стан пацієнтів, за якими ведеться спостереження.

Припустимо, що є багатовимірна матриця станів $X = \{x_{i,j}\}$ ($i$=1..$I$, $j$=1..$J$), де $I$ – кількість пацієнтів у вибірці, $J$ – кількість вимірюваних змінних стану. Традиційно рядки цієї матриці називаються прецедентами. Центрування та нормування даних виконується за формулою

$$x_{ij}^{\circ} = \left(x_{ij} - \langle X_j \rangle\right)/\sigma_j, \quad (1)$$

де $\langle X_j \rangle$ – середнє значення $j$-ї змінної стану, $\sigma_j$ – її середнє квадратичне відхилення.

Задача побудови діагностичної моделі: задана векторна функція набором навчальних пар $\left(\vec{X}^{(0)}, \vec{d}\right)_p$, $p$=1..$P$, де $\vec{X}^{(0)}, \vec{d}$ – вектори входу, розмірності $H_0$, і виходу, розмірності $H_{k+1}$, відповідно. Необхідно апроксимувати дану вибірку. Результатом розв'язання задачі повинен бути деякий математичний механізм, який дозволив би отримати будь-яке значення векторної функції $\vec{Y}^{(K+1)}\left(\vec{X}^{(0)}\right)$, яка подається у вигляді навчальної вибірки, за заданим вектором входу у діапазоні, який обмежений вхідними даними.

Сформулюємо постановку задачі класифікації. Нехай $\vec{X}^*$ – вектор змінних, які описують стан прецедентів, $M$ – множина номерів класів (сценаріїв). Відома кількість можливих сценаріїв, а також для кожного сценарію (класу) сформовані підмножини змінних стану (симптоми), за якими спостерігають. За значеннями проекцій вектору $\vec{X}^*$ прецедент відносять до однієї з можливих множин $R_m$, де $m$=0..$M$-1. Необхідно знайти такий $m$-й сценарій, для якого максимальна щільність розподілу умовної ймовірності виникнення $\vec{X}^*$ у прецедента за $m$-м сценарієм:

$$\exists!\ m^* \in C_m\left(\rho\left(\vec{X}_m^* \middle| R_m\right)\right)\ (m = 0..M-1)\colon \rho\left(\vec{X}_m^* \middle| R_m\right) \to max, \quad (2)$$

де $C_m\left(\rho\left(\vec{X}_m^* \middle| R_m\right)\right)$ – множина $m$-х індексів щільностей розподілу умовної ймовірності виникнення $\vec{X}^*$ у прецедента за $m$-м сценарієм.

В якості значень цільової функції використовувалась аргумент логістична функції. Використання логістичної регресії дає можливість передбачення ймовірності виникнення деякої події за значеннями певної кількості ознак.

**Застосування методів класифікації для визначення стану пацієнтів у системах медичного моніторингу**

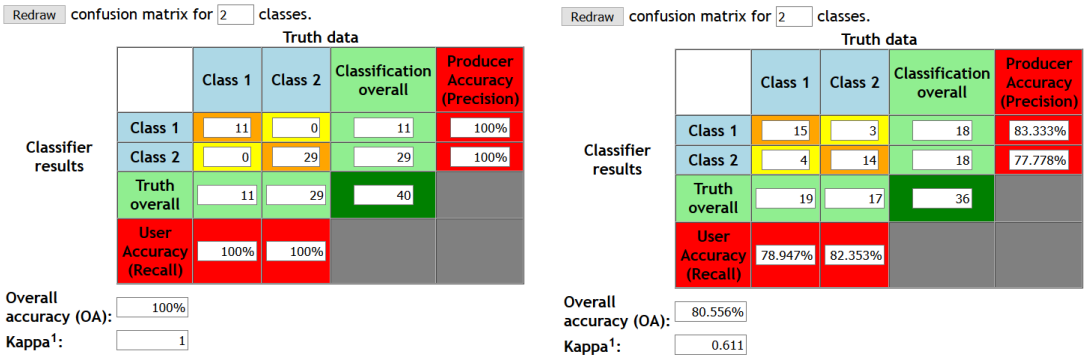
Були розглянуті дві вибірки, які представляють собою дані лабораторних досліджень пацієнтів по двом видам захворювань: захворювання печінки та урологічні захворювання. Дані досліджень надані кафедрою загальної, дитячої та онкологічної урології Харківської медичної академії післядипломної освіти.

Для класифікації були використані наївний байєсівський класифікатор, класифікатор k-найближчих сусідів, класифікатор випадкових лісів, логістична регресія, класифікатор AdaBoost та радіально-базисна нейронна мережа з логістичною активаційною функцією. Результатом класифікації є визначення стану пацієнта (здоровий або хворий).

Вибірка за даними урологічних захворювань містить 47 змінних стану, значення яких були трьох типів: дійсні числа, булеві та перелічувані. Навчальна вибірка складалась з даних щодо 30 пацієнтів, тестова вибірка містила дані щодо 10 пацієнтів.

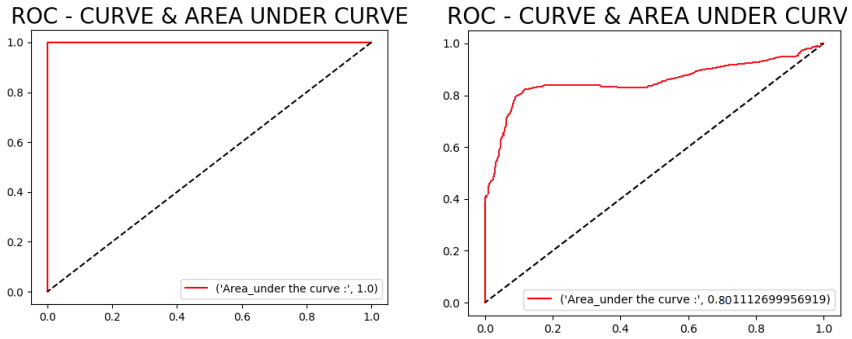
Вибірка за даними захворювань печінки містить 10 змінних стану, значення яких були трьох типів: дійсні числа, булеві та перелічувані. Навчальна вибірка складалась з даних щодо 550 пацієнтів, тестова вибірка містила дані щодо 40 пацієнтів.

В результаті класифікації за обома вибірками даних для аналізу якості класифікації були побудовані матриця помилок [8] та ROC-криві [9] (на рис. 1, 2 наведені результати із використанням радіально-базисної нейронної мережі). Кількісна інтерпретація ROC дає показник AUC. Чим вищий показник AUC, тим якісніше працює класифікатор, при цьому значення 0,5 демонструє непридатність вибраного методу класифікації (відповідає випадковому вгадуванню класу).




*а)* *б)*

*Рис. 1. Матриця помилок за результатами розв'язання задачі класифікації: а – за даними урологічних захворювань; б – даними захворювань печінки*




*а)* *б)*

*Рис. 2. ROC-крива за результатами розв'язання задачі класифікації: а – за даними урологічних захворювань; б – даними захворювань печінки*

Результати оцінки якості класифікації для двох вибірок даних за всіма застосованими методами надані у табл. 1.

*Табл.1 Показники якості класифікації*

| Метод класифікації | Дані за урологічними захворюваннями | | Дані за захворюваннями печінки | |
|---|---|---|---|---|
| | Загальна точність | AUC | Загальна точність | AUC |
| Наївний байєсівський | 73,3% | 0,971 | 60% | 0,727 |
| К-найближчих сусідів | 80% | 0,821 | 81,71% | 0,898 |
| Логістична регресія | 100% | 1,0 | 80,98% | 0,787 |
| Випадкові ліси | 96,67% | 1,0 | 98,86% | 0,996 |
| AdaBoost | 100% | 1,0 | 85,71% | 0,938 |
| Радіально-базисна НМ | 100% | 1,0 | 80,56% | 0.801 |

Таким чином можна зробити висновок, що найякіснішу класифікацію було проведено за допомогою методу випадкових лісів. Цей метод показав кращі результати за обома наборами даних.

ЛІТЕРАТУРА

1. Rish I. An empirical study of the naive Bayes classifier. *IJCAI Workshop on Empirical Methods in AI*. 2001.
2. Altman N.S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*. 1992. 46(3). P. 175–185.
3. Ho Tin Kam. Random Decision Forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*. 1995. P. 278–282.
4. Divya T. A survey on Data Mining approaches for Healthcare. *International Journal of Bio-Science and Bio-Technology*. 2013. 5(5). P. 241-266.
5. Ben-Gal I., Dana A., Shkolnik N. Efficient Construction of Decision Trees by the Dual Information Distance Method. *Quality Technology & Quantitative Management*. 2014. 11(1). P. 133–147.
6. Kégl Balázs. The return of AdaBoost. MH: multi-class Hamming trees. 2013
7. Strilets V., Bakumenko N., Chernysh S., et al. Application of artificial neural networks in the problems of the patient's condition diagnosis in medical monitoring systems. *Intelligent Systems and Computing Integrated Computer Technologies*. 2020. P. 173-185.
8. Stehman S.V. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*. 1997. 62(1). P. 77–89.
9. David M.W. Powers. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation. *Journal of Machine Learning Technologies*. 2011. 2 (1). P. 37–63.

**СТРІЛЕЦЬ Вікторія Євгенівна –** к.т.н.; доцент кафедри прикладної і теоретичної системотехніки; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, м. Харків, Україна, 61022; e-mail: striletsvictoria@gmail.com; ORCID: 0000-0002-2475-1496.

Наукові інтереси:

- *методи машинного навчання;*
- *методи штучного інтелекту.*

**УГРЮМОВ Михайло Леонідович** – д.т.н., професор; професор кафедри прикладної і теоретичної системотехніки; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, м. Харків, Україна, 61022; e-mail: ugryumov.mykhaylo52@gmail.com; ORCID: 0000-0003-0902-2735.

Наукові інтереси:

- *методи машинного навчання;*
- *комп'ютерне та математичне моделювання.*

**АНТОНЯН Ігор Михайлович** – д.мед.н., доцент; завідувач кафедри загальної, дитячої та онкологічної урології; Харківська медична академія післядипломної освіти, вул. Амосова, 58, м. Харків, Україна, 61176; e-mail: im.antonyan@gmail.com; ORCID: 0000-0002-5594-6210.

Наукові інтереси:

- *регенераційна медицина.*

**ГЕГЛЮК Оксана Миколаївна**– кафедра загальної, дитячої та онкологічної урології; Харківська медична академія післядипломної освіти, вул. Амосова, 58, м. Харків, Україна, 61176; e-mail: oksanagegluk@gmail.com; ORCID: 0000-0002-7237-228X.

Наукові інтереси:

– *методи аналізу даних у медичній діагностиці.*

**УДК 004.42**

ТЕЛЕЖЕНКО Д.О.

# СТАНДАРТИЗАЦІЯ ФОРМУЛЮВАННЯ ЗАПИТІВ ТА ОБРОБКИ ВІДПОВІДЕЙ ШЛЯХОМ ВИКОРИСТАННЯ МОВИ ЗАПИТІВ GRAPHQL НА ПЛАТФОРМІ FLUTTER

## Загальні дані про GraphQL у мобільних рішеннях

GraphQL спрощує робочий процес для створення клієнтських додатків, таких як iOS, Android. Майже кожен бізнес сьогодні "йде" мобільним шляхом та інвестує значні кошти у створення багатоплатформних додатків. Взаємодія замовника з продуктом чи послугою значно збільшується, якщо він має можливість доступу до нього через свої телефони, що призводить до збільшення їх доходу. Сьогодні є понад 2,5 мільярда користувачів смартфонів, і прогнозується, що це зросте в майбутньому. Це робить ще важливішим, щоб мобільні додатки були чуйними та мали мінімальні затримками. Не менш важливо швидко створювати ці додатки. Додатки також будуються на різних факторах форми, і є необхідність у спрощенні розробки додатків на всіх цих платформах. Саме тут GraphQL стане у нагоді. Це допомагає клієнтам отримати потрібну кількість даних, необхідних для надання представлення даних. GraphQL дозволяє клієнтам визначати форму відповіді на кожен запит. На додаток до цього, це усуває складність управління API Endpoint для цих клієнтів, оскільки він відкриває єдину кінцеву точку HTTP для отримання необхідних даних.

Так як сучасні мобільні додатки є «розширенням» вже існуючого продукту, чи то будь-який сервіс, мета якого поліпшити життя користувача, додаток повинен швидко та беззбійно працювати. Розглянемо наступний приклад. Припустимо, що створюється додаток для вивчення іноземної мови, де на головному екрані будемо мати список тем. Коли буде обрана певна команда, ми перейдемо до іншого екрану, який буде відображати список питань певного рівня та буде містити наступні дані: тип питання, питання, список відповідей, який у свою чергу є вірним чи невірним та додаткові дані (наприклад список зображень для вивчення конкретного слова).

## Порівняння з REST

Якщо писати такий додаток з використанням REST API, даний додаток буде викликати багато запитів, та деякі запити будуть містити зайву інформацію.

Завдяки GraphQL ми можемо уникнути зворотних «походів» на серверну частину для отримання ієрархічних або пов'язаних даних. У вищенаведеному прикладі, коли була обрана певна команда, ми спочатку повинні отримати список рівнів, а потім «запитати» інформацію про певний рівень та зв'язати їх разом на мобільному додатку. Насправді ми можете зробити два запити паралельно, але це не завжди можливо. Якщо для цього прикладу ми зробимо розширення під планшети, та повинні будемо відображати у горизонтальному режимі зліва список рівнів, а праворуч відображати питання. У цьому випадку, якщо робити стандартним підходом через REST API, спершу отримаємо список рівнів, а вже потім для кожного вибору, відобразимо список рівнів. Це призводить до декількох зворотних переходів до сервера, що безпосередньо впливає на затримку та UX програми. Використовуючи GraphQL, є можливість спростити цей робочий процес, побудувавши єдиний вкладений запит, де ми отримуєте всю необхідну інформацію в один круговий пробіг. Відповідальність за отримання пов'язаних даних покладається на сторону сервера.

GraphQL запит є структурованим, та клієнт може побудувати наступний запит, щоб отримати інформацію про список активних чи вибраних питань за допомогою єдиного мережевого виклику:

```
query FetchLevelsAndQuestions(questionId: 1)
 QuestionsData{
 questionList
 id
 answers
 }
)
```

**Висновок**

Так як GraphQL є перспективним протоколом, його можна використовувати у будь-якій мові програмування, та на будь-якій платформі. Проте, призначення протоколу, як і будь-якої платформи призначена для полегшення та прискорення розробки програми. На даний момент на платформі Flutter з цим великі проблеми, бо немає простої та якісної реалізації, що буде задовольняти цим важливим питанням.

Оптимізація цього процесу, буде складатися з наступних кроків:

- Написання коду-бібліотеки, що полегшить використання даного протоколу на практиці. Це дасть змогу дійсно прискорити написання коду, що буде зв’язуватись з серверною частиною через даний протокол.

- Написання «гнучкого» коду, який не буде прив’язаний до одного проекту, а буде можлива конфігурація для його використання для різних проектів.

Дані кроки дозволять зробити використання GraphQL простим для реалізації цього протоколу. Та позбавлять користувача від написання дуже багатьох строк коду.

ЛІТЕРАТУРА

1. Бенкс А., Порселло А. GraphQL: язык запросов для современных веб-приложений. Питер, 2019. 240 с.
2. Віндмайл Е., Flutter in Action. Manning, січень 2020. 368 с.
3. Документація з мови програмування Dart. URL: https://dart.dev/
4. Документація по платформі Flutter. URL: https://flutter.dev/

**ТЕЛЕЖЕНКО Денис Олександрович** — бакалавр, студент кафедри теоретичної та прикладної системотехніки, Харківський національний університет імені В.Н. Каразіна, площа Свободи, 4, Харків-22, Україна, 61022; e-mail: denisque75@gmail.com;
ORCID: 0000-0002-8377-8517

Наукові інтереси:

- *Розробка мобільних додатків.*

**УДК 004.7**

ТЕРЬОХІН В.Л, СТЕРВОЄДОВ М.Г, РІДОЗУБ О.В.

# ІНТЕЛЕКТУАЛЬНІЙ ВУЗОЛ СЕНСОРНОЇ МЕРЕЖІ РАДІАЦІЙНОГО МОНІТОРИНГУ

**Постановка проблеми**

Бездротові сенсорні мережі (БСМ) є найбільш перспективними технологіями для моніторингу великих територій і контролю складних об'єктів. Вони являють собою сукупність територіально розподілених сенсорних вузлів призначених для збору, попередньої обробки даних про параметри оточуючого середовища і передачі інформації віддаленим користувачам. Їх доцільно використовувати в таких застосунках, де неможливо, важко або дорого експлуатувати провідні системи.

Аналіз літературних даних і інформації з Інтернету показує, що для такої важливої проблемі, як с і контролю радіаційних умов на радіаційно-небезпечних об'єктах вкрай недостатньо розроблено датчиків випромінювання, які б задовольняли сучасним вимогам до вузлів сенсорних мереж. Тому, метою роботи було створення інтелектуального сенсорного вузла, на базі напівпровідникового CdZnTe детектора, прецизійних аналогових мікросхем, потужного 32-розрядного мікроконтролера (МК) і радіочастотного модуля ESP32 TTGO Lora. Довготривала робота вузла забезпечується літій-іонними акумуляторами з підзарядкою від сонячних елементів.

**Структурна схема сенсорного вузла**

На рис. 1 представлено структурну схему розробленого вузла для бездротової мережі радіаційого моніторингу навколишнього середовища.

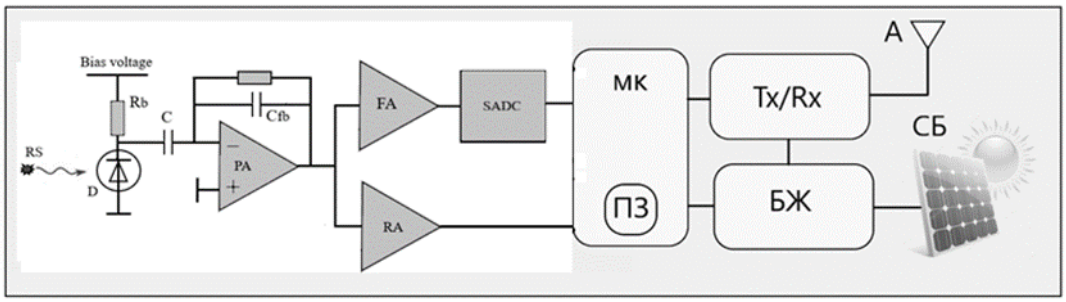



*Рис. 1 Структура сенсорного вузла для бездротової мережі радіаційного моніторингу.*

Радіаційний детектор D з попереднім зарядочутливим підсилювачем PA підключено до основного спектрометричного FA і швидкого RA підсилювачів, вихід яких підключений до спектрометричного аналого-цифрового перетворювача SADC і далі до мікроконтролера МК STM32F4. Попередньо оброблені дані передаються в трансивер Tx/Rx, в якості якого застосовано модуль ESP32-SX1278-Lora - нова зручна, ефективна й економічна розробка для роботи з мережевими програмами. Основним чіпом модуля є Lexin ESP32, двоядерний процесор Tensilica LX6 з тактовою частотою 240 МГц, обчислювальна потужність до 600DMIPS, вбудована мікросхема 520 КБ SRAM, 802.11 b / g / n HT40 Wi-Fi приймач, базова частота, стек протоколів і LWIP, вбудований дворежимний Bluetooth - стандартний Bluetooth і Bluetooth з низьким енергоспоживанням. Вбудована 32MByte Flash пам'ять, Wi-Fi антена, 0,96-дюймовий синій OLED-дисплей, USB інтерфейс на CP2102. Зручною для роботи з модулем є підтримка середовища розробки Arduino.

**Принципова схема мікроконтролерної частини вузла**

На рис. 2 зображена принципова схема процесорної частини приладу. Її ядром є мікроконтролер STM32L485RGT6A. Вибір цього сучасного МК визначено його достатньою потужністю, малим енергетичним споживанням і багатою внутрішньою архітектурою.

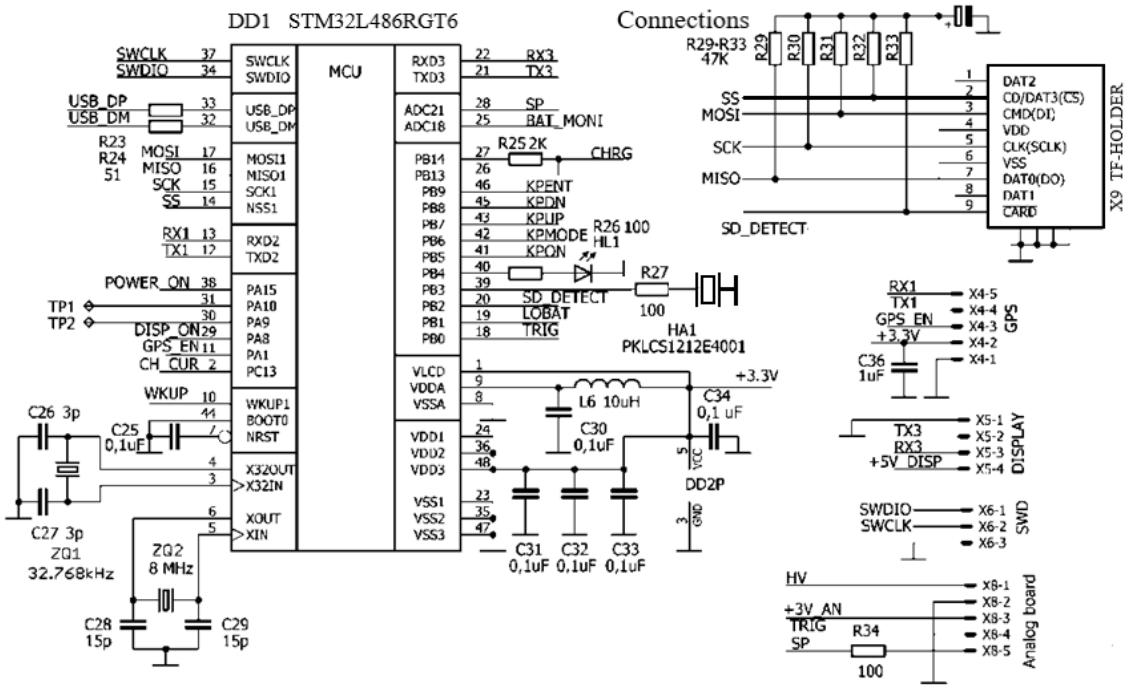



*Рис.2 Процесорна частина сенсорного вузла*

Алгоритми визначення потужності експозиційної дози.

CdZnTe детектор відрізняється високою роздільною здатністю, але має сильну енергетичну залежність чутливості. Тому, для використання його в якості датчика потужності експозиційної дозі необхідно проводити адаптивну програмну корекцію результатів вимірювань. [1]€

Для економії часу і енергії програма розрахунку дози випромінювання працює в двох режимах - режимі попередньо встановленого часу експозиції або в режимі апроксимації, чи заданої статистичної невизначеності. [2] На рис.1 приведено діаграму діяльності, яка показує послідовність дій, необхідних для реалізації цього завдання.

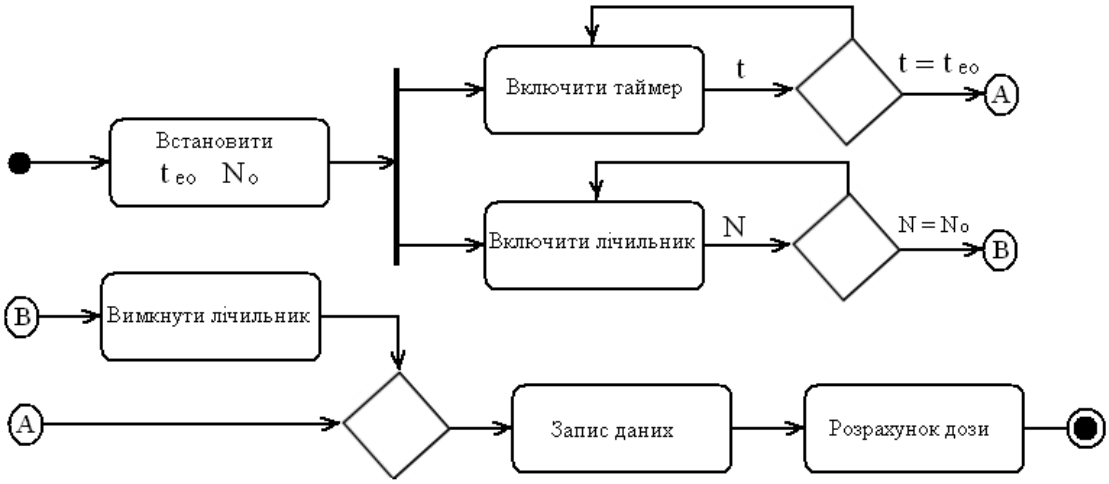



*Рис. 3. Діаграма діяльності*

Після входу в програмний модуль встановлюється час експозиції $t_{eo}$ та кількість імпульсів $N_o$ що визначає задану статистичну погрішність. Потім одночасно включається таймер зворотного відліку часу і лічильник, який підраховує кількість імпульсів з детектору. Коли виконується умова досягнення числа імпульсів N заданого їх значення $N_o$, то підрахунок імпульсів припиняється. В подальшому дані записуються в пам'ять процесора. Програма, що враховує подібність трикутників, обчислює кількість імпульсів, яка була би при цьому за час експозиції. Відповідно, це числа N4 і N3. Якщо за максимальний час експозиції $t_{eo}$ кількість

імпульсів N не досягається попередньо встановленого значення $N_o$, то підрахунок імпульсів також припиняється при $t = t_{eo}$. При цьому, статистична невизначеність ($\sqrt{N}$ / N) буде більшою.

Потім програма переходить до реалізації модуля розрахунку дози згідно із запропонованим алгоритмом [3].

Експозиційна доза $D_{exp}$ обчислюється по алгоритму середньої амплітуді імпульсів, яка визначається з амплітудного спектру. Амплітудний спектр імпульсів відповідає енергетичному спектру гама випромінювання, який реєструється. Програма розрахунку експозиційної дози $D_{exp}$ обчислюється за формулами:

$$D_{exp} = Nt(M E_{ph} + C) \quad (1)$$

де

$N_t$ - загальне число імпульсів за час експозиції в обраному діапазоні енергій,

M і C - константи, які визначаються при калібруванні детектора,

$E_{ph}$ - енергетичний еквівалент середньої амплітуди імпульсів:

$$E_{ph} = \{[\sum_k kN(k)]/N_t\}E_{adc} \quad (2)$$

Кінцево

$$D_{exp} = M [E_{adc} \sum_k kN(k)] + Nt C \quad (3)$$

N (k) - число імпульсів в каналі k,

$E_{adc}$ - ціна каналу аналого-цифрового перетворювача (АЦП) в багатоканальному амплітудному аналізаторі.

На вбудованому в мікроконтролер 12 розрядному АЦП в залежності від застосованого детектора можна реалізувати від 256 до 4096 каналів аналізатора. Тому ціна каналу адаптивно змінюється в залежності від точності вимірювань, яка задається, і потужності дози. Така зміна режимів АЦП також дозволяє економити енергію джерел живлення.

ЛИТЕРАТУРА

1. Ажажа В.М. Приборы на основе CdTe и CdZnTe для технологического контроля и мониторинга радиационной обстановки на АЭС / В.М. Ажажа, В.Е. Кутний, А.В. Рыбка, Л.Н. Давыдов, И.Н. Шляхов, А.А. Захарченко, Д.В. Кутний, Д.В. Наконечный // Наука та інновації. – 2006. – Т. 2, № 6. – С. 31–38.
2. Терьохин В.Сенсорний вузол для бездротової мережі радіаційного моніторингу О. Рідозуб, М. Стєрвоєдов, В. Терьохін, С. Фомін. N. Вісник Харківського національного університету імені В.Н. Каразіна, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»: 2018, Том 39.
3. Захарченко А.А. Моделирование дозиметрических характеристик CdTe (CdZnTe) детекторов гамма - излучения / А.А. Захарченко, В.Е. Кутний, А.В. Рыбка, М.А. Хажмурадов // Радиоэлектроника и информатика: Научн.-техн. журнал. – 2006. – № 2. – С. 28–33.

**ТЕРЬОХІН Віталій Леонідович** – аспірант кафедри електроніки та управляючих систем Харківського національного університету імені В. Н. Каразіна, пл. Свободи, 6, Харків-22, Україна, 61022; e-mail: kbs-com@karazin.ua ORCID: 0000-0001-7653-4488.

Наукові інтереси:

*– Бездротові сенсорні мережі (БСМ)*

**РІДОЗУБ Олег Володимирович** – студент 4 курсу факультету компь'ютерних наук Харківського національного університету імені В. Н. Каразіна, пл. Свободи, 6, Харків-22, Україна, 61022; e-mail: ridozub@gmail.com; ORCID: 0000-0003-0136-6437.

**СТЕРВОЄДОВ Микола Григорович** – к. т. н., доцент; завідувач кафедри електроніки та управляючих систем Харківського національного університету імені В. Н. Каразіна, пл. Свободи, 6, Харків-22, Україна, 61022; e-mail: styervoyedov@yahoo.com; ORCID: 0000-0003-0136-6437.

**УДК 004.728:519.87**

ТКАЧЕНКО А.М, АРТЮХ О.А.

# МОДЕЛЬ МУЛЬТИСЕРВІСНОЇ МЕРЕЖІ ДЛЯ ПЕРЕДАЧІ АУДІО І ВІДЕО ДАНИХ

**Мета**

Метою даної роботи є розробка моделі мультисервісної мережі для передачі аудіо та відео даних.

**Завдання**

Під час виконання роботи виконано наступне:

- розглянуті та проаналізовані можливі архітектури мультисервісних мереж;
- розглянуті методики контролю та діагностики мереж;
- розроблена модель мультисервісної мережі;
- проведені випробування розробленої моделі та проаналізовані отримані результати.

**Об'єкт дослідження**

Об'єктом дослідження є методи та засоби інформаційних технологій для автоматизації розробки, контролю та управління обладнанням мультисервісних мереж.

**Предмет дослідження**

Предметом дослідження є модель мультисервісної мережі для передачі аудіо та відео даних.

**Актуальність**

В сучасному технологічному світі одною з важливих проблем є створення високоефективного телекомунікаційного середовища. Без вирішення цієї проблеми неможлива побудова якісного інформаційного товариства та впровадження новітніх технологій в усі сфери життя: медицину, науку, освіту, бізнес, різноманітні верстви виробництва та повсякдення. Інформація – найважливіший світовий ресурс, тому для високоефективного обміну даним ресурсом та задоволення потреб інформаційної спілки необхідно створення сучасних мереж доступу, які здатні забезпечити можливість надання широкої гами послуг.

Таким чином, вирішення поставленого завдання та отримання готового ефективного продукту є доцільним та актуальним [1].

**Основна частина**

**Мультисервісна мережа** - універсальне багатоцільове середовище, призначене для передачі мови, зображення, аудіо та відео даних з використанням технології комутації пакетів (IP). Мультисервісна мережа відрізняється ступенем надійності, характерної для телефонних мереж і забезпечує низьку вартість передачі певного об'єму інформації [2].

**Основне завдання мультисервісних мереж** полягає в забезпеченні роботи різнорідних інформаційних та телекомунікаційних систем і додатків в єдиному транспортному середовищі, коли для передачі звичайного трафіку і трафіку іншої інформації використовується єдина інфраструктура.

**Вимоги до мультисервісних мереж**

Мультисервісні мережі дозволяють операторам розширити свої мережеві магістралі в напрямку надання нових сервісів, пропонуючи додаткові послуги для широкого кола корпоративних клієнтів. Під мультисервісними мережами ми розуміємо надання різнорідних телекомунікаційних послуг по єдиній інфраструктурі передачі даних.

Коли мова заходить про реалізацію мультисервісних мереж, зазвичай підлягають розгляду чотири технічні питання: пропускна здатність, затримка, розсинхронізація, управління.

Мультисервісні мережі вимагають зовсім іншого підходу. Доставка відео і голосу повинна здійснюватися в реальному часі - з необхідністю пріоритетності у разі перевантажень транспортної мережі. Однак мережева індустрія ніколи не орієнтувалася на мережі реального часу, дані доставлялися у відповідності з можливостями мережі в конкретний проміжок часу.

**Головною перевагою** мультисервісної мережі є використання єдиного каналу для передачі даних різних типів. Також дозволяє зменшити різноманітність типів обладнання, застосовувати єдині стандарти, технології і керувати комунікаційним середовищем.

**Основними складовими мультисервісної мережі є:** телепорт, транспортна мережа і кластери. Топологія мережі визначається специфікою місцевості, на якій вона розгортається.

**Телепортом** - єдиний центр управління, отримання, обробки, створення і передачі інформації. Телепорт будується за модульною технологією (з можливістю поетапного нарощування послуг, що надаються) і формується з обладнання і програмного забезпечення (ПО) для організації прийому ефірних і супутникових ТВ- і радіопрограм.

**Транспортна мережа** - двонаправлена широкосмугова магістральна кабельна мережа, побудована по волоконно-оптичній технології зі структурою «кільце» або «зірка». На транспортній мережі розташовуються вузли введення-виведення і обробки інформації, до яких здійснюється підключення телепорту і кластерів.

**Кластери** являють собою групи від 500 до 2 тисяч абонентів. Абоненти територіально розташовані в безпосередній близькості один від одного, і охоплюються інтерактивною розподільною мережею.

При створенні проектів мультисервісних мереж рекомендується використовувати активне мережеве обладнання від провідних світових виробників, серед яких виділяють компанії Cisco Systems і Nortel Networks. Вибір продукції саме цих виробників обумовлений широким спектром пропонованого обладнання і підтримкою найсучасніших технологій. Це дозволяє створювати рішення з централізованим управлінням і вирішує проблеми сумісності обладнання.

В даній роботі для реалізації поставленої задачі використано один з найзручніших програмних пакетів для моделювання мереж – Cisco Packet Tracer.

Cisco Packet Tracer – це багатофункціональна програма моделювання мереж, яка дозволяє експериментувати з поведінкою мережі. Packet Tracer надає функції моделювання, візуалізації, авторської розробки, атестації та співробітництва, а також полегшує викладання і вивчення складних технологічних принципів [3].

Розроблена модель надає такі можливості при моделюванні функціонування мережі:

- можливість передачі відео файлів різного типу та розміру;
- можливість передачі аудіо файлів різного типу та розміру;
- можливість легкого додавання окремих елементів мережі;
- можливість припинення несанкціонованого доступу до мережі;
- можливість збільшення абонентів;
- можливість збільшення ресурсів мережі;
- перевірка доступності вузлів;
- підтримка різних протоколів.

Для реалізації схеми моделі мультисервісної мережі передачі аудіо та відео даних в середі моделювання необхідно використати такі пристрої:

- роутер;
- комутатори;
- сервер;
- ноутбук чи персональні комп’ютери.

Аналіз функціонування моделі та отриманих результатів моделювання показав, що розроблений продукт відповідає висунутим вимогам і спроможний відтворювати процеси, які протікають в мережі при передачі аудіо та відео даних, створювати різні робочі сценарії та оцінювати ступінь їх виконання.

**Висновки**

Концепція мультисервісності охоплює безліч аспектів побудови мережі, що дозволяє домогтися необхідної якості вирішення завдань користувачів.

Мультисервісна мережа здатна надати єдину платформу передачі аудіо та відео даних, за рахунок застосування інтегрованих рішень. Спектр пропонованого обладнання та його можливості дозволяють вже сьогодні будувати складні інтегровані мережі, здатні задовольнити вимоги вибагливих клієнтів.

Використання моделювання в процесі розробки, створення та експлуатації МСМ дозволяє суттєво зменшити капіталовкладення на протязі всього життєвого циклу мережі, створити та проаналізувати умови та ситуації, поява яких в ході її використання виникають дуже рідко, або можуть викликати незворотні наслідки в реальних умовах [4].

ЛІТЕРАТУРА

1. online.rae.ru - Актуальность мультисервисной сети связи [Електронний ресурс] Режим доступу: http://online.rae.ru/1359
2. znetwork.narod.ru - Мультисервисные сети [Електронний ресурс] Режим доступу: http://znetwork.narod.ru/Theory/MSS.htm
3. cisco.com - Cisco Packet Tracer [Електронний ресурс] Режим доступу: https://www.cisco.com/c/ru_ua/training-events/netacad/training-courses/cisco-packet-tracer.html
4. compress.ru - Мультисервисные сети [Електронний ресурс] Режим доступу: https://compress.ru/article.aspx?id=9404
5. dut.edu.ua - Телекомунікаційні системи та мережі наступного покоління [Електронний ресурс] Режим доступу: http://www.dut.edu.ua/uploads/l_1762_12226724.pdf
6. ea.donntu.org - Концепція мультисервісних мереж Бєльков Д.В. [Електронний ресурс] Режим доступу: http://ea.donntu.org:8080/bitstream/123456789/3252/1/Концепція%20мультисервісних%20мереж.pdf
7. ru.wikipedia.org – Мультисервиснаяя сеть связи [Електронний ресурс] Режим доступу: https://ru.wikipedia.org/wiki/Мультисервисная_сеть_связи
8. dut.edu.ua - Проектування мультисервісної мережі [Електронний ресурс] Режим доступу: http://www.dut.edu.ua/uploads/l_471_35899605.pdf

**ТКАЧЕНКО Андрій Максимович** – студент кафедри теоретичної та прикладної системотехніки Харківського національного університету імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: gadess11@outlook.com; ORCID: 0000-0002-0547-3101.

Наукові інтереси:

*– моделювання інформаційних процесів у складних і розподілених системах.*

**АРТЮХ Олексій Анатолійович** – старший викладач кафедри теоретичної та прикладної систематехніки; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail:; ORCID:.

Наукові інтереси:

*– моделювання інформаційних процесів у складних і розподілених системах.*

**УДК 65.0(075.8)**

ТОЛСТОЛУЗЬКИЙ Є.Д., БЕРДНІКОВ А.Г.

# ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ СЧС ПРИ ОЦЕНКЕ РИСКОВ В IT ПРОЕКТАХ

**Введение**

Современный рынок IT проектов характеризуется постоянным ростом требований к качеству проектов и снижению затрат, связанных с их разработкой. Поэтому при организации работ по проекту необходимо учитывать все критичные факторы, влияющие на разработку. В связи с этим необходимо уметь оценивать риски предприятия, учитывать возможные последствия, которые они могут повлечь за собой, и разрабатывать рекомендации по их преодолению.

Так как большинство управленческих решений принимаются в условиях неопределенности, то правильная оценка рисков помогает минимизировать финансовые убытки, предотвратить нарушения сроков работ и существенно снизить возможные затраты человеческих и материальных ресурсов. Поэтому тема работы, связанной с исследованием возможности применения семантико-числовых спецификаций (СЧС) при оценке рисков в IT проектах представляется актуальной.

**Постановка задачи**

Методы оценки вероятности появления рисков, их последствий, а также эффективности принятых решений по предупреждению рисковых ситуаций могут быть различными: построение матриц, деревьев решений, использование возможностей теории игр и т.д.

В данной работе уделяется внимание управлению рисками в IT проектах на основе метода дерева решений, который применяется при решении сложных многоэтапных вероятностных проблем. В теории игр дерево решений представляет собой иерархическую модель, которая позволяет разбить большую и сложную проблему принятия решения в условиях риска на совокупность структур (подмоделей) с меньшими проблемами (разбиение работ, организация управления проектом, стоимость проекта, материальные ресурсы проекта).[1]

На рис.1 приведен общий вид виртуального дерева решений, которое может быть использовано для оценки рисковой ситуации.

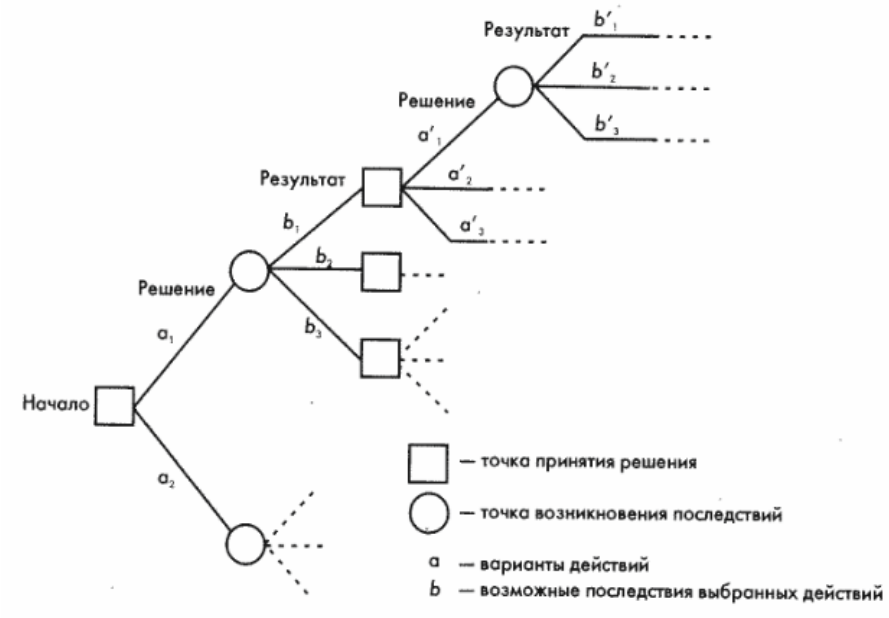



*Рис.1. Дерево решений*

Дерево решений включает в себя следующие элементы: корневой узел исходной задачи (начало), узлы (точки) принятия решений, узлы (точки) возникновения событий, линии возможных вариантов действий, конечные узлы возможных последствий предпринятых действий.

Такое дерево решений обеспечивает процесс систематизации и идентификации рисков в зависимости от уровня детализации проблемы и может быть использовано для проведения качественного и количественного анализа рисковой ситуации.

Этапы разработки дерева решений содержат следующие операции:

1) разделение процесса идентификации рисков на отдельные фазы;

2) оценка вероятности результатов для каждой отдельной фазы;

3) определение точек принятия решений;

4) вычисление результатов принятых решений в конечных узлах;

5) прохождение дерева решений в обратном порядке (с целью убедиться в том, что каждая ситуация идентифицирована и оценена).

**Суть модели**

Оценку рисков при реализации IT проекта предлагается провести на базе семантико-числовых спецификаций (СЧС).

Семантико-числовые спецификации представляют собой математический аппарат, использующийся в алгоритме формального структурно — семантико — числового синтеза вычислительных подсистем параллельных неперестраиваемых спецпроцессоров.

Для получения временной параллельной граф-схемы необходимо выполнить синтез структур семантико - числовой спецификации. При этом формируются базовые структуры СЧС: структура BF (basic file) СЧС состава операторов и структура CF(comunication file) СЧС связей операторов.

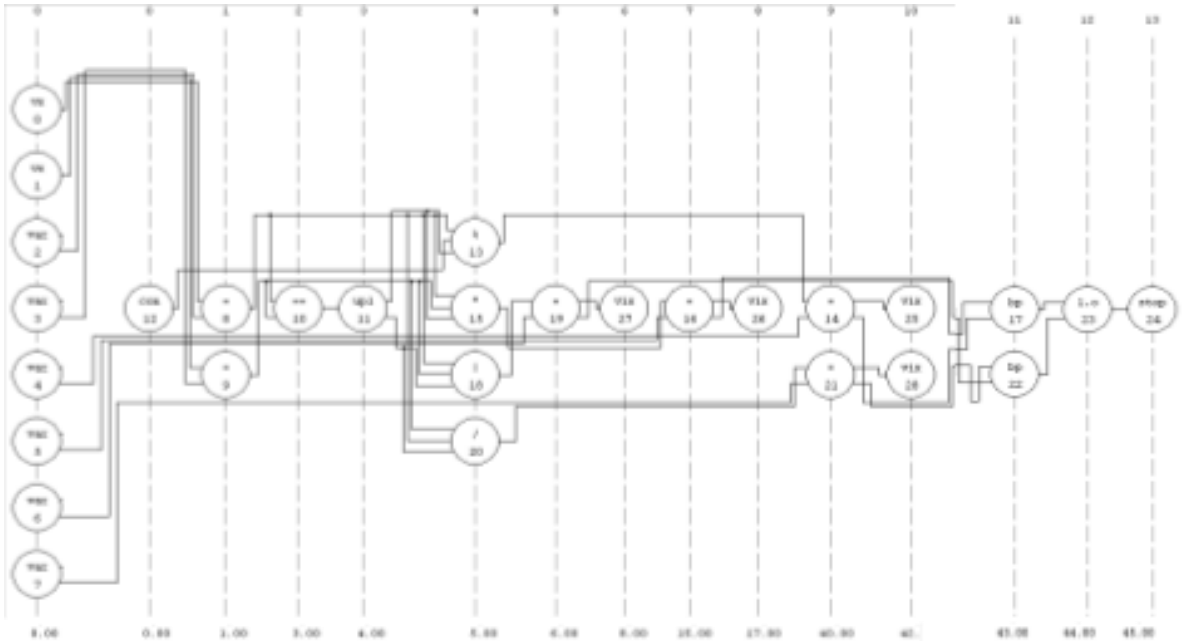


*Рис. 2. Граф-схема возможных рисковых ситуаций*

Динамические субъекты процесса синтеза (параллельных временных моделей алгоритмов) представляются в формате СЧС с помощью трех структур данных, оформленных в виде соответствующих файлов: основного файла BF и файла указателей CF, и временного файла TF, который имеет следующий вид [2-6]

$$TF = (N, NT),\ N \in 0,p-1,$$

где p − количество вершин;

N− номер вершины графа или номер вершины модели;

NT− момент времени, в который начинается выполнение j-й инструкции параллельной модели алгоритма, интерпретируемой j-й вершиной соответствующей временной параллельной граф-схеме, изображенной на рис.2. Где вершины – точки принятия решений и точки возникновений последствий. Каждая вершина харакеризуется временем реализации задачи. Вертикальные ярусы – характеризуют моменты дискретного времени наступления следующего этапа.

Анализ представленной граф-схемы с точки зрения возможных рисковых ситуаций в IT проекте (задержка времени выполнения определенных операций, нарушение синхронизации вычислительного процесса, некорректные промежуточные результаты и т.п.) позволит оценить возможные временные или финансовые потери при реализации проекта.

Таким образом, аппарат семантико-числовых спецификаций можно применять для визуализации деревьев решений анализа рисков IT проектов, спрогнозировать возможные риски и рекомендовать мероприятия по их преодолению.

ЛІТЕРАТУРА

1. Управление проектом. Основы проектного управления : учебник / кол. авт. ; пол ред. проф. М.Л. Разу. — М. : КНОРУС, 2006. — 768 с.
2. Гергель, В.П., Стронгин, Р.Г. (2003, 2 изд.). Основы параллельных вычислений для многопроцессорных вычислительных систем. - Н.Новгород, ННГУ.
3. Воеводин В.В., Воеводин Вл.В. (2002). Параллельные вычисления. – СПб.: БХВ-Петербург.
4. Немнюгин С. (2009). Модели и средства программирования для многопроцессорных систем– СПб.: БХВ-Петербург.
5. Хьюз К., Хьюз Т.(2004). Параллельное и распределенное программирование на C++.: Пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 672с.
6. Поляков Г.А., Шматков С.И., Толстолужская Е.Г., Толстолужский Д.А. (2012). Синтез и анализ параллельных процессов в адаптивных времяпараметризованных вычислительных системах. – Х.: ХНУ имени В.Н. Каразина, 2012.-672с.

**ТОЛСТОЛУЗЬКИЙ Євген Дмитрович –** студент факультету комп`ютерних наук Харківського національного університету імені В.Н. Каразіна, майдан Свободи, 4, м. Харків, Україна, 61022; e-mail: evventol@gmail.com; ORCID: 0000-0002-2039-0267.

Наукові інтереси:

− *застосування методів теорії ухвалення рішень при управлінні складними телекомунікаційними системами.*

**БЕРДНІКОВ Анатолій Георгійович** – к.т.н., доцент кафедры ТПС, доцент; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, м. Харків, Україна, 61022; e-mail: a.berdnikov@karazin.ua.

Наукові інтереси:

− *застосування методів теорії ухвалення рішень при управлінні складними телекомунікаційними системами.*

**УДК 004.94**

ТОТКАЛ С.О.

# РОЗРОБКА ОПТИМАЛЬНИХ АЛГОРИТМІВ ЕМІСІЇ ЕЛЕКТРОНІВ ІЗ ПЛАЗМОВОГО ФАКЕЛА

**Вступ**

Комп'ютерне моделювання широко використовується як інструмент дослідження фізичних процесів, зокрема в сильнострумових прискорювачах електронних потоків та генераторах електромагнітного випромінювання на їх основі. В таких приладах в якості джерела електронів часто використовуються так звані вибухоемісійні катоди [1]. Під дією електричного поля високої напруженості поблизу такого катода утворюється тонкий шар щільної плазми (плазмовий факел) з якої електричне поле «витягає» електрони в вакуумну область де вони взаємодіють з електромагнітними полями. Предметом дослідження при цьому як правило є динаміка електронів та електромагнітних полів у вакуумній області.

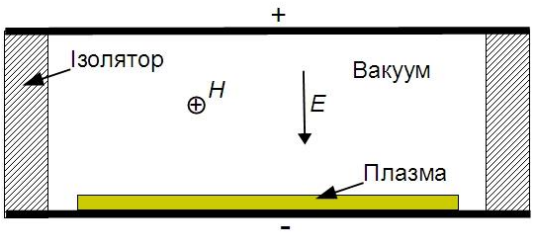



*Рис.1. Конфігурація області.*

Найпростіша конфігурація розрахункової області приведена на рис. 1. Реально у більшості випадків конфігурація меж системи набагато складніша. Для досягнення прийнятного рівня флуктуацій полів (т. зв. «шумів») та точності розрахунків доводиться одночасно мати в області взаємодії велику кількість модельних електронів – типово $\geq 10^6$ для двовимірної моделі. Все це вимагає величезних обчислювальних витрат на кожен комп'ютерний експеримент.

В розглянутих задачах моделювання електронної емісії може додавати суттєві накладні витрати обчислень. Тому одна з головних вимог до будь-якого алгоритму моделювання емісії – його економічність.

Разом з тим, дуже важливо щоб модель емісії забезпечувала мінімально можливий рівень неоднорідності розподілу густини заряду та струму електронів для зменшення нефізичних шумів в електромагнітних полях. Виявляється що ці вимоги суперечать одна одній.

**Обчислювальна модель**

Конкретна реалізація довільного алгоритму емісії залежить від типу та обмежень повної моделі системи. Розглядалася така цільова модель. Тип моделі – «частинки в комірках» (particles in cells -PIC) розмірності 2D2V в прямокутній системі координат [2]. Сітка, в вузлах якої визначаються густина струмів та напруженості електромагнітних полів, прямокутна, рівномірна. Модельні електрони мають кінцевий розмір відповідний розміру комірок сітки. Метод обчислення густини струмів – зважування по площах [2]. Самоузгоджені миттєві електромагнітні поля обраховуються розв'язанням рівнянь Максвела за методом Йі [3]. Релятивістські рівняння руху електронів розраховуються методом Бориса [4].

Стосовно підсистеми моделювання електронної емісії – досліджувалось дві базових моделі. Перша передбачала моделювання плазмового факела в межах системи, як на рис. 1. Друга модель передбачала винесення плазмової області за межі області взаємодії. При цьому на нижній (прикатодній) стінці моделювався процес «витягання» електронів із плазмового факела під дією поперечного електричного поля. В межах кожної моделі можлива реалізація різних алгоритмів поповнення системи модельними частинками. Вони різняться в основному критеріями, за якими визначається необхідність поповнення.

**Методика та основні результати**

Особливістю роботи є неможливість прямої оцінки ефективності алгоритму без його тестування в складі програми моделювання всього приладу. Основними показниками ефективності алгоритму являлись рівень шумів напруженостей миттєвих електричних та магнітного полів пов'язані з шумовою складовою густини електронних струмів. З іншого боку для оцінки обчислювальних витрат покроково контролювались кількості інжектованих в систему електронів, поглинутих на катоді та тих, що знаходяться в безпосередній близькості до катоду. Таким чином контролювалась досягнутий рівень оптимальності алгоритмів з точки зору обчислювальної ефективності та мінімізації шумової складової електромагнітних полів,

Досліджувались алгоритми які реалізували одну з двох можливих стратегій. Перша і найпростіша, це інжекція «надлишку» електронів в розрахунку на те, що зайві будуть виштовхнуті назад на катод, в двох варіантах – з використанням локального критерію інжекції за порогової напруженості поперечного електричного поля або примусової інжекції по всій поверхні катода. При цьому виявилось що використання локального критерію необхідності інжекції приводить до більш великомасштабних флуктуацій густини струмів і може спричиняти розвиток в системі нефізичних нестійкостей. При примусовій інжекції по всій поверхні катода флуктуації густини струмів мають значно менші просторові масштаби і зменшуються із збільшенням кількості інжектованих на кожному кроці електронів. Без особливих труднощів можна забезпечити рівень шумів напруженостей полів меншу 1 % від напруженості зовнішнього поперечного поля. Недоліком даної стратегії є необхідність обчислювати рух великої кількості електронів в при катодній області.

Альтернативна стратегія полягає в інжекції мінімально необхідної кількості електронів таким чином що локальне значення напруженості поперечного поля в при катодній області і після інжекції чергового електрона не опускається нижче порогового. Цей підхід дає змогу обійтись меншою кількістю частинок але вимагає додаткових обчислень локальної напруженості поля. Крім того, при великих значеннях коефіцієнту укрупнення модельних електронів (ступеня дискретизації заряду) зберігається небезпека великомасштабних флуктуацій густини струмів.

Загалом можна зробити висновок що питання оптимальності довільного алгоритму інжекції суттєво залежить від параметрів та вимог повної моделі в котрій він працює.

ЛІТЕРАТУРА

1. Сливков И. Н. Процессы при высоком напряжении в вакууме. М.: «Энергоатомиздат», 1986, с.90-102.
2. Бэдсел Ч., Ленгдон А.. Физика плазмы и численное моделирование. М.: «Энергоатомиздат», 1989, 452 с.
3. Yee K. S., Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Ewuations in Isotropic Media, *IEEE Trans. Antennas Prop.*, 14 (1966)., p.p. 302-307.
4. Boris J. P. Relativistic plasma simulation-optimization of a hybrid code. *Proc. IV Conf. Num. Sim. Plasmas*, Naval Res. Lab., Wash. D.C., **3-67**, 2-3 Nov. 1970.
5. Горбань А.М., Лонин Ю.Ф.. «Влияние периодической неоднородности эмиссии электронов на спектр колебаний в генераторе на магнитоизолированной передающей линии». // *ВАНТ, Серия: "Плазменная электроника и новые методы ускорения"*, (68), 2010, № 4, с. 30-33.

**ТОТКАЛ Станіслав Олексійович** – студент; В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: stanislavtotkal@gmail.com; ORCID: 0000-0001-6112-8604.

Наукові інтереси:

– *комп'ютерне моделювання*
– *системи штучного інтелекту*
– *технології створення програмного продукту*
– *плазмові технології*

**УДК 004.458**

ЧЕРНЯЕВ И.Н., ЛАЗУРИК В.М.

# ИСПОЛЬЗОВАНИЕ GRAPHQL ДЛЯ РАБОТЫ С БАЗАМИ ДАННЫХ

**Введение**

Веб-разработка − самое востребованное направление в IT технологиях. Под Веб-разработкой понимают процесс создания веб-сайта или веб-приложения [1]. Современная веб-разработка гораздо шире, чем простое создание сайтов. Сайты, реклама, электронная коммерция, мобильные приложения, игры – не полный перечень областей использования веб технологий. Интернет постоянно развивается, поэтому важным является понимание того, что в Вебе актуально, что главенствует сегодня и будет востребовано завтра. Постоянно появляются новые языки программирования, инструменты, меняются фреймворки, меняется вёрстка, одни элементы заменяются другими. Следует отметить, что из-за огромного количества средств разработки часто бывает достаточно тяжело оценить целесообразность применения конкретных программных средств и инструментов. Опять-таки большое значение имеет мода на использование определенных инструментов разработки. И не всегда их выбор бывает оправданным. Традиционный протокол прикладного уровня HTTP/1 вытесняется HTTP/2 [2]. То, что было писком моды пару лет назад и остается востребованным и сейчас, подвергается сомнениям [3,4].

Работа посвящена рассмотрению GraphQL − языку запросов, используемого клиентскими приложениями для работы с данными. В работе уделено внимание особенностям использования GraphQL. На примере решения задачи учета динамики записи на курсы оценивается целесообразность использования этого программного обеспечения при реализации не высоконагруженного проекта, в котором не используются несколько баз данных. Оцениваются возможные проблемы и накладные расходы, связанные с применением GraphQL для решения поставленной задачи.

**История создания GraphQL**

GraphQL − это язык запросов и манипулирования данными с открытым исходным кодом для API (Application Programming Interface), а также среда для выполнения запросов с существующими данными. GraphQL был разработан внутри компании Facebook в 2012 году, а затем был опубликован в 2015 году [5]. В 2012 году разработчиками Facebook была начата работа по перестройке собственных мобильных приложений. Приложения для iOS и Android, которые постепенно становились все более сложными, были тонкими клиентами для просмотра мобильного веб-сайта. Они имели низкую производительность и часто зависали. Возникла также проблема доставки данных новостной ленты в мобильные приложения. Разработчиками были оценены разные возможности, включая ресурсы сервера RESTful и таблицы FQL (SQL-подобный API Facebook). Обнаружено было существенное различие между данными, которые требовалось использовать в приложениях, и запросами к серверу, которые им требовались. Так же было оценено большое количество серверного кода для подготовки данных и кода на клиенте для их анализа. Это привело к тому, что был начат проект GraphQL, который позволил бы переосмыслить выборку данных мобильного приложения с точки зрения дизайнеров и разработчиков продукта, и тем самым сместил бы акцент разработки на клиентские приложения.

**Особенности GraphQL**

**GraphQL** − это декларативная спецификация выборки данных и язык запросов для API. Он предназначен для обеспечения общего интерфейса между клиентом и сервером для выборки данных и манипуляций с более гибким подходом. Используя терминологию веб-разработчиков, можно сказать, что GraphQL серверное решение для Frontend. Его можно использовать как в веб- интерфейсе, так и в Backend приложениях. В то время как REST, используемый для

построения распределенных масштабируемых веб-сервисов, для получения больших наборов данных, объединяющих связанные ресурсы, вынужден работать с несколькими конечными точками, GraphQL работает с одной конечной точкой. Актуальность использования такого подхода определяется отсутствием необходимости несколько раз запрашивать данные. GraphQL может принимать сложные запросы, а затем преобразовывать вывод данных в ту форму, которая требуется клиенту. С использованием GraphQL можно решить проблему избыточного или недостаточного извлечения данных. Разработчики сосредоточены на том, *что* хотят получить, а не *как* это сделать.

Практически, GraphQL – это слой, который живет между клиентом и одним или несколькими источниками данных (рис.1), получая клиентские запросы и выбирая необходимые данные в соответствии с инструкциями, заданными разработчиком [6,7].

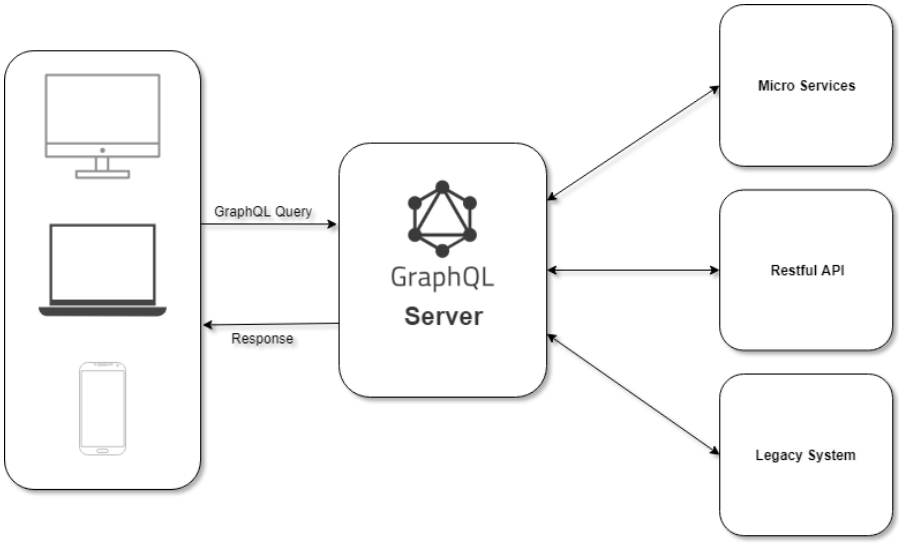



*Рис.1- Схема взаимодействия GraphQL сервера*

Работа с сервером GraphQL всегда начинается с разработки схемы. Она состоит из двух взаимосвязанных объектов: TypeDefs и Resolvers. TypeDefs − типы, определенные пользователем, и корневые. Корневые типы:

- Query –запросы (аналог GET в REST) для получения необходимых данных с сервера.
- Mutation – мутации используются для создания, обновления и удаления данных.
- Subscriptions – подписки могут быть использованы для установки и сохранения связи с сервером в режиме реального времени, результат отправляется клиенту каждый раз, когда на сервере происходит определенное событие.

```
type Book {
  id: ID
  title: String
  publishDate: Date
  price: String
  author Author
}
```

```
type Author {
  id: ID
  firstName: String
  lastName: String
  books: [Book]
}
```

```
type Query {
  book(id: ID!): book
  author(id: ID!): Author
}
```

*Рис.2- Список типов, доступных для взаимодействия с GraphQL*

Типы, определяемые пользователем – объекты, с которыми работают запросы. Поля – характеристики объекта. Все в схеме должно быть типизировано. Например, если работаем с информацией о книгах и их авторах, то целесообразно создать два пользовательских типа Book и Author и корневой тип Query (рис.2). В GraphQL описание отдельного ресурса не связано со способом его получения. Запрос на получение информации, использующий объекты, показанные на рис.2 может выглядеть, например, как на рис.3.

```
Execute Query
query {
  book(name: "clean-code") {
    name
    authors {
      name {
        books {
          name
        }
      }
    }
  }
}
```

*Рис.3- Запрос в GraphQL*

Resolver или распознаватель − функция, которая возвращает данные для определённого поля. Распознаватели возвращают данные того типа, который определён в схеме. Они должны быть определены для всех корневых типов, для всех пользовательских и для полей, которые могут использоваться как параметры в запросе. GraphQL организует данные в граф, используя один интерфейс. Объекты представлены узлами (определенными с использованием схемы), а связь между узлами представлена ребрами в графе. Каждый объект поддерживается распознавателем, который обращается к данным сервера. Когда сервер GraphQL отвечает на запрос конечного пользователя, он начинается с корня запроса, и resolver заполняет каждое поле запрошенного объекта. Пример ответа на выполненный запрос представлен на рис. 4.

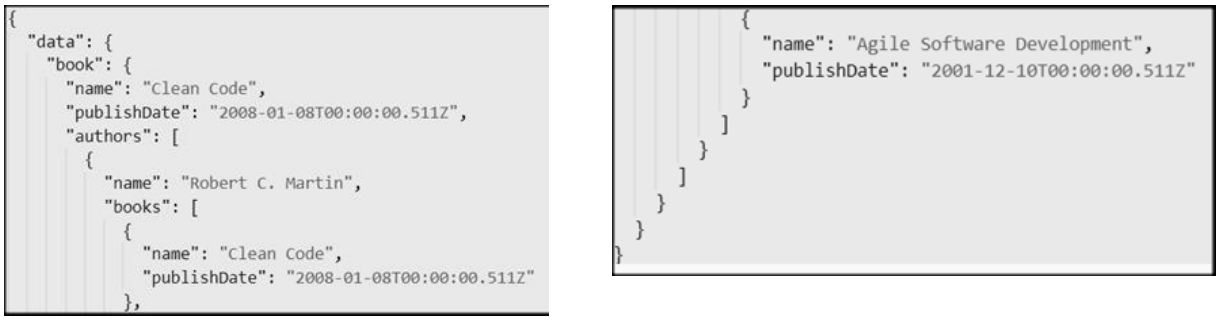



*Рис.4- Результат выполнения запроса в GraphQL*

**Экосистема GraphQL**

После широкого применения вокруг GraphQL стала создаваться богатая экосистема. Она растет не только горизонтально, распространяясь на другие языки программирования, но и вертикально – поверх GraphQL надстраиваются различные библиотеки, инструменты и сервисы, которые делают разработку с использованием GraphQL еще доступней и прозрачней.

**Клиентская часть:**

- Relay: мощный клиент GraphQL, хорошо оптимизированный для производительности.
- Appolo Client: клиент GraphQL для всех основных платформ разработки, поддерживает различные фронт-енд фрейморки (React, Angular и Vue) и платформы (iOS, Android).

**Серверная чать:**

- Graphql.js: реализация , предназначенная для запуска GraphQL в среде Node.js.
- Graphql-tools: пакет, который позволяет создавать готовую к производству схему GraphQL, используя язык схемы GraphQL.
- Appolo-server: серверная библиотека Node.js GraphQL, поддерживающая Express, Connect, Hapi, Koa и другие популярные HTTP-серверы Node, со встроенными функциями.

**Инструменты:**

- Graphiql**:** интерактивная среда просмотра в браузере.
- Graphcool**:** серверная часть GraphQL для приложений с мощным веб-интерфейсом для управления базой данных и хранимыми данными.

**Задача учета динамики записи на курсы**

Для задачи учета записи студентов на курсы рассмотрим имплементацию на Node.js, используя GraphQL и MongoDB. Архитектура приложения показана на рис.5.

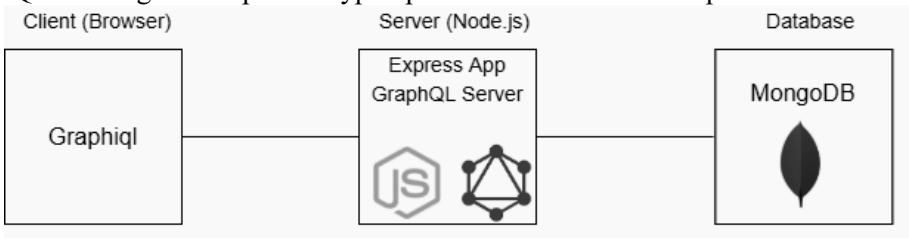



*Рис.5- Архитектура системы*

Схема, созданная для этой задачи, содержит запрос для получения данных, мутации для обновления данных и два типа Teacher и Course (рис.6).

```
const CourseType = new GraphQLObjectType({
  name: "Course",
  fields: () => ({
    id: { type: GraphQLID },
    name: { type: GraphQLString },
    description: { type: GraphQLString },
    students_quantity: { type: GraphQLInt },
    status: { type: GraphQLString },
    price: { type: GraphQLInt },
    teacher: {
      type: TeacherType,
      resolve(parent, args) {
        return Teacher.findById(parent.teacherId);
      },
    },
  }),
});
```

*Рис.5- Объявление типа Course*

```
const TeacherType = new GraphQLObjectType({
  name: "Teacher",
  fields: () => ({
    id: { type: GraphQLID },
    fio: { type: GraphQLString },
    contact_info: { type: GraphQLString },
    salary: { type: GraphQLInt },
    email: { type: GraphQLString },
    scope: { type: GraphQLList(GraphQLString) },
    courses: {
      type: new GraphQLList(CourseType),
      resolve(parent, args) {
        return Course.find({ teacherId: parent.id });
      },
    },
  }),
});
```

*Рис.6- Объявление типа Teacher*

Объявление типов осуществляется с помощью *GraphQLObjectType*. В свойстве fields определяются поля. Функция *resolve* указывает действие, которое должно выполниться при запросе, а именно «Выбрать преподавателя и курс по его уникальному идентификатору». На рис.7 приведен код объявления типа запроса, где определяются конечные точки типа read-only в схеме.

Этим конечным точкам задается определенный тип, чтобы при выполнении запроса был доступ к полям этого типа. Код запроса приведен на рис.8. Для отсылки запроса серверу используем интегрированную среду разработки Graphiql.

```
const RootQuery = new GraphQLObjectType({
  name: "RootQueryType",
  fields: {
    course: {
      type: CourseType,
      args: { id: { type: GraphQLID } },
      resolve(parent, args) {
        return Course.findById(args.id);
      },
    },
    teacher: {
      type: TeacherType,
      args: { id: { type: GraphQLID } },
      resolve(parent, args) {
        return Teacher.findById(args.id);
      },
    },
    courses: {
      type: new GraphQLList(CourseType),
      resolve(parent, args) {
        return Course.find({});
      },
    },
    teachers: {
      type: new GraphQLList(TeacherType),
      resolve(parent, args) {
        return Teacher.find({});
      },
    },
  },
});
```

*Рис.7- Объявление главного типа запросов*

В запросе указываем только те поля, которые хотим получить. Предположим, необходимо получить информацию о преподавателях – их Ф.И.О., контактную информацию, название и цену курсов, которые они ведут. Результат выполнения запроса может быть таким, как показано на рис.9.

Возможность добавления нового курса реализуется с помощью интегрирования мутации в схему GraphQL. Мутация содержит тип, аргументы, которые не являются типом *null, resolve* функцией записываем курс в базу данных и возвращаем записанную запись (рис.10) .

```
{
 teachers {
   fio,
   contact_info
   courses {
     name,
     price
   }
 }
}
```

*Рис.8- Пример запроса*

```
{
  "data": {
    "teachers": [
      {
        "fio": "Thomas Green",
        "contact_info": "thomas@green.com",
        "courses": [
          {
            "name": "JS Fullstack",
            "price": 3000
          },
          {
            "name": "Ruby",
            "price": 4000
          }
        ]
      },
      {
        "fio": "Ben Brown",
        "contact_info": "ben@brown.com",
        "courses": [
          {
            "name": "PHP",
            "price": 4000
          }
        ]
      }
    ]
  }
}
```

*Рис.8- Получение данных*

```
const Mutation = new GraphQLObjectType({
  name: "Mutation",
  fields: {
    addCourse: {
      type: CourseType,
      args: {
        name: { type: new GraphQLNonNull(GraphQLString) },
        status: { type: new GraphQLNonNull(GraphQLString) },
        price: { type: new GraphQLNonNull(GraphQLID) },
        teacherId: { type: new GraphQLNonNull(GraphQLID) },
      },
      resolve(parent, args) {
        let course = new Course({
          name: args.name,
          status: args.status,
          price: args.price,
          teacherId: args.teacherId,
        });

        return course.save();
      },
    },
  },
});
```

*Рис.10-Создание мутации*

При вызове любого рода мутации можно задать набор возвращаемых полей. В нашем случае, после записи в базу данных получаем имя курса и информацию о преподавателе.

**Заключение**

В работе описаны, и на примере решения задачи учета записавшихся на курсы, проанализированы некоторые возможности использования GraphQL. Рассмотрены статьи и электронные ресурсы, в которых присутствует как восторженное отношение к этому средству веб-разработки, так и откровенно негативные отзывы. Резюмируя свой небольшой опыт и высказывания признанных авторитетов можно сделать вывод, что применение GraphQL не всегда бывает оправдано. Если задача большая и многофункциональная, в ее разработке участвует большой коллектив специалистов, информация хранится во многих разнообразных источниках, крайне важна скорость разработки и производительность реализованного программного решения – тогда , несомненно GraphQL.

Для того, чтобы решить в каждом конкретном случае нужен ли GraphQL, необходимо учесть много разных факторов. В случае решения задачи, рассмотренной в работе, проще напрямую обращаться к базе данных, поскольку применение GraphQL для этой не сложной задачи не оправдано, оно требует внесения промежуточного слоя между клиентом и сервером, накладные расходы при этом серьезные.

Согласно [7] необходимо оценить является ли разрабатываемый проект приложением с множеством пользователей, или приложением, обрабатывающим огромные объёмы данных, когда использование GraphQL улучшит производительность его клиентской части, или вполне оправдан выбор проверенной временем технологии REST. Стоит помнить, что GraphQL не является заменой REST, это решение проблемы, с которой люди сталкиваются с помощью REST. GraphQL, как и любая другая технология, приносит свои проблемы и накладные расходы. Так же необходимо учитывать цену или выгоду от замены REST на GraphQL. Если приложение не предоставляет свои бэкэнд-сервисы для использования в нескольких интерфейсах, польза GraphQL тоже сомнительна.

ЛИТЕРАТУРА

1. Веб-разработка. [Электронный ресурс] Режим доступа: https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0
2. Mark Nottingham. How Multiplexing Changes Your HTTP API. [Электронный ресурс] Режим доступа: https://www.mnot.net/blog/2019/10/13/h2_api_multiplexing
3. Marc-André Giroux. Is GraphQL Still Relevant in an HTTP2 World? [Электронный ресурс] Режим доступа: https://medium.com/@__xuorig__/is-graphql-still-relevant-in-an-http2-world-64964f207b8
4. Esteban Herrera. 5 reasons you shouldn’t be using GraphQL. [Электронный ресурс] Режим доступа: https://blog.logrocket.com/5-reasons-you-shouldnt-be-using-graphql-61c7846e7ed3/?gi=f67074d77004
5. Lee Byron. GraphQL: A data query language. [Электронный ресурс] Режим доступа: https://engineering.fb.com/core-data/graphql-a-data-query-language/
6. Alexei Medvedev. Время для GraphQL-изации. [Электронный ресурс] Режим доступа: https://blog.maddevs.io/%D0%B2%D1%80%D0%B5%D0%BC%D1%8F-%D0%B4%D0%BB%D1%8F-graphql-%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8-2a31a6ac667c
7. Ryan Glover. Подробности о GraphQL: что, как и почему. [Электронный ресурс] Режим доступа: https://habr.com/ru/company/ruvds/blog/445268/

**ЧЕРНЯЕВ Игорь Николаевич** – студент факультета компьютерных наук Харьковского национального университета имени В. Н. Каразина.

Научные интересы: – *разработка программного обеспечения.*

**ЛАЗУРИК Валентина Михайловна** – старший преподаватель кафедры искусственного интеллекта и программного обеспечения факультета компьютерных наук Харьковского национального университета имени В. Н. Каразина.

Научные интересы: – *разработка компьютерных систем для моделирования процессов в радиационных технологиях; организация баз данных.*

**УДК 004.7**

ЧІСТОВ А.І., МОРОЗ О.Ю.

# МОДЕЛЬ КОМП'ЮТЕРНОЇ СИСТЕМИ З ГОЛОСОВИМ УПРАВЛІННЯМ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ ARDUINO

**Вступ**

На сьогодні в світі технології розвиваються небаченим темпом. У людей залишається все менше часу на виконання тих чи інших завдань. Те що здавалося ще не давно могла виконати людина, сьогодні делегується на виконання до різних систем, програм і технологій. Одним з прикладом такої технології є Arduino.

Фірма Arduino що знаходиться в Італії, полегшує процес роботи з мікроконтролерами, дає ряд переваг перед іншими пристроями за допомогою простого та зрозумілого середовища програмування, малої ціни та безліччю плат розширення. Для науковців та студентів платформа Arduino може стати основним елементом для дослідження і вирішення завдань в галузі автоматизації та робототехніки.

Це плата невеликих розмірів з власними пам'яттю і процесором. На платі присутні кілька десятків контактів, до яких можна підключити безліч компонентів: лампочки, датчики, мотори, монітори, магнітні дверні замки, роутери та все, що працює від електрики.

У процесор Arduino можна завантажити програму, яка буде керувати всіма цими пристроями за заданим алгоритмом. Таким чином можна створити нескінченну кількість універсальних і унікальних гаджетів, зроблених власноруч і з власного задуму. Популярність Arduino отримала завдяки своїй доброзичливості а також простоті. Програми для Arduino пишуться на звичайному C ++, доповненим простими і зрозумілими функціями для керування вводу / виводу на контактах. Для зручності роботи з Arduino існує безкоштовне офіційне середовище програмування Arduino IDE, що працює під Windows, Mac OS і Linux. За допомогою нього завантаження нової програми в Arduino стає справою одного кліку, якщо підключити плату до комп'ютеру через USB. Хоча для більш допитливих можлива робота і через Visual Studio, Eclipse, інші IDE або командний рядок, а новачкам підійде візуальне середовище програмування XOD IDE.

Метою даної роботи є розробка системи з голосовим управлінням на технології Arduino, яку при успішній реалізації можна рекомендувати до використання клієнтові.

**Постановка задачі**

Клієнту необхідна комп'ютерна система за допомогою якої можна управляти електричними приладами голосом за допомогою мобільного пристрою.

Комп'ютерна система складається з плати Arduino Uno, модуля Bluetooth (HC-05), двох-канального реле-модулю (SRD-5VDC-SL-C), резисторів, перемичок і макетної плати.

На цей проект виділяються певні людські, часові та фінансові ресурси, а також пред'являються певні вимоги до якості.

Для створення оптимального виконавчого пристрою необхідно кількісно та якісно оцінити можливі варіанти використаних технологій.

**Суть моделі комп'ютерної системи з голосовим управлінням**

На плату Arduino Uno за допомогою офіційного додатку розробимо і запишемо скетч, який буде відповідати за включення / вимикання пристроїв за допомогою голосу. Плату необхідно з'єднати перемичками з модулем Bluetooth. Цей модуль відповідає за бездротове підключення виконавчого пристрою до мобільного пристрою.

Також необхідно підключити перемичками до плати Arduino Uno двоканальне реле (SRD-5VDC-SL-C) за допомогою якого можна підключити до електричних пристроїв, якими і належить управляти. Даний модуль містить два канали реле фірми SONGLE модель SRD-05VDC-SL-C, перемикання здійснюється за допомогою напруги 5В. Схематично модуль спеціально розроблений для управління за допомогою слабкострумових плат, таких як arduino,

які на виході можуть видати струм не більше 40 мА, так само для захисту долучено оптопару EL817, яка реалізує гальванічну розв'язку. При необхідності щодо збільшення числа електроприладів, двоканальне реле можна замінити восьмиканальним і підключити до восьми пристроїв.

На мобільний пристрій слід встановити додаток через який приладами можна управляти як голосом так і звичайним натисканням.

У тому числі присутні макетна плата і постійні резистори на 1 кОм і на 2 кОм. Контакт Rx на Bluetooth платі вимагає 3,3 вольт, а контакт Tx на платі Arduino працює з напругою в 5 вольт. Для того щоб привести їх до одного значення скористаємося резисторами.

Приклад реалізації комп'ютерної системи представлений у вигляді схеми (рис.1)

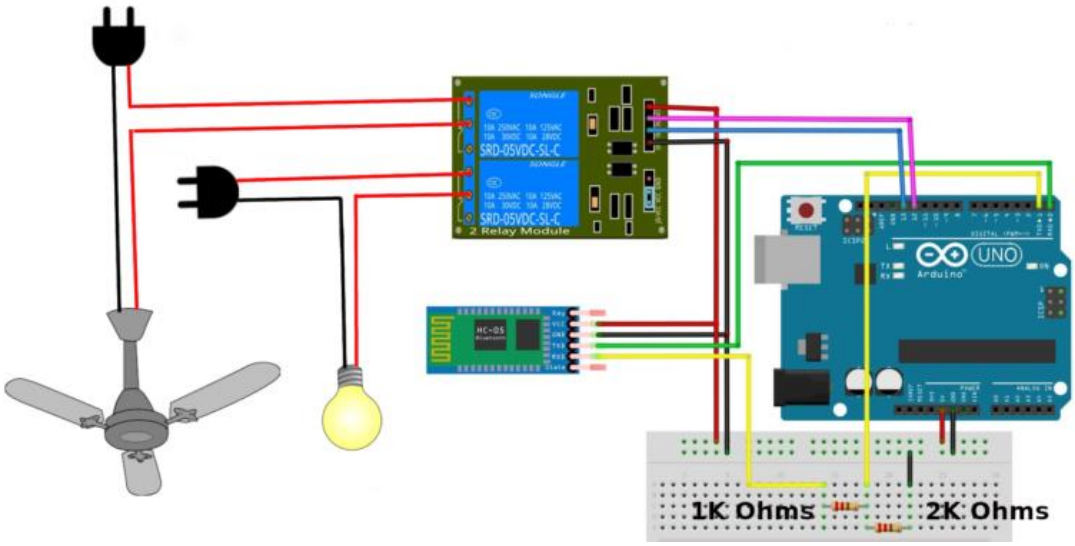



*Рис. 1 Приклад реалізації комп'ютерної системи*

ЛІТЕРАТУРА

1. Петин В.А. Вивчаємо Практична енциклопедія Arduino: навч. посіб. Москва: ДМК Пресс, 2017.
2. Петін В.А. Проекти з використанням контролера Arduino. 2-е изд. - СПб .: БХВ-Петербург, 2015
3. Іго Т. Arduino, датчики і мережі для зв'язку пристроїв. 2-е видання - СПб .: БХВ-Петербург, 2015
4. Макаров С. Л. Arduino Uno и Raspberry Pi 3. От схемотехники к интернету вещей. ДМК Пресс, 2019. 202с.

**ЧІСТОВ Артем Ігорович** – студент групи КУ-41 факультету комп'ютерних наук; Харківський національний університет імені В. Н. Каразіна, майдан Свободи 6, Харків, Україна, 61022; e-mail: artyom12213@gmail.com; ORCID: 0000-0002-8490-9021.

Наукові інтереси:

- *Програмування. Управління проектами. Алгоритми і структури даних.*

**МОРОЗ Ольга Юріївна** – старший викладач кафедри теоретичної та прикладної системотехніки факультету комп'ютерних наук; Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 6, Харків, Україна, 61022; e-mail: o.moroz@karazin.ua; ORCID: 0000-0002-4920-4093.

Наукові інтереси:

- *Технології автоматичного проектування паралельних програм.*

**УДК 681.5.01**

ШАРАПА О.В., БЕРДНІКОВ А.Г.

# МОДЕЛЬ СИСТЕМИ УПРАВЛІННЯ РЕЖИМАМИ РОБОТИ ТЕПЛИЧНОГО ГОСПОДАРСТВА АГРОПРОМИСЛОВОГО КОМПЛЕКСУ

**Введення**

Україна — аграрна країна, аграрний сектор має колосальне значення для економіки в цілому, є основою добробуту громадян.

У 2018 році агропромисловий сектор формував близько 13% українського ВВП. Частка продукції сільського господарства в загальному експорті України за цей період склала 39,8%, або рекордні $ 18,8 млрд. Сільське господарство як завдяки об'єктивному розвитку, так і внаслідок падіння інших галузей, зміцнилося в статусі донора бюджету і є поряд з інформаційними технологіями (IT) - одним з провідних секторів української економіки.

Оскільки тепличне господарство є важливою частиною аграрного сектору, то системи автоматизованого управління режимами роботи тепличного господарства, її мікрокліматичні показники, конструкторські та експлуатаційні показники є важливим фактором підвищення продуктивності виробництва. Актуальним є модель системи управління режимами роботи тепличного господарства, її мікрокліматичні показники, конструкторські та експлуатаційні фактори.

Тепличне господарство — це підприємство, що вирощує в закритому ґрунті овочі або розсаду для відкритого ґрунту. Основним об'єктом кожного тепличного господарства є теплиця — спеціальне культиваційне приміщення з покриттям зі світлопроникного матеріалу (скло, поліетиленова плівка). Обігрів рослин у теплицях — сонячний, біологічний (шляхом біопалива), технічний (гаряча вода, пара, електроенергія, теплові відходи промислових підприємств тощо) та геотермальний (тепло гарячих джерел). У теплицях вирощують головним чином тепличні культури — овочеві (огірки, томати, перець, салат, хрін, петрушка, редис, цибуля, селера, капуста), баштанні (кабачки, дині, кавуни), плодові (лимон, персик), ягідні (полуниці, суниці), декоративні (гвоздика, хризантема, троянда та ін.) та гриби.

Теплиці дуже економічно ефективні: затрати на їх будівництво окупаються через 4-5 років. В Україні тепличне господарство у переважній своїй більшості зосереджено навколо великих міст.

**Постановка задачі**

Врожай у теплицях значною мірою визначається мікрокліматом, який забезпечують за допомогою сучасних комп'ютерних систем управління.

Для оцінювання енергоефективності теплиць використовують показник "енергомісткість", який характеризує рівень витрат паливно-енергетичних ресурсів на одиницю виробленого продукту. Основними проблемами невисокої енергоефективності теплиць є: надмірне споживання енергоресурсів через низьку енергоефективність технологій і обладнання та недосконаліть схем через завищену частку споживання природного газу і недостатній обсяг використання енергії з альтернативних видів палива та відновлювальних джерел; низький рівень управління енергоефективністю і споживанням енергоресурсів.

Підвищити енергоефективність теплиць можна такими заходами, як: зменшення обсягу технологічних і невиробничих втрат енергоресурсів внаслідок модернізації схем енергопостачання, обладнання, впровадження сучасних енергоефективних технологій; оптимізації структури споживання паливно-енергетичних ресурсів, зокрема заміщення традиційних видів енергоресурсів іншими видами, в тому числі з відновлювальних джерел енергії та альтернативних видів палива; вдосконалення системи управління енергоефективністю і споживанням енергоресурсів.

Основна проблема розроблення інтелектуальних компонентів систем управління мікрокліматом у теплицях полягає у формуванні вимог, виборі методів опрацювання даних і засобів їх реалізації (програмні, апаратні чи програмно-апаратні). Тому актуальною проблемою

є проєктування інтелектуальних компонентів та синтез на їх основі адаптивних комп'ютерних систем управління мікрокліматом та енергоефективністю теплиць.

**Суть моделі**

В основу проєктування сучасних систем управління мікрокліматом теплиць покладено системну інтеграцію, яка ґрунтується на системному підході, який охоплює всі рівні інтеграції процесів управління теплицею з врахуванням вимог та ефективності їх застосування. Розробляти системи управління мікрокліматом теплиці доцільно на основі комплексного підходу, який охоплює комунікаційні та інформаційно-управляючі технології та системи, сучасну елементну базу, програмне забезпечення з використанням ОС Android, засоби підтримки прийняття рішень і ґрунтується на таких принципах: системності, змінного складу обладнання, відкритості, модульності та використання комплексу базових проєктних рішень.

У дипломній роботі показано, що сучасні системи управління розвиваються в напрямку зменшення розмірів, маси та енергозатрат, підвищення надійності, нарощення функціональності та інтелектуалізації. В таких умовах з'являються нові системи управління, які ґрунтуються на нових інтелектуальних інформаційних технологіях і сучасній елементній базі. Після проведеного аналізу елементної бази та інших показників виявлено, що запропонована плата Arduino на базі мікроконтролера ATmega2560 є ефективною та не поступається конкурентам.

Управляють мікрокліматом у теплиці за допомогою системи управління, основними компонентами якої є: мікроконтролерна система, мобільний пристрій зв'язку під управлінням ОС Android, датчиків і активаторів (рис 1.)


*Рис. 1 Структурна схема управління мікрокліматом теплиці*

Мікроконтролерна система складається із плати Arduino на базі мікроконтролера ATmega2560, Ethernet модуля, модуля пам'яті та модуля інтерфейсів.

Мікроконтролер з'єднаний з усіма модулями системи в єдину систему. Він відповідає за автономне управління мікрокліматом теплиці. З цією метою мікроконтролер зчитує через модуль інтерфейсів показники датчиків та згідно із внутрішньою програмою керує активаторами.

Внутрішня програма аналізує межі необхідних параметрів мікроклімату та формує керівні сигнали. У разі наявності зв'язку із мобільним пристроєм управління мікроконтролер перевіряє, чи наявні зміни у параметрах управління теплицею та перезаписує конфігураційний файл.

Розробляючи та використовуючи інтелектуальні компоненти у системі управління можна ефективно управляти мікрокліматом та енергоефективністю тепличного господарства. Проведення дослідження експлуатації доводить підвищення ефективності даної моделі.

**Висновок**

Істотними тенденціями сучасного сільськогосподарського виробництва є, з одного боку, постійне зростання його масштабів, підвищення кількості і якості сільськогосподарських продуктів, з іншого – прогресивний дефіцит робочої сили, непопулярність монотонної та важкої фізичної ручної праці в рільництві й тваринництві. Найважливішим, а часто і єдиним засобом вирішення розбіжностей між ними є комплексна механізація й автоматизація виробництва.

Завдяки механізації й автоматизації різко зростає продуктивність праці. Питання комплексної автоматизації мають велике народногосподарське значення, тому що їх впровадження гарантує економічний ефект.

Автоматизація сільськогосподарського виробництва підвищує надійність і продовжує термін роботи устаткування, полегшує й оздоровляє умови праці, підвищує безпеку праці і робить його престижнішим, скорочує текучість робочої сили та економить затрати праці, збільшує кількість і підвищує якість продукції, прискорює процес стирання відмінностей між працею розумовою і фізичною, промисловою і сільськогосподарською.

ЛІТЕРАТУРА

1. Денисенко В. В. Компьютерное управление технологическим процессом, экспериментом, оборудованием. — М.: Горячая линия Телеком, 2009. — 408 с.
2. Автоматизація технологічних процесів і системи автоматичного керування: Навчальний посібник /Барало О.В., Самойленко П.Г.,Гранат С.Є., Ковальов В.О. – К.: Аграрна освіта, 2010. – 3-4 с.
3. Тепличне господарство: Geograf.com.ua [Електронний ресурс] // Режим доступу: http://www.geograf.com.ua/glossary/suspilno-geografichni-terminy/teplichne-gospodarstvo
4. Arduino Mega 2560: kosmodrom.com [Електронний ресурс] // Режим доступу: http://www.kosmodrom.com.ua/el.php?name=MEGA2560-R3-NO
5. Сільське господарство України: https://uk.wikipedia.org/ [Електронний ресурс] // Режим доступу: https://uk.wikipedia.org/wiki/Сільське_господарство_України

**ШАРАПА Олександр Вадимович** – студент, студент 5-го курсу кафедри теоретичної і прикладної системотехніки, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків, Україна, 61022; e-mail: sasha10ok64@gmail.com,10sasha@email.ua; ORCID: 0000-0003-0284-2802.

Наукові інтереси:

*– автоматизація систем управління, штучний інтелект, технологія блокчейн.*

**БЕРДНІКОВ Анатолій Георгійович** – к. т. н., доцент; доцент кафедри теоретичної і прикладної системотехніки, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків, Україна, 61022; e-mail: a.berdnikov@karazin.ua;

Наукові інтереси:

*– застосування методів теорії ухвалення рішень при управлінні складними телекомунікаційними системами.*

**УДК 004.7(075)**

ШАРОВ В.О., БЕРДНИКОВ А. Г.

# МОДЕЛЬ ПОМЕХОУСТОЙЧИВОГО КАНАЛА ПЕРЕДАЧИ ДАННЫХ

**Введение**

В современных системах управления широко применяются интегральные цифровые сети (Integrated Services Digital Network – ISDN), которые обеспечивают передачу всех видов дискретной информации в стандартном формате.

В качестве примеров сетей с интегрированным обслуживанием можно привести универсальные протоколы ATM (Asynchronous Transfer Mode — асинхронный режим передачи) или Broadband ISDN (B-ISDN), которые обеспечивают передачу любого типа трафика, как компьютерного, так и мультимедийного.

Технология ATM используется на многих магистралях крупнейших операторов связи поддерживает все виды трафика, может использоваться непосредственно прикладным уровнем протоколов и работать автономно без протоколов TCP/IP и TCP/UDP, однако наличие достаточно сложного собственного протокола маршрутизации привело к тому, что в современных компьютерных сетях для объединения сетей используются, как правило, протоколы TCP/IP и TCP/UDP, но ATM остается одной из технологий, на основе которой работают многие локальные сети, образующие сложную составную сеть. Следовательно, совершенствование и развитие протокола с целью его дальнейшего применения и изучения представляет интерес, и разработка модели помехоустойчивого канала передачи данных, используемого в интегральной сети, представляется актуальной.

**Постановка задачи**

Основной особенностью эффективного функционирования интегрального канала является решение задачи кодирования при применении больших кадров, когда начинают проявляться такие нежелательные эффекты, как ожидание кадров в очередях и задержка пакетизации.

Время ожидания кадра в очереди можно уменьшить, если обслуживать кадры чувствительного к задержкам трафика в приоритетной очереди. Однако даже при придании чувствительным к задержкам кадрам высшего приоритета обслуживания в коммутаторах время ожидания компьютерного пакета может все равно оказаться недопустимо высоким.

Задержка пакетизации представляет собой это время, в течение которого первый замер, например, голоса ждет момента окончательного формирования пакета и отправки его по сети. При передаче голоса замеры делаются через одинаковые интервалы времени с частотой 8 КГц, то есть через каждые 125 мкс, а каждый замер кодируется одним байтом данных. В результате, при большом размере кадра время задержки может оказаться недопустимо большим для передачи голосовой информации (т.е. информации реального времени). Таким образом, кадр ATM (ячейка) размером в 53 байта, включающий заголовок в 5 байт и поле данных 48 байт, явился результатом компромисса между требованиями эластичного и чувствительного к задержкам трафиков.

Повышение помехозащищенности заголовка протокола ATM обеспечивается контрольной суммой, которая вычисляется с помощью корректирующего кода Хэмминга, исправляющего все однократные ошибки и обнаруживающего двукратные ошибки.

Анализ статистики ошибок показывает, что для небольших и фиксированных кадров, передаваемых на высокой скорости, наиболее вероятными являются именно однократные и двукратные ошибки.

Для повышения эффективности изучения технологии ATM стоит задача разработать модель помехоустойчивого канала передачи данных, реализующего помехоустойчивый код Хэмминга, имеющий в передаваемом блоке 8 разрядов (байт).

Вероятности появления этого типа ошибок определяются в соответствии с выражениями на следующих формулах:

$$P_1 = n * P_{ош} * (1 - P_{ош})^{n-1}, (1)$$
$$P_2 = C_n^2 * P_{ош}^2 * (1 - P_{ош})^{n-2}, (2)$$

где $P_1$ – это вероятность единичной ошибки в кодовой комбинации, а $P_2$ – вероятность двукратной ошибки в кодовой комбинации.

При подсчете, если брать за основу вероятность искажения одного разряда при передаче в канале $10^{-3}$, то при основе в один байт вероятность единичной ошибки будет составлять 0,0079, чуть менее одного процента. Если посчитать для двукратной, то вероятность ошибки существенно упадет до 0,00002. В случае трехкратной – вероятность ошибки составляет 0,00000005. В дальнейшем, убывание происходит в геометрической прогрессии.

Из анализа приведенных выражений следует, что именно однократные и двукратные ошибки являются самыми опасными при передаче данных, а ошибки большей кратности столь маловероятны, что учитывая наши ресурсы ими можно пренебречь.

**Суть модели**

За основу модели взят каскадный код, первой ступенью которого является известный классический код Хэмминга (7,4), обеспечивающий исправление однократной ошибки, а второй ступенью является расширенный код Хэмминга (8,4), позволяющий исправлять однократную ошибку и обнаруживать двукратную ошибку.

Код (7,4) означает, что при 7 разрядах мы имеем 4 разряда информационных. Принцип построения такого кода базируется на определенной последовательности проверочных (П) и информационных (И) разрядов, приведенных в таблице № 1.

*Табл.1*

| Номер позиции | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Назначение позиции | П 1 | П 2 | И 1 | П 3 | И 2 | И 3 | И 4 |

Система порождающих уравнений выглядит таким образом:

$$П1 = И1 \oplus И2 \oplus И3$$
$$П2 = И2 \oplus И3 \oplus И4$$
$$П3 = И1 \oplus И2 \oplus И4$$

Где “$\oplus$” – суммирование по модулю 2.

Для проверки в коде Хэмминга (7,4) будет использоваться матрица, представленная в таблице № 2, где “*” – это номера разрядов, который суммируются по модулю 2.

*Табл.2*

| № позиции<br>№ проверки | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | * | | * | | * | | * |
| 2 | | * | * | | | * | * |
| 3 | | | | * | * | * | * |

В таком случае, система порождающих уравнений будет выглядеть следующим образом:

$$1\ п.р. = 1\ р.к. \oplus 3\ р.к. \oplus 5\ р.к. \oplus 7\ р.к.$$
$$2\ п.р. = 2\ р.к. \oplus 3\ р.к. \oplus 6\ р.к. \oplus 7\ р.к.$$
$$3\ п.р. = 4\ р.к. \oplus 5\ р.к. \oplus 6\ р.к. \oplus 7\ р.к.$$

Где “п.р.”– проверочный разряд, а “р.к.” – разряд в закодированной комбинации.

При такой проверочной матрице, если в полученной комбинации будет искажен один разряд, то в результате проверки будет получена проверочная комбинация из трех проверочных разрядов,

указывающая на номер позиции, где произошло искажение. Эти комбинации указаны в таблице № 3.

*Табл.3*

$$H(7,4)=\begin{array}{c} \text{№столбцов} \\ \left\|\begin{matrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{matrix}\right\| \end{array} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix}$$

Если будет искажен первый разряд, то будет получена проверочная комбинация 001, если второй, то будет получена проверочная комбинация 010 и так далее.
Для примера можно взять кодирование числа 12. Оно записывается в двоичном коде как 1100.
Пользуясь таблицей № 1 и приведенной ниже ее системе порождающих уравнений, кодируем комбинацию 1100 и получаем комбинацию 0111100. По назначению разрядов она будет иметь вид, приведенный в таблице № 4.

*Табл.4*

| № позиции | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Назначение позиций | п | п | и | п | и | и | и |
| Код числа | 0 | 1 | 1 | 1 | 1 | 0 | 0 |

Допустим, в момент принятия мы получили комбинацию 0111000.
Тогда, в результате проверки по таблице № 2, у нас сформируется проверочная комбинация 101, которая в двоичном коде нам указывает, что ошибка находится на 5 позиции, следственно, мы легко можем откорректировать код.

Возвращаясь к модели, если использовать каскадный код, то после кодирования первой ступени, а именно кода Хэмминга (7,4), мы приступаем к кодированию второй ступени – кода Хэмминга (8,4).
Для того, чтобы исправлять однократную ошибку и обнаруживать двукратную ошибку, необходимо добавить еще один бит, что будет 8-й позицией, а именно бит проверки на чётность. Его порождающее уравнение имеет вид:

П4=П1+П2+И1+П3+И2+И3+И4.

В таблице № 5 приведена расстановка по позициям.

*Табл.5*

| Номер позиции | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Назначение позиции | П1 | П2 | И1 | П3 | И2 | И3 | И4 | П4 |

Проверочная матрица имеет вид, представленный в таблице № 6.

*Табл.6*

$$H(8,4)=\begin{Vmatrix}1&0&1&0&1&0&1\\0&1&1&0&0&1&1\\0&0&0&1&1&1&1\\1&1&1&1&1&1&1\end{Vmatrix}$$

Допустим, мы закодировали ту же комбинацию (7,4) 0111100 – число 12, что есть первая ступень нашего каскадного кода. При добавлении бита проверки на четность комбинация приобретает вид 01111000 – вторая ступень каскадного кода.
Если искажаются, к примеру, 2 и 4 позиции, то бит проверки на чётность ошибки не выявит, тогда мы его отбрасываем, что есть отбрасывание второй ступени, и декодируем комбинацию уже по известному нам алгоритму(7,4), что есть первая ступень нашего каскадного кода. В результате проверки сформируется проверочное число 110, что укажет на ошибку, но поскольку проверка на четность ошибки не выявила, то можно утверждать, что ошибка не одиночная, а двойная. В таком случае целесообразно будет послать запрос для повторного отправления данной комбинации.

**Вывод**
Разработанная модель может быть использована в учебном процессе при изучении возможностей помехоустойчивых кодов Хэмминга, протоколов асинхронных сетей и демонстрации работы интегрального канала передачи данных.

ЛИТЕРАТУРА

1. Олифер В. Г., *Олифер* Н. А. 0-54 Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 4-е изд. — СПб.: Питер, 2010
2. Лосев Ю. І. Основи *теорії* передачі інформації : навчальний посібник / Ю. І. Лосев, С. І. Шматков ; за ред. Ю. І. Лосева. – Х. : ХНУ імені В. Н. Каразіна, 2013.
3. Интернет ресурс : *https*://studfile.net/preview/5857828
4. Интернет ресурс : https://ru.wikipedia.org/wiki/ATM
5. С. С. Владимиров., Сравнение вероятностных характеристик 8-разрядных кодов с прямой коррекцией ошибок, Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича, Санкт-Петербург, 193232
6. Лит.:Гойхман Э.Ш. Передача информации в АСУ / Э.Ш. Гойхман, Ю.И. Лосев

**ШАРОВ Владислав Олегович** – студент группы КУ-41 факультету компьютерных наук; Харьковский национальный университет имени В.Н. Каразина, майдан Свободы, 4, Харків-22, Украина, 61022; e-mail: WyCTPiY@gmail.com; ORCID: **0000-0003-3152-0650**.
Научные интересы:
- *программирование;*
- *управление проектами.*

**БЕРДНІКОВ Анатолій Георгійович** – к. т. н., доцент; доцент кафедри автоматизації та комп'ютерно інтегрованих технологій; Харківський національний університет ім. В.Н. Каразіна; м. Харків, майдан. Свободи, 6, 61022; e-mail: a.berdnikov@karazin.ua .
Наукові інтереси:
*– оптимізація робіт по керуванню проектами.*
*– проектування автоматизованих систем управління;– організація менеджменту при управлінні проектом.*

**УДК 519.6,51-76**

ШАЦКИЙ К.В., ЯНОВСКИЙ В.В.

# КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ РАСПРОСТРАНЕНИЯ И ЭВОЛЮЦИИ ИДЕЙ В ОБЩЕСТВЕ

**Введение**

Эволюция популяций является областью, вызывающей неизменный интерес исследователей. Понимание проблемы возникновения кооперативного поведения в популяциях интенсивно исследуются в последние десятилетия (см. например ). Основой для этого служит теория игр и, в частности, дилемма заключенных [1], которая определяет взаимодействие агентов популяции. Обнаружено несколько механизмов возникновения и поддерживание кооперативного поведения в популяциях. Большое внимание уделяется изучению структурированных популяций, на решетках [2], сложных сетях [3]. При этом учитываются различные факторы влияющие на действие индивидуума. В качестве таких факторов рассматриваются: наказание [4], сходство [5], добровольное участие [6] и т. п.

В этой работе рассматривается эволюция популяции стратегий, распределенных по определенному числу носителей. В результате «общения» $N$ носителей, обладающих исходно каждый $n$ стратегиями происходит изменение стратегий носителей. Правила замены стратегий определяется победой стратегии при соревновании. Проигравшая меняется на выигравшую. Набор очков эволюционных преимуществ при соревновании зависит от выбора определенной матрицы выплат. В качестве стратегий носителей, рассматриваются стратегии разной глубины памяти [7]. Глубина памяти характеризует количество ходов, которые способна запомнить одна стратегия. Чем больше глубина, тем больше вариативность ходов, что позволяет увеличивать сложность стратегий. Каждая стратегия может либо сотрудничать, либо отказываться от сотрудничества. Соревнование осуществляется согласно правилам итерированной дилеммы заключенных.

Носители взаимодействуют между собой попарно и используют все имеющиеся у них стратегии во время взаимодействия. Поочередно каждый выбирает одну свою стратегию, которая еще не использовалась. Носитель, стратегия которого проиграла, заменяет её на выигравшую стратегию оппонента. Взаимодействие продолжается, пока у кого-либо не закончатся не взаимодействовавшие стратегии. Для взаимодействия выбираются случайные пары носителей, пока все множество носителей не будет разделено на пары. Взаимодействие между таким набором пар считается соответствующим этапом эволюции. После окончания этапа происходит следующий этап. Все повторяется до выхода популяции носителей в стационар. На каждом этапе эволюции изучены основные характеристики популяции и установлена их зависимость от времени. Роль дискретного времени играет номер этапа эволюции. Показано, как меняется сложность стратегий носителей, глубина памяти и разнообразие стратегий носителей.

Максимальная рассмотренная глубина памяти стратегий в популяции равна 2 и используется полный набор всех вариантов стратегий, что позволяет не упустить возможного лидера. Популяция с максимальной памятью 2 включает стратегии с памяти 0, 1 и 2.

**Стратегии с памятью**

Будем считать стратегией правила, по которым осуществляется ход при знании хода противника [8]. Глубина памяти характеризуется числом ходов оппонента, о которых «помнит» стратегия. Память глубины 0 означает, что стратегия делает ход наблюдая только текущий ход. Под ходом будем понимать кооперацию либо отказ (1 и 0). Стратегия определяется описанием ответов на все возможные хода. Соответственно, для глубины памяти 0 возможными ходами противника могут быть только 0 или 1. В Таблице 1 приведены ответы стратегии на соответствующий ход, при этом установив в лексикографическом порядоке записи возможных ходов, строку ответов считать именем стратегии, которое однозначно и определяет действие стратегии.

*Табл. 1 Пример ответов стратегии с глубиной памяти 0*

| Возможный ход | 0 | 1 |
|---|---|---|
| | ⇓ | ⇓ |
| Ответ стратегии | 0 | 0 |

Стратегия глубины памяти 1 делает ход зная не только текущий ход, но помня и предыдущий. В этом случае число возможных ходов увеличивается Имя стратегии глубины памяти 1, как и правило ее действия, определяется четырьмя символами. В случае глубины памяти $k$ имя стратегии – последовательность длины

$$S_k = 2^{k+1}. \quad (1)$$

Однако, если противник делает всего один ход, то возникает ситуация неполноты данных. Поэтому надо включить в стратегию первый ход, и стратегию с меньшей памятью. Полный вид стратегия глубины памяти 1 может выглядеть как [0](10)(0001). Нетрудно подсчитать, число стратегий памяти 1 – 128, а глубины памяти $k$ как

$$N_k = 2^{2^{k+2}-1}. \quad (2)$$

Таким образом, каждая стратегия характеризуется глубиной памяти. Еще одна характеристика стратегий её сложность. Для заданной стратегии последовательностью 0 и 1 рассмотрим ее как функцию номера значения. Тогда сложность равна порядку производной этой функции равной 0. Это отвечает естественным представлениям о полиномах: чем выше степень полинома, тем сложнее его поведение. Например, для стратегии 00 сложность равна 0, для 11 сложность равна 1, для 10 и 01 сложно равна 2. Еще одна характеристика стратегий носителя это число разных стратегий у носителя. Сгруппировав стратегии внутри носителя по сложности или глубине памяти можно посчитать их количество и пропорциональное соотношение.

**Распределение стратегий у носителей в популяции**

На начальном этапе носителям присваивается определенное число стратегий. Распределение стратегий по носителям происходит случайно и без повторений внутри носителя. Сначала выбирается какой глубины памяти будет стратегия, затем выбирается индекс стратегии внутри выбранной глубины памяти.

**Взаимодействие носителей и их стратегий**

Сначала все носители случайным образом разделяются на пары между которыми будет происходить взаимодействие. Пары перемешивают все свои стратегии и последовательно выбирают стратегии для взаимодействия без возврата, т.е. каждая стратегия взаимодействует только один раз. Соревнование между парой стратегий происходит 100 раз для уменьшения зависимости от первого хода. Проигравшая стратегия заменяется на выигравшую, если её нет у носителя, иначе только удаляется проигравшая.

После взаимодействия всех пар заканчивается этот этап эволюции. Дальше, на следующем этапе, всё повторяется с носителями уже с измененным набором стратегий.

Определение какая стратегия победила а какая проиграла происходит по сумме очков после взаимодействия пары. Очки начисляются согласно матрице выплат и в работе был использован вариант Аксельрода [9].

**Моделирование взаимодействия**

Известно, что для общества, в котором каждый носитель имеет одну стратегию, эволюционно выгодными являются стратегии большей сложности и глубины памяти [7]. Рассмотрим теперь случай, когда каждый носитель обладает большим числом стратегий.

Пусть существует 50000 человек и каждый имеет по 50 стратегий разной глубины памяти до 2 включительно. После распределения среди стратегий носителей присутствовали все возможные стратегий (32904 варианта). Далее моделировалась эволюция такой популяции носителей. На каждом этапе эволюции определяются основные свойства стратегий носителей. Эволюция прекращается после выхода в стационарное состояние.

В результате моделирования установлена зависимость сложности стратегий среднего носителя (рис. 1). Сначала чаще побеждали стратегии большей сложности, после чего с 2 поколения наступил примитивный период, который длился 5 поколений. После достижения минимума начался монотонно увеличение сложности, которое длилось до 22 поколения и выхода на стационарное значение.

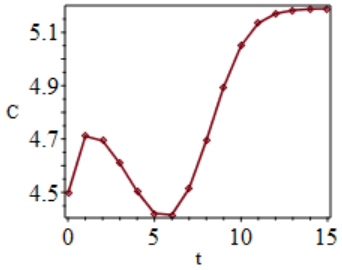



*Рис. 1 Зависимость сложности стратегии среднего носителя от времени*

Рассматривая разделение стратегий по глубине памяти показано, что сложность случайно выбранной стратегии каждой глубины уменьшается, и чем меньше глубина, тем сильнее убывание (рис 2). Несмотря на это, существуют периоды развития . Это происходит из-за изменения соотношения внутри носителя стратегий разной сложности и глубины памяти.

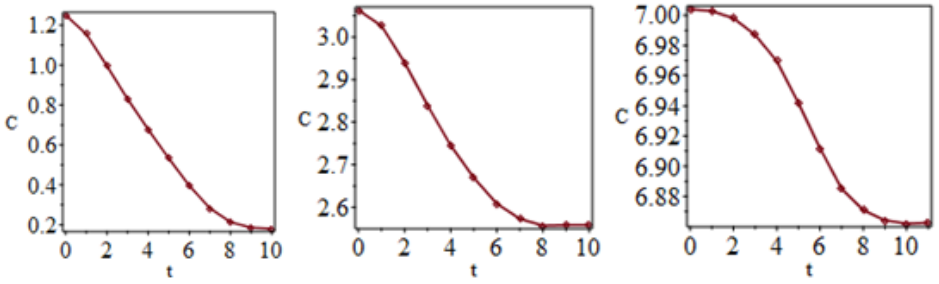



*Рис. 2 Зависимость сложности стратегии среднего носители от времени. Слева для глубины памяти 0, посередине для глубины памяти 1, справа для глубины памяти 2.*

Характер взаимодействия создает условия, при которых количество стратегий у среднего носителя неизбежно будет уменьшаться: изначальные 50 стратегий к концу эволюции достигает стационарной численности 17 (рис. 3). Сгруппировав стратегии внутри носителей по глубине памяти и отобразив их пропорциональной соотношение можно заметить убывание стратегий нулевой глубины и рост стратегий с большей глубиной памяти (рис. 3). То есть стратегии, которые знают больше предыдущих ходов оппонента, чем сам оппонент, имеют преимущество.

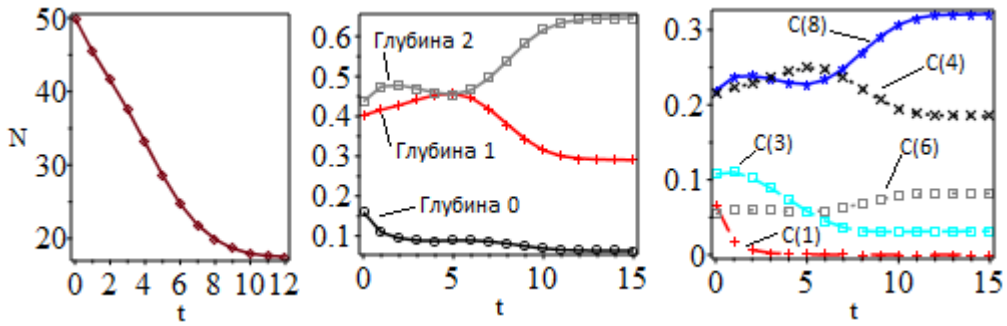



*Рис. 3 Зависимость количества стратегий у носителя от времени. Слева изменение общего числа со временем. Посередине пропорциональное соотношение, сгруппированное по глубине памяти. Справа пропорциональное соотношение, сгруппированное по сложности.*

Сгруппировав стратегии внутри носителей по сложности легко заметить явное преимущество самых сложных стратегий сложности 8 (рис. 3). Из-за изначально разного

количества стратегий разной сложности имеет смысл наблюдать за формой графика. Тогда можно заметить, что у больших сложностей график возрастающий, и в стационаре превышает значение на начальном этапе эволюции. Таким образом, более сложные стратегии имеют эволюционное преимущество. Если определенные качества стратегии являются эволюционно выгодными, то логично предположить полное исчезновение менее выгодных стратегий из популяции. Действительно, количество разных стратегий в популяции со временем уменьшается с 32904 до 2019. Так, например, уменьшение по глубине памяти стратегий с глубиной «0» из 8 осталось в стационаре 2, с глубиной «1» из 128 осталось 16, с глубиной «2» из 32768 осталось 2001.

**Выводы**

В результате моделирования взаимодействия общества индивидов, где каждый обладал определенным набором стратегий, а проигравшая стратегия сменялась выигравшей, были установлены общие закономерности эволюции такого общества. Установлено, что стратегии большей сложности и глубины памяти имеют эволюционное преимущество, разнообразие стратегий популяции падает, а набора стратегий среднего носителя уменьшается.

Показано существование так называемого «примитивного периода», при котором сложность общества падает: каждый носитель приобретает больше примитивных стратегий, чем сложных. После примитивного периода наступает период развития, где стратегии большей сложности и глубины побеждают и заменяют примитивные.

ЛИТЕРАТУРА

1. A. Rapoport and A. M. Chammah, Prisoner’s Dilemma // Univ.of Michigan Press, Ann Arbor, 1965.
2. Szab'o G., Hauert C., Phase transitions and volunteering in spatial public goods games / G.Szab'o, C.Hauert // Phys. Rev. Lett. -2002. - v.89, P.118101.
3. Szab’o G., F’ath G. Evolutionary games on graphs / G.Szab’o, G.F’ath // Phys Rep.-2007. - 446, P.97–216.
4. Hauert C. et al. Via freedom to coercion: the emergence of costly punishment // science. – 2007. – Т. 316. – №. 5833. – С. 1905-1907.
5. Traulsen A., Claussen J. C. Similarity-based cooperation and spatial segregation // Physical Review E. – 2004. – Т. 70. – №. 4. – С. 046128.
6. Szabó G., Hauert C. Evolutionary prisoner’s dilemma games with voluntary participation // Physical Review E. – 2002. – Т. 66. – №. 6. – С. 062903.
7. В.М.Куклин, А.В.Приймак, В.В.Яновский. Память и эволюция сообществ, Вісник Харківського національного університету імені В. Н. Каразіна, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління» т.35, с.38-60, 2017.
8. В.М.Куклин, А.В.Приймак, В.В.Яновский. Влияние памяти на эволюцию популяций, Вісник Харківського національного університету імені В. Н. Каразіна, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління» т.29, с.41-66, 2016.
9. R.Axelrod. The evolution of cooperation, Basic Books, New York (1984).

**ШАЦКИЙ Кирилл Витальевич** – студент кафедры искусственного интеллекта и программного обеспечения, Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 4, Харьков-22, Украина, 61022; e-mail: sckir04@gmail.com; ORCID: 0000-0001-9128-6830.

**ЯНОВСКИЙ Владимир Владимирович** – д. ф.-м. н., профессор; профессор кафедры искусственного интеллекта и программного обеспечения, Харьковский национальный университет имени В. Н. Каразина, площадь Свободы, 4, Харьков-22, Украина, 61022; e-mail:yanovsky@isc.kharkov.ua; ORCID: 0000-0003-0461-749X.

**УДК 004.75.**

ШВИДКИЙ Ю.К.

# РОЗПОДІЛЕНА ОБРОБКА ІНФОРМАЦІЇ В РЕЖИМІ РЕАЛЬНОГО ЧАСУ ЗА ДОПОМОГОЮ APACHE KAFKA

**Вступ**

Поняття "real-time processing" та "stream processing" є ключовими при обробці великих масивів даних та означають що інформація обробляється по мірі надходження або обробка не перевищує часового періоду, призначеного для прийняття рішення бізнес-користувачем. Сьогодні, в еру "Big Data", ці терміни все дедалі частіше з'являються у вимогах до програмного продукту та інтегруються до великих розподілених програмних систем. Це є досить очікувано бо з розвитком таких систем як "IoT", "VR", "Self-Driving Car" задачі аналізу інформації в режимі реального часу становляться актуальним, де кожен байт даних може бути важливим та визначати подальші дії. Ці зміни також впливають і на архітектуру систем бо розробникам доводиться отримувати, агрегувати, аналізувати терабайти або навіть петабайти даних які надходять з великої кількості різних сервісів. За цією причини, у сфері обробки інформації, старіші, традиційні методи планування щоденних оновлень як "batch processing" з Apache Hadoop[1] змінюється на "real time processing" з Apache Kafka технологію яка може досить сильно спростити життя розробникам.

**Мета статті**

Метою статті є опис Apache Kafka платформи та її вплив на розв'язання проблем розподільної обробки інформації у сучасних системах реального часу.

**Аналіз технології**

Apache Kafka був розроблений у компанії LinkedIn в 2011 році та написаний на Java, Scala. Даний продукт не стоїть на місці. Сьогодні Kafka - досить популярна open source платформа для обробки потоків повідомлень. Іншими словами Kafka - це publisher / subscriber черга[2], яка має низьку затримку, високу пропускну здатність та заснована на базі журналу коммітів – незмінна довгострокова упорядкована структура даних, в яку повідомлення можна тільки додавати. Серед ключових моментів платформи можна виділити:

- Розподільність. Kafka може працювати відразу на безлічі машин (вузлах), які утворюють цілісний кластер. Для кінцевого користувача це виглядає як єдиний вузол. Зберігання, отримання та розсилка повідомлень в Kafka таким чином дозволяє досягти високо доступність і відмовостійкість.
- Горизонтально масштабованість. Kafka здатна справлятися зі збільшенням робочого навантаження (збільшувати свою продуктивність) шляхом додавання кількості серверів (фізичних машин, вузлів).
- Відмовостійкість. Kafka спроектована таким чином, що через конфігурацію системи можна коригувати та підлаштовуватися під відмови системи. Це означає, що кластер Kafka з п'яти вузлів залишається працездатним, навіть якщо два вузли ляжуть.

**Навіщо потрібна Kafka ?**

Основне використання Kafka - у ситуаціях, коли додатки вимагають високої пропускної здатності для обробки потоку повідомлень, наявних при одночасному дотриманні необхідних затримок, доступності (відмовостійкості) та масштабованості.

Kafka зазвичай використовується для двох широких класів додатків:

- Побудова потокових конвеєрів повідомлень в режимі реального часу, для надійної комунікації даними між системами або додатками.
- Створення потокових додатків в реальному часі, які проводять задачі перетворення та аналізу потоку даних.

Якщо розглядати на прикладі простого додатку, де відбувається взаємодія двох сервісів то скоріше за все Kafka буде оверкілом для вашої системи. Якщо цей додаток з часом буде масштабуватися, ускладнюватися, додаватимуться нові сервіси, модулі, то архітектура може виглядати так.

Така система представлена на рисунку 1.

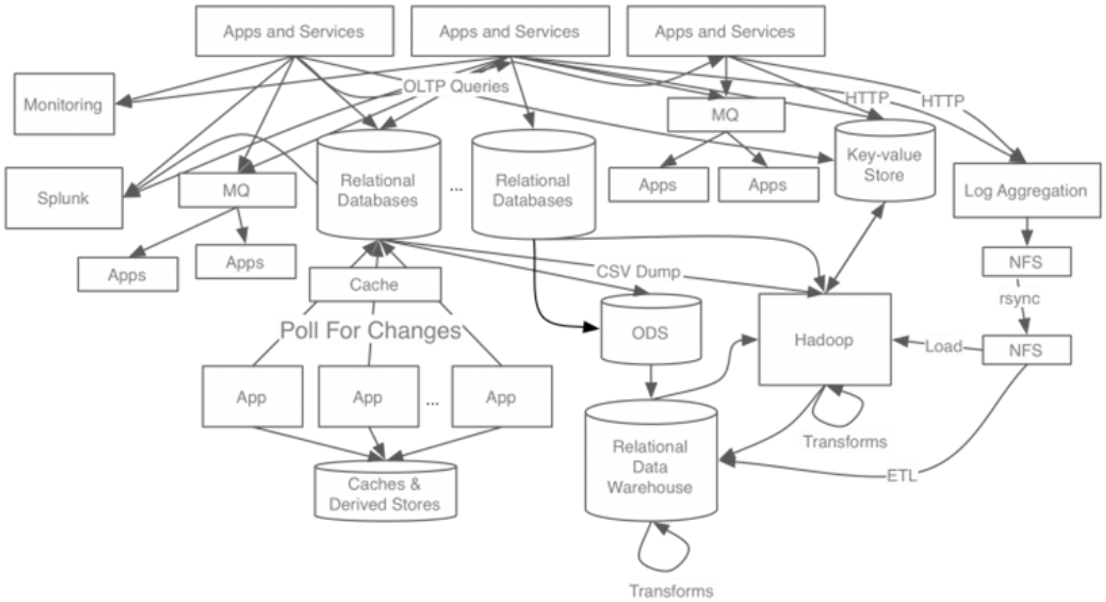



*Рисунок 1. Приклад сучасної складної системи.*

Підтримувати, розвивати та розширювати такий додаток стає все складніше і складніше. А також потрібно більше людино-ресурсів.

Kafka ж дозволяє будувати структуру системи як на рисунку 2.

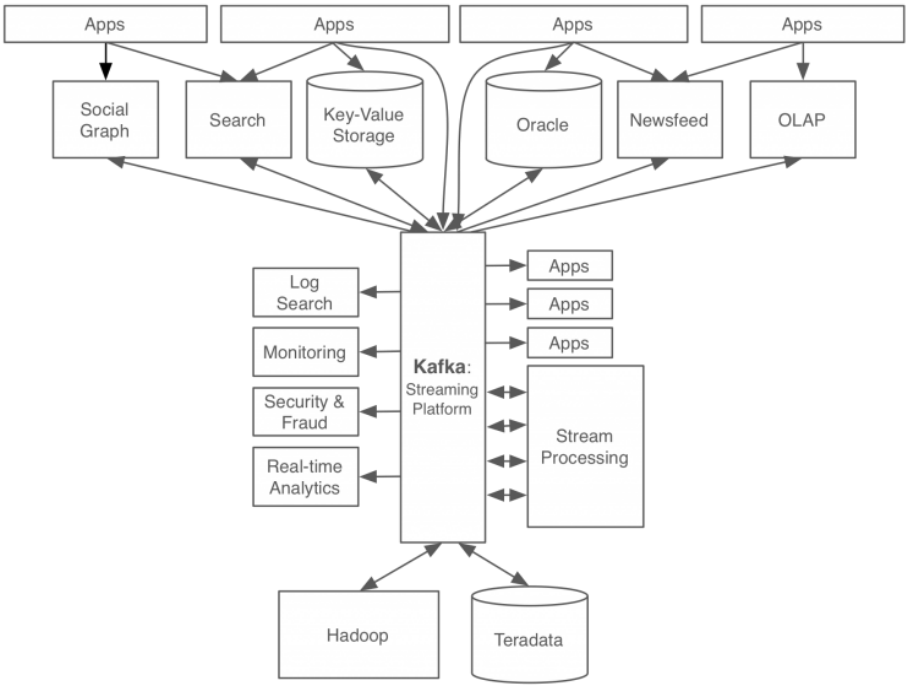



*Рисунок 2. Приклад системи з використанням Apache Kafka.*

Що в значній мірі полегшує розуміння загальної структури програми. Представлену архітектуру легше підтримувати та розширювати.

**Огляд Apache Kafka**

Загальна концепція досить проста. Виробники (Producer)[4] надсилають записи в брокер (один сервер в Kafka кластері)[4], який зберігає ці записи та передають їх споживачам (Consumer)[4]. Ключова абстракція в Kafka - це тема (Topic). Виробники публікують свої записи в тему, а споживачі підписуються на одну чи кілька тем. Тема – це категорія або назва каналу в якому зберігаються повідомлення. Властивості теми:

- Тема має ім'я. Це ім'я має бути унікальним в межах кластера;
- Тема може мати нуль, один або n споживачів;
- Тема розбита в упорядкований журнал коммітів – Partitions[4];
- Кожен Partition - зберігає тільки свої унікальні повідомлення;
- Кожному з повідомлень в Partition присвоюється унікальний Id який називається Offset;
- Повідомлення зберігаються протягом заданого періоду часу або по ліміту розміру займаного місця;

Повідомлення, відправлені одним виробником на конкретну партіцію теми, будуть збережені в послідовності відправлення. Споживачі прочитають повідомлення в такому ж порядку, як вони були збережені. Одна партіція в одній групі читається одним споживачем, це досить проста та зрозуміла схема яка дозволяє масштабування системи.

Якщо треба, щоб все працювало швидше, тоді збільшуємо кількість партіцій в темі та читачів в “Consumer group”.

**Consumer groups**

Кожний унікальний запис, збережений в тему буде доставлений тільки одному споживачеві в рамках групи споживачів. При цьому споживачі можуть бути в різних процесах або навіть на різних машинах. Якщо всі споживачі тільки в одній групі, то повідомлення будуть ефективно збалансовані між екземплярами споживачів. Якщо всі споживачі мають свою власну групу, то кожне повідомлення теми буде доставлено всім споживчим процесам.

«Logical Subscriber» - безліч екземплярів споживачів в одній групі. Така стратегія дозволяє:

- Додавати споживачів для забезпечення масштабованості та відмовостійкості;
- Кожен екземпляр користувача читає один або кілька розділів для теми;
- Не може бути споживачів більше, ніж розділів.

За рахунок «Logical Subscriber» стратегії розпаралелювання, все працює швидше. Партіції можуть поширюватися по кластеру машин, що дозволяє темі зберігати більше даних, ніж вміщається на машині.

На рисунку 3 представлено звичайний воркфлоу двух тем, виробників та групу споживачів:

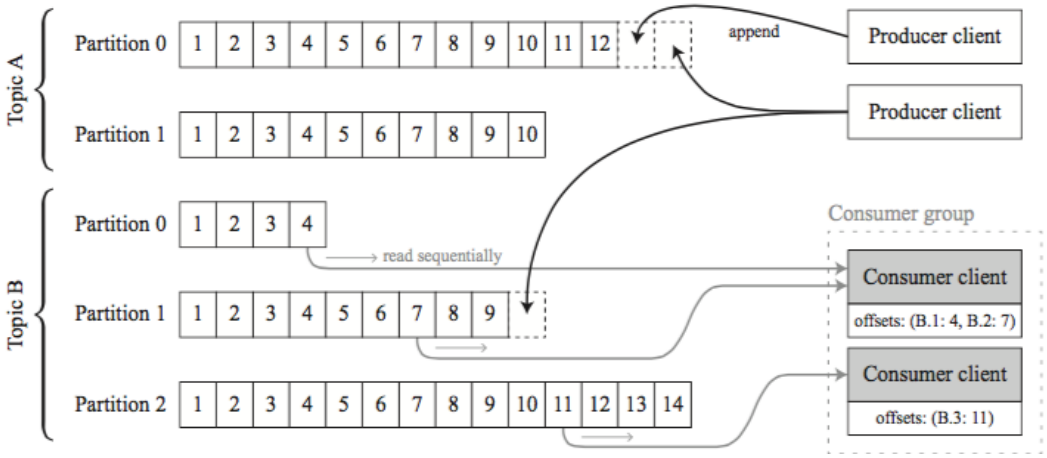



*Рисунок 3. Запис та читання повідомлень в темах.*

**Висновок**

Apache Kafka - це розподілена потокова платформа, що дозволяє обробляти трильйони подій в день. Цінність Kafka полягає у комбінуванні:

- черги повідомлень – розподілене транспортування даних.
- сховища - надійно зберігає дані в розподіленому та відмовостійких кластерах.
- платформи для обробки даних – завдяки Kafka Stream API[6] з'являється можливість обробки даних у "real-time" або "near real-time" режимі, не інтегруючи додаткові інструменти.

У цій статті була представлена базова семантика Kafka (producer, consumer, broker, topic). Кафка зберігає повідомлення в темах, які розділені та реплікуються між декількома виробниками в кластері. Виробники відправляють повідомлення на теми, з яких читають споживачі.

Kafka гарантує мінімальні затримки, високу пропускну здатність, надає відмовостійкі конвеєри, що працюють за принципом «публікація / підписка» та дозволяє обробляти великі потоки подій у режимі реального часу. Якщо звертати до уваги бенчмаркінг який проводив інженер та розробник платформи [7] в Kafka навіть з одним кластером в якому було піднято 3 Kafka брокера, обробка інформації досягає понад півмільйона повідомлень на секунду. Варто зазначити, що Кафку часто порівнюють із таким проектом з відкритим кодом як RabbitMQ[8].

ЛІТЕРАТУРА

1. Official Documentation Apache Hadoop. URL: https://hadoop.apache.org/ (Last accessed: 14.03.2020).
2. Message Queue Overview. URL: https://en.wikipedia.org/wiki/Message_queue (Last accessed: 14.03.2020).
3. Neha Narkhede. Kafka: The Definitive Guide: Real-Time Data and Stream Processing at Scale 1st Edition. O'Reilly Media, 2017. С. 15–50.
4. Kafka Official Documentation Apache. URL: https://kafka.apache.org/ (Last accessed: 09.03.2020).
5. Apache Kafka: обзор. URL: https://habr.com/en/company/piter/blog/352978/ (дата звернення: 09.03.2020).
6. Kafka Streams. URL: https://kafka.apache.org/documentation/streams/ (Last accessed: 10.03.2020).
7. Benchmarking Apache Kafka: 2 Million Writes Per Second (On Three Cheap Machines). URL: https://engineering.linkedin.com/kafka/benchmarking-apache-kafka-2-million-writes-second-three-cheap-machines (Last accessed: 10.03.2020).
8. Official Documentation RabbitMQ. URL: https://www.rabbitmq.com/ (Last accessed: 10.03.2020).

**ШВИДКИЙ Юрій Костянтинович** – студент кафедри штучного інтелекту та програмного забезпечення; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: yura.shvidkoy33@gmail.com; ORCID: 0000-0003-4473-7189.

Наукові інтереси:

*– розробка програмного забезпечення, аналіз великих даних.*

**УДК 004.452**

ШОФУЛ К.А. ЛАЗУРИК В.М.

# ИСПОЛЬЗОВАНИЕ МУЛЬТИМОДЕЛЬНОГО ПОДХОДА ПРИ ПРОЕКТИРОВАНИИ ПРИЛОЖЕНИЯ

**Постановка проблемы**

В информационных технологиях используется термин Big Data (большие данные) – это обозначение структурированных и неструктурированных данных огромных объёмов и различных инструментов, подходов и методов их обработки для конкретных задач и целей. Для больших данных выделяют традиционные определяющие характеристики, которые называются «Три V»: Volume − величина физического объёма, Velocity − скорость прироста и необходимости быстрой обработки данных для получения результатов, Variety − возможность одновременно обрабатывать различные типы данных. Именно разнообразие обрабатываемых данных приводит к необходимости решения задачи оптимального выбора языка программирования и системы управления данными. В настоящее время для обработки Big Data все чаще используются разные языки программирования и разные модели хранения данных в рамках одного программного приложения [1]. Такой подход получил название Polyglot Persistence [2]. Базы данных (БД) Polyglot Persistence используются, когда необходимо решить сложную проблему, разбив ее на сегменты и применяя различные модели баз данных.

В работе рассматривается мультифункциональный и мультимодельный подходы в рамках Polyglot Persistence баз данных. На примере проектирования системы управления взаимоотношениями с клиентами (CRM) рассматривается возможность использования мультимодельной системы управления базами данных (СУБД) MS SQL Server.

**Мультифункциональный подход**

Мартину Фаулеру, автору ряда известных книг, и одному из его соавторов Agile Manifesto, принадлежит название многовариантного хранения, когда используются разные БД, как Polyglot Persistence [3]. Ему же принадлежит и следующий пример организации хранения данных в полнофункциональном и высоконагруженном приложении в сфере электронной коммерции (рис.1).


*Рис. 1- Базы данных для задачи электронной коммерции*

На рис.1 приведен вариант мультифункционального проектирования, когда для разных сегментов приложения используются разные базы данных. При этом объем кода для хранения данных пропорционален числу используемых СУБД, как и объем кода для синхронизации данных. Практически невозможно в этом случае обеспечить транзакционность, возникают

проблемы с обеспечением масштабируемости, отказоустойчивости, высокой доступности каждой из используемых СУБД [4].

**Мультимодельные базы данных**

История реляционных баз данных (РБД) насчитывает около 40 лет. Поскольку РБД коммерческих предприятий продолжали развиваться с течением времени, они охватили несколько моделей данных и методов доступа в рамках единой системы управления базами данных. Эта концепция называется Multimodel Polyglot Persistence и позволяет многим приложениям использовать одну и ту же базу данных [5]. На рис. 2 представлен рейтинг систем управления базами данных на начало апреля 2020 года [6], выделены восемь наиболее популярных мультимодельных БД. Для некоторых из них показан перечень реализованных моделей данных, на первом месте в этом списке реляционная модель как изначальная.

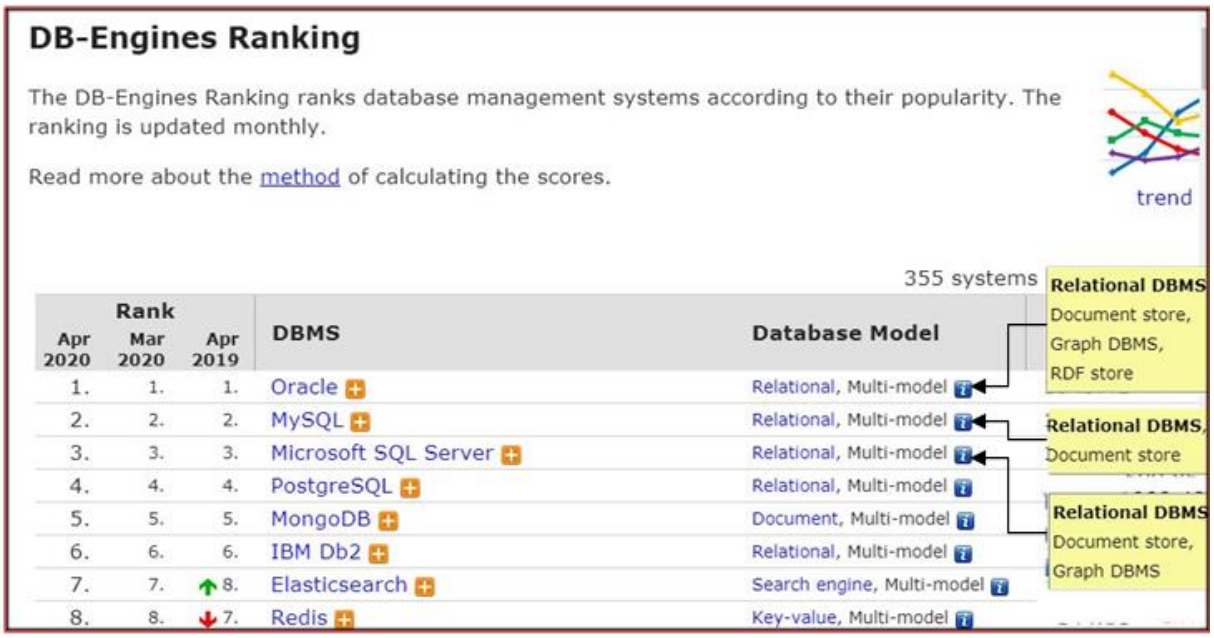



*Рис. 2- Рейтинг мультимодельных баз данных*

Возможность объединить несколько моделей в одну базу данных позволяет группам по информационным технологиям (ИТ) и другим пользователям удовлетворять различные требования приложений без необходимости развертывания различных систем баз данных. Сложность работы сокращается за счет использования одного хранилища данных, при этом обеспечивается непрерывная доступность и линейная масштабируемость.

**MS SQL Server**

Microsoft SQL Server [7] – популярная мультимодельная система управления данными, используется для работы с БД размером от персональных до крупных, масштаба предприятия. Исходной является реляционная модель. Основной язык запросов – Transact-SQL, который является реализацией структурированного языка запросов SQL с расширениями.

SQL Server поддерживает документную модель данных. JSON документы хранятся в обычных текстовых полях. Для работы с JSON используются операторы JSON_VALUE и JSON_QUERY. Для конструирования JSON документов из содержимого таблиц используется оператор FOR JSON PATH, а для того, чтобы разложить JSON по таблицам – OPEN JSON. Индексирование полей с JSON документами отсутствует, это делает затруднительными операции соединения таблиц по значениям этих полей и даже выборку документов по этим значениям. Конечно, такого рода модель хранения документов обеспечивает удобство разработки, но не быстродействия.

SQL Server предоставляет возможность работать с графовой моделью данных. Для моделирования графа используются таблицы специального вида для узлов и ребер. В

документации [8] приведен пример графа (рис.3), в котором персона, город, ресторан представляют собой узлы графа, и определены связи (ребра) между этими узлами. Таблицы создаются операторами CREATE TABLE AS NODE и CREATE TABLE AS EDGE. При создании таблиц NODE SQL Server формирует системное поле `$node_id`, представляющее собой уникальный в пределах базы данных идентификатор узла графа. Таблицы типа EDGE имеют системные поля `$from_id` и `$to_id` для определения связей между узлами. И, если значение `$node_id` формируется сервером, то значения `$from_id` и `$to_id` устанавливаются пользователем при заполнении таблицы данными. При этом может использоваться SQL оператор INSERT. Вид оператора для формирования ребра `livesIn` (проживает в) представлен в листинге 1. При этом значения системных полей `$node_id` запрашиваются с использованием конструкции `Insert ..Select`. Запрос на поиск людей, проживающих в городе `Seattle`, может выглядеть так, как представлено в листинге 2.

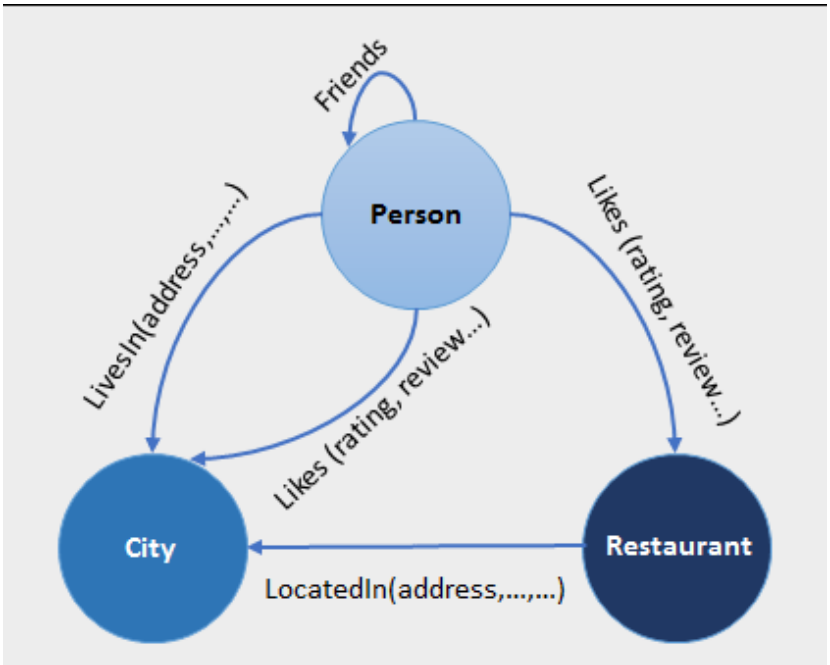



*Рис. 3- Граф: узлы – restaurant, city, person, ребра - LivesIn, LocatedIn, Likes.*

Листинг 1.

```
INSERT INTO Person VALUES (1,'John');
INSERT INTO Person VALUES (2,'Mary'); …
...
INSERT INTO City VALUES (2,'Seattle');
INSERT INTO City VALUES (3,'Redmond');
...
INSERT INTO livesIn VALUES ((SELECT $node_id FROM Person WHERE ID=1),
      (SELECT $node_id FROM City WHERE ID=1));
INSERT INTO livesIn VALUES ((SELECT $node_id FROM Person WHERE ID=2),
      (SELECT $node_id FROM City WHERE ID=2));
```

Листинг 2.

```
SELECT Person.name
FROM Person, livesIn, City
WHERE MATCH (Person-(livesIn)->City);
```

**Проектирование системы управления взаимоотношениями с клиентами**

Постановка задачи выглядит как обеспечение проектирования CRM (Customer Relationship Management) – системы управления взаимоотношениями с клиентами. Объекты: Заказчик, Исполнитель, Договор, Этап. Диаграмма вариантов использования представлена на рис.4. При проектировании можно выделить три отдельные подзадачи: оперативная работа с

договорами и этапами их выполнения (рис.5), хранение полных документов по договорам, установление взаимоотношений между исполнителями. Последняя задача вытекает из того момента, что у подрядчика (исполнителя) могут быть разные субподрядчики для выполнения отдельных операций по этапу договора. Если идти по пути использования различных СУБД для разных задач, то для оперативного отслеживания выполнения этапов договоров можно выбрать реляционную БД, например, MySQL. Для хранения документов − MongoDB, а для построения графа взаимоотношений между исполнителями графовую БД Neo4j.


*Рис. 4- Use case диаграмма приложения*

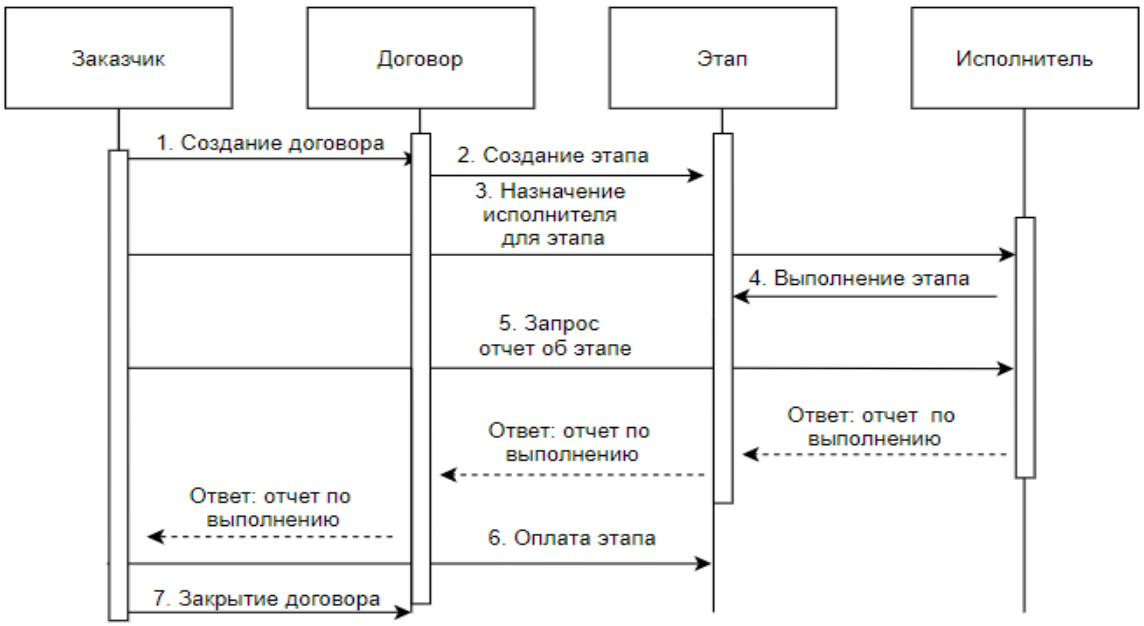



*Рис. 5- Диаграмма последовательности для сценария выполнения договора*

Но все эти задачи можно решить в рамках одной мультимодельной БД. Для этих целей можно использовать MS SQL Server. Схема БД приведена на рис.6. Таблица `Customer` описывает объект Заказчик. Таблица `Dovogor` описывает объект Договор, поле `customer_id` представляет собой внешний ключ (`foreign key`) к таблице `Customer`. Таблица `Stage` хранит данные по Этапу. Поля `dogovor_id` и `performer_id` представляют собой `foreign_key` к таблицам `Dovogor` и `Performer` (исполнитель) соответственно. Таблица `Pay` введена для отслеживания процесса оплаты по этапам договора. Все перечисленные выше таблицы (кроме `Performer` ) – реляционные. Отдельная таблица `AllDocs`, не представленная на схеме БД, будет создана специально для хранения как основных, так и сопутствующих документов по договорам. При этом будет использована документная модель данных SQL Server, хранение документов в JSON формате. Таблицы `Performer` и `Relation` реализуют граф взаимоотношений между исполнителями, `Performer` создана как NODE таблица и представляет собой узлы графа, `Relation` – таблица типа EDGE, представляющая собой ребра графа. Проектирование графа внутри БД с использованием графовой модели SQL Server дает возможность установить связь между таблицами `Stage` и `Performer`, формируя `foreign key` в таблице `Stage` к таблице узлов графа `Performer`.

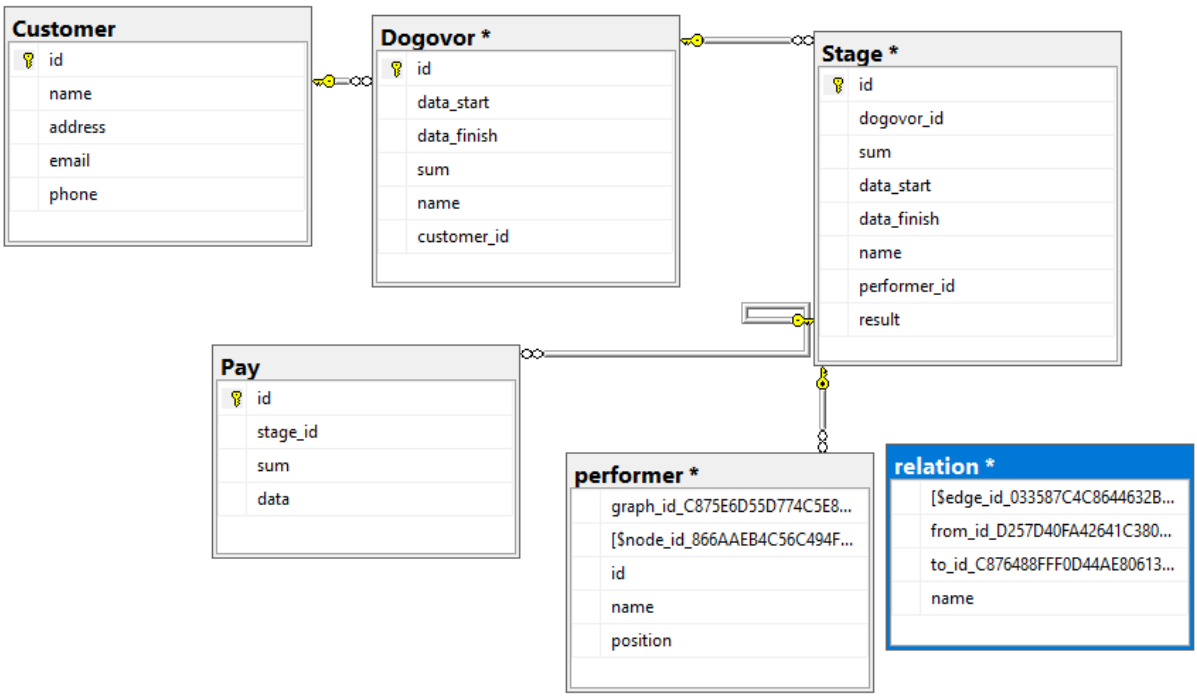



*Рис. 6- Схема базы данных для CRM*

**Заключение**

По результатам проведенного в работе исследования можно сделать вывод о целесообразности применения мультимодельных БД для решения разнообразных задач, в которых можно предусмотреть деление на отдельные сегменты с выбором оптимальной модели хранения данных в каждом из них. При таком подходе есть возможность избавиться от проблемы синхронизации данных и нелинейной доступности, облегчить проектирование и реализацию.

Для дальнейшего исследования интересен также вопрос об оптимальном выборе конкретной мультимодельной БД среди всех их представителей. При этом следует учитывать еще и такие факторы как стоимость программного обеспечения, наличие форума пользователей, наличие хорошей документации и другие важные критерии.

## ЛИТЕРАТУРА

1. Judith Hurwitz, Alan Nugent, Fern Halper, Marcia Kaufman. Big Data and Polyglot Persistence. [Электронный ресурс] Режим доступа: https://www.dummies.com/programming/big-data/engineering/big-data-and-polyglot-persistence
2. Polyglot persistence. Материал из Википедии − свободной энциклопедии. [Электронный ресурс] Режим доступа: https://en.wikipedia.org/wiki/Polyglot_persistence
3. Pramod J. Sadalage and Martin Fowler. Introduction to Polyglot Persistence: Using Different Data Storage Technologies for Varying Data Storage Needs. [Электронный ресурс] Режим доступа: https://www.informit.com/articles/article.aspx?p=1930511
4. Мультимодельные СУБД − основа современных информационных систем. [Электронный ресурс] Режим доступа: https://habr.com/ru/post/462493/
5. Multimodel Database. [Электронный ресурс] Режим доступа: https://www.oracle.com/a/tech/docs/multimodel19c-wp.pdf
6. DB-Engines Ranking provided by solid IT, September 2019. [Електроний ресурс] Режим доступу: https://db-engines.com/en/ranking
7. SQL Server technical documentation. [Электронный ресурс] Режим доступа https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15
8. Create a graph database and run some pattern matching queries using T-SQL. [Электронный ресурс] Режим доступа https://docs.microsoft.com/en-us/sql/relational-databases/graphs/sql-graph-sample?view=sql-server-ver15

**ШОФУЛ Кирилл Андреевич** – студент факультета компьютерных наук Харьковского национального университета имени В. Н. Каразина.

Научные интересы:

*– разработка программного обеспечения.*

**ЛАЗУРИК Валентина Михайловна** – старший преподаватель кафедры искусственного интеллекта и программного обеспечения факультета компьютерных наук Харьковского национального университета имени В. Н. Каразина.

Научные интересы:

*– разработка компьютерных систем для моделирования процессов в радиационных технологиях; организация баз данных.*

**V.N. KARAZIN KHARKIV NATIONAL UNIVERSITY**
**NATIONAL SCIENCE CENTER KHARKIV INSTITUTE OF PHYSICS AND TECHNOLOGY**
**MAX PLANCK INSTITUTE OF MICROSTRUCTURE PHYSICS**
**TARAS SHEVCHENKO NATIONAL UNIVERSITY OF KYIV**
**INSTITUTE OF NUCLEAR CHEMISTRY AND TECHNOLOGY (Warsaw, Poland)**
**RIVNE STATE HUMANITARIAN UNIVERSITY**
**NATIONAL AEROSPACE UNIVERSITY NAMED AFTER N.E. ZHUKOVSKY (Kharkiv)**
**CJSC "INSTITUTE OF INFORMATION TECHNOLOGY" (Kharkiv)**
**KHERSON NATIONAL TECHNICAL UNIVERSITY**
**LLC "BUREAU IRIS" (Kyiv)**
**TEAM INTERNATIONAL SERVICES, INC. (Lake Mary, USA)**

**PROGRAM COMMITTEE:**

Azarenkov M.O., Acad. Of the NAS of Ukraine, Prof., Dr. Sc, Kharkiv, Chairman
Bardachov Yu. M., prof., Prof., Dr. Sc, Kherson
BombaA.Ya., Prof., Dr. Sc, Rivne
Buy D. B., Prof., Dr. Sc, Kyiv
VaninV.A., Prof., Dr. Sc, Kharkiv
Gorbenko I.D., Prof., Dr. Sc, Kharkiv
Dolya G.M., Prof., Dr. Sc, Kharkiv
Zholtkevich G.M., Prof., Dr. Sc., Kharkiv
Kuklin V.M., Prof., Dr. Sc, Kharkiv
Lazurik V.T., Prof., Dr. Sc, Kharkiv
Rossomakhin S.G., Prof., Dr. Sc, Kharkiv
Savula Y. H., Prof., Dr. Sc, Lviv
Sporov O. E.,Assoc., Ph.D., Kharkiv
Styervoyedov A. Dr., Halle, Germany
Styervoedov M.G., Assoc., Ph.D., Kharkiv
Tolstoluzhska O.G., Prof., Dr. Sc, Kharkiv
Tkachuk M.V., Prof., Dr. Sc, Kharkiv
Kharchenko V.S., Prof., Dr. Sc, Kharkiv
Khomchenko A.N. Prof., Dr. Sc, Mykolaiv
Shmatkov S.I., Assoc., Dr. Sc, Kharkiv
Shulga M.F., Acad. Of the NAS of Ukraine, Prof., Dr. Sc, Kharkiv
Zimek Z., Ph.D., Warsaw, Poland
Yanovsky V.V., Prof., Dr. Sc, Kharkiv

**ORGANIZING COMMITTEE:**

*Lazurik V.T.*, Dr. Sc, Prof., Dean of the CSD of V.N. Karazin Kharkiv National University, Chairman,
*Sporov O. E.*, Assoc., Ph.D., V.N. Karazin Kharkiv National University, Deputy Chairman,
*Tolstoluzhska O.G.*, Dr. Sc, Prof. of V.N. Karazin Kharkiv National University, Deputy Chairman,
*Tkachuk M.V.*, Dr. Sc, Prof., Head of MST Dept of V.N. Karazin Kharkiv National University,
*Kuklin V.M.*, Dr. Sc, Prof. Head of the AIS Dept of V.N. KarazinKharkiv National University,
*Dyuldya S.V.*, Ph.D., Kharkiv Institute Of Physics And Technology,
*Esin V.I.*, Dr. Sc, Prof. of V.N. Karazin Kharkiv National University,
*Artyukh A.A.*, Head of Laboratory of V.N. Karazin Kharkiv National University,
*Shevtsov S. O.*, CEO LLC "Bureau Iris" (Kyiv)
*Zholtkevich G.M.*,Dr. Sc, Prof. Dean of the MCS of V.N. Karazin Kharkiv National University,
*Vanin V.A.*, Dr. Sc, Prof. NTU “KhPI”,
*Zinoviev D.V.*, Senior Lecturer of V.N. Karazin Kharkiv National University,
*Rassomakhin S.G.*, Dr. Sc, Prof., Head of ISTS Dept of V.N. Karazin Kharkiv National University,
*Styervoyedov A. Dr.*,Max Planck Institute of Microstructure Physics (Germany),
*Petersen S.*, CEO TEAM International (Kharkiv),
*Styervoedov M.G.*, Ph.D., Head of ECS Dept of V.N. Karazin Kharkiv National University,
*Shmatkov S.I.*, Dr. Sc, Prof. of V.N. KarazinKharkiv National University,
*Kruhol M.M.*, assistent of Power Plant Department, NTU KhPI.

**http:// www.univer.kharkov.ua**
**http:// www-csd.univer.kharkov.ua**