CONFERENCE PROCEEDINGS

МАТЕРІАЛИ КОНФЕРЕНЦІЇ

XIII МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ
З ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

InfoCom Advanced Solutions 2025

13th INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE
ON INFORMATION SYSTEMS AND TECHNOLOGIES

21-22 травня 2025 року
Україна, Київ

May 21-22, 2025
Ukraine, Kyiv

Міністерство освіти і науки України
Факультет інформатики та обчислювальної техніки
КПІ ім. Ігоря Сікорського

INFOCOM ADVANCED SOLUTIONS 2025

МАТЕРІАЛИ

ТРИНАДЦЯТОЇ МІЖНАРОДНОЇ
НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ
З ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ
21–22 травня 2025 р., Київ

PROCEEDINGS

OF THE THIRTEENTH INTERNATIONAL
SCIENTIFIC AND PRACTICAL CONFERENCE
ON INFORMATION SYSTEMS AND TECHNOLOGIES
May 21–22, 2025, Kyiv

УДК 004

Наказ ректора КПІ ім. Ігоря Сікорського № НОД/330/25 від 15 квітня 2025 р.

ПРОГРАМА КОНФЕРЕНЦІЇ

CONFERENCE PROGRAM

21 травня / May	***Інформаційні системи та технології / Information systems and technologies***	
	Онищенко В. Власюк Є.	Процес трансформації інформаційної системи обробки та аналізу даних від монолітної до децентралізованої та розподіленої
	Alpert M. Onyshchenko V.	Combined centralized-cooperative mathematical game model of UAV and UGV control
	Rolik O. Smolii N.	Prospects for the Use of Morphological Filters in Rapid Image Processing Tasks
	Ролік О. Волков В.	Метод горизонтального автомасштабування подів у кубернетес для запобігання перегулювання
	Нагайко Д. Ролік О.	Розподіл задач в ієрархічних IoT-системах на основі осмотичних обчислень
	Тимошин Ю. Бачкала Б.	Проблеми забезпечення відмовостійкості хмарних систем на основі оцінок метрик надійності
	Корнага Я. Олексій А.	Вплив інструментів з підтримкою штучного інтелекту на архітектуру та тестування високонавантажених інформаційних систем
	Корнага Я. Лемешко В.	RMRT-EDF: реактивний мультиресурсний токен-планувальник для гарантованого запуску задач у реальному часі
	Корнага Я. Губарєв О.	Метод оптимізації архітектурної структури програмного забезпечення на основі графового аналізу залежностей при переході на мікросервісну архітектуру
	Smovzhenko O. Pysarenko A.	Evaluating Vision-Based Drone Swarms Performance under Visual Occlusions
	Koval V. Zhdanova O. Popenko V.	Multi-UAV mission planning models
	Жданова О. Букасов М. Цимбал С. Чимшир В. Омельченко Р.	Модель і метод формування пакетів сервісів для провайдерів інфокомунікацій
	Вітковський Д. Теленик С.	Технологія взаємодії компонентів інформаційних систем на основі бази даних реактивних сигналів

21 травня / May	***Інформаційні системи та технології / Information systems and technologies***	
	Король С. Олійник В.	Узагальнена модель процесу відстеження погляду користувача вебзастосунків
	Шерстюк І. Голубєв Л.	Удосконалення кореляції Пірсона для подолання проблеми розрідженності матриці в рекомендаційних системах
	Чикрій А. Галушко Д.	Метод ідентифікації транспортних засобів на базі аналізу комбінованих відеопотоків
	М'яч Д. Ромашкевич Я. Солдатова М.	Сучасний стан автоматизованого аналізу публікаційної активності науковців на основі даних цифрового ідентифікатора
	Терентьєв Г. Гавриленко О.	Використання адаптивного підходу для методів формування фінансових портфелів
	Штучний інтелект / Artificial intelligence	
	Новиков Д. Онищенко В.	Підхід на основі комп'ютерного зору для визначення зони посадки БПЛА
	Смовж С. Галушко Д.	Метод виявлення та розпізнавання хвороб пшениці на основі аналізу зображень

22 травня / May	***Інформаційні системи та технології / Information systems and technologies***	
	Галушко Д. Знова К.	Метод обчислення інтегрального показника ризику змін до програмного продукту
	Муліш В. Крилов Є.	Пришвидшення процесу узгодження даних у високонавантажених розподілених інформаційних системах шляхом впровадження розподіленого транзакційного годинника
	Білий М. Крилов Є.	Модифікація методу Reciprocal Rank Fusion для поліпшення результатів гібридного пошуку у інформаційних системах з векторними базами даних
	Мягкий М. Гавриленко О.	Математична модель оцінки експертних якостей

22 травня / May	Гонтар М. Гавриленко О.	Методи аналізу та формування замовлень при роботі з постачальниками продукції
	Фещенко К. Гавриленко О.	Багатофакторна модель виявлення пропаганди в умовах інформаційної війни
	Rekechynska L. Zhurakovska O.	System for Short-Term Job Search
	Lytovchenko A. Zhurakovska O.	Information system for developing listening comprehension skills in foreign language learning
	Kosobutsky A. Zhurakovska O.	Information and Analytical System for UAV Routing
	Kamchatnyi V. Zhurakovska O.	Task Scheduling System
	Vorobiov O. Zhurakovska O.	Multicriteria Choice of Pre-established Businesses on the Marketplace Using the TOPSIS Method
	Makovii V. Zhurakovska O.	Information System to Support the Activities of Fast-food Establishments
	Куцарський М. Хмелюк М.	Гібридний алгоритм підбору елементів з урахуванням обмежуючих параметрів та його застосування в підборі туристичного спорядження
	Нєлєпін Д. Амонс О.	Підхід до підвищення продуктивності веб-застосунків за рахунок перетворення JavaScript у WebAssembly
	Коваленко В.	Підходи до реалізації технології ефективного управління ресурсами у хмарі
	Шутенко В. Жураковський Б.	Телеметрично-покращений адаптивний Bloom-фільтр для розподілених систем з динамічною зміною кількості хеш-функцій
	Штучний інтелект / Artificial intelligence	
	Pysarenko A. Drahan M.	Efficiency of Reinforcement Learning Algorithms in the Mountain Car Task
	Іванов А. Онищенко В.	Автоматичне виявлення цінників на зображеннях товарів
	Технології програмування / Programming Technologies	
	Жиритовський О. Зубко Р.	Формування графічних візуалізацій за допомогою засобів мови програмування з прямим доступом до відеопам’яті

МАТЕРІАЛИ КОНФЕРЕНЦІЇ

CONFERENCE PROCEEDINGS

ЗМІСТ / CONTENTS

Інформаційні системи та технології / Information systems and technologies

Штучний інтелект / Artificial intelligence

Технології програмування / Programming Technologies

ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

INFORMATION SYSTEMS AND TECHNOLOGIES

The Process of Transformation Monolith Data Platforms to Decentralized Data Mesh

Viktoriia Onyshchenko, Yevhenii Vlasiuk
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

Abstract. **This research is dedicated to a transformation process of monolithic centralized platform to distributed Data Mesh. It covers rationale for organizations when and why they should think about modernizing their data platform to Data Mesh, proposes how standard components and layers of traditional centralized data platform should be adopted to Data Mesh architecture to drive efficient level of data quality, data ownership and governance.**

***Keywords*: data platform, Data Mesh, data governance.**

Процес трансформації інформаційної системи обробки та аналізу даних від монолітної до децентралізованої та розподіленої

Онищенко Вікторія, Власюк Євгеній
КПІ імені Ігоря Сікорського
Київ, Україна

Анотація. **Робота присвячена вивченню процесу модернізації централізованих інформаційних систем обробки даних до розподілених систем. В рамках роботи розглядаються причини переходу від централізованих до децентралізованих систем обробки даних, запропоновані компонентні діаграми обох типів систем, а також описаний процес трансформації програмної архітектури, запропоновані програмні модулі як частини програмної архітектури розподіленої системи для вирішення питань власності, доступності і якості даних.**

Ключові слова: інформаційна система обробки даних, аналіз даних, децентралізована система.

Вступ

Однією з фундаментальних змін минулих десятиліть в галузі цифровізації та інформатизації є здешевлення зберігання даних. Це запустило ланцюгову реакцію змін як компанії та організації сприймають свої дані на всіх етапах життєвого циклу, починаючи від збору даних, закінчуючи їх обробкою, аналізом і використанням для прийняття рішень. Це дозволило компаніям з індустрій, які історично не пріоритезували інформатизацію за основу свого бізнесу, як от легка промисловість, машинобудування та інші, почати будувати свої IT відділи, розбудовувати свою IT інфраструктуру, будувати свої сховища даних і системи обробки і аналізу даних [1]. Дані стали критичним фактором в прийнятті рішення щодо оптимізації роботи підприємств, збільшення доходу, залучення нових користувачів, тощо.

Інформаційні системи обробки даних пройшли довгий шлях еволюції від мейнфреймів до хмарних систем, від систем основаних на реляційних базах даних до гібридних систем з використанням розподілених файлових систем, поєднання реляційних, нереляційних баз даних, платформ індексації даних, повнотекстового пошуку по даних, тощо.

Проте в останні роки, нові виклики змушують адаптувати системи обробки і аналізу даних до поточних реалій. Стрімке збільшення даних різних за своєю природою, формою, а головне, суттю в рамках однієї організації, змушує замислитись, чи доцільно мати одну систему обробки всіх даних чи варто ще більше децентралізувати цей процес [2]. На користь монолітної централізованої системи обробки і аналізу даних говорить можливість об'єднувати дані з різних бізнес доменів в цілісні аналітичні продукти. Мати розуміння як рекламна кампанії маркетингового відділу вплинула на продажі і збільшення виручки при цьому спрогнозувати об'єм виробництва в наступному кварталі, позитивно впливатиме на роботу компанії. В той же час підхід до об'єднання обробки всіх даних в одну платформу призводить до значного розростання останньої, змішування зон

відповідальності і власності даних серед різних команд і відділів. Спеціалісти з фінансового відділу не зможуть достовірно розуміти і трактувати маркетингові дані і навпаки.

Саме тому організації все щастіше рухаються від монолітних систем обробки даних до децентралізованих і орієнтованих під конекретний домен бізнесу, до якого будуть відноситись дані. Процес такої трансформації вимагає суттєвих змін в програмній архітектурі системи, модифікації існуючих програмних компонентів, а також створення нових програмних модулів та підсистем. Саме тому варто розглянути цей процес детальніше.

Рівні монолітної інформаційної системи обробки та аналізу даних

На рис. 1 представлена компонентна діаграма рівнів монолітної інформаційної системи обробки та аналізу даних. Дані з першоджерел завантажуються в продуктовий рівень системи. Модулі цього рівня відповідають за очищення, доповнення, фільтрацію, агрерування даних і в решті-решт за створення програмних продуктів на основі даних – звітів, візуалізацій [3]. Операційний рівень системи містить в собі модулі контролю якості даних, моніторингу і планування виконання процесів обробки даних, а також керування інтеграцією основних даних компанії задля уникнення їх дублювання. Фінансові, маркетингові та інші набори даних обробляються однією платформою, контролюються і моніторяться одними модулями, що дозволяє перевикористовувати дані для різних цілей, виділяти набори даних, що використовуються в різних сферах, в якості спільних і надавати доступ до них різним користувачам, залишаючи контроль над якістю і наповненістю цих наборів даних за централізованим операційним компонентом. Цей аспект є критично важливим для забезпечення унікальності та консистентності даних – одних з найважливіших аспектів якості даних.

Рис. 1. Компонента діаграма рівнів централізованої інформаційної системи обробки та аналізу даних

Якщо розглянути, наприклад, набір даних про компанії, що оперують на ринку, то є критично важливим аби цей набір даних був унікальний в рамках системи і використовувався всіма процесами і продуктами. Зворотнє може призвести до ситуації коли інформація про одну й та ж компанія буде присутня в двох наборах даних і певні атрибути можуть відрізнятись, наприклад, інформація про розмір чи адресу компанії.

Основні складнощі і недоліки монолітних систем обробки і аналізу даних виникають в аспектах керування якістю даних, а також володінням і доступом до даних. Для коректного використання даних критично важливо правильно розуміти і тлумачити кожен атрибут набору даних. Набір даних, який представляє фінансові показники і результати продажів за минулий квартал може містити десятки фінансових показників і представлень доходу, отриманого від продажів. Лише експерт відповідної галузі зможе коректно розтлумачити таку інформацію і прийняти рішення щодо її коректного використання для аналізу впливу нещодавно реалізованої маркетингової кампанії. Саме тому чітке розмежування володінням і відповідальністю за дані є важливим аспектом коректного використання даних. Не менш важливим цей аспект є і для коректної перевірки і забезпечення необхідної якості даних. Тільки експерт і власник набору даних може прийняти рішення з приводу коректності, повноти і консистентності. Відсутність значення в певному атрибуті певного запису, дробове значення чи запис з нестандартним кодуванням пересічному користувачу можуть здаватись як аномалії даних проте можуть бути цілком очікуваними відповідно до поточних бізнес процесів в організації. Тільки експерт і власник даних може прийняти остаточне рішення, щодо достовірності даних, можливості чи неможливості використання набору даних для прийняття рішення. Проте маючи всі дані в рамках однієї системи складно розмежувати кордони володіння даними і визначити відповідальність за дані. Це призводить до некоректного використання і трактування даних, узагальнення правил контролю за якістю даних і врешті-решт прийняттю помилкових рішень на основі даних.

Децентралізовані розподілені інформаційні системи обробки та аналізу даних

Одним із варіантів вирішення цих питань є перехід від монолітної до розподіленої інформаційної системи обробки та аналізу даних (Data Mesh) [4]. Компонента діаграма такої системи наведена на рис. 2. Така система має ряд додаткових рівнів і компонентів аби забезпечити розділення даних, процесів їх обробки та аналізу, а також володіння, доступу і відповідальності за дані між різними бізнес доменами, командами і відділами. В запропонованій інформаційній системі спеціалізовані дані кожного бізнес домену обробляються спеціалізованим і незалежним рівнем застосунків обробки і аналізу даних. Операційний рівень зазнає найбільших змін, оскільки має забезпечувати федеративне керування ресурсами системи. В децентралізованій системі пропонується розділити операційний рівень на два: доменний і корпоративний. Доменний рівень буде відповідати за керування процесами обробки і аналізу даних, які специфічні до природи і походження даних, і будуть залишатись в зоні відповідальності експертів і власників даних.

Це зокрема контроль якості даних, контроль доступу до даних, моніторинг та планування оскільки вони залежать від частоти оновлення даних, правил перетворення і агрегування. Важливою зміною є те, що тепер дані розподілені між доменами, мають різні рівні доступу і тому немає одного централізованого місця зберігання всіх даних. Як наслідок, постає питання побудови аналітичних продуктів, які поєднують в собі дані різних доменів, як наприклад, вищезгаданий звіт про вплив рекламної кампанії на результати продажів і прогнозування виробництва.

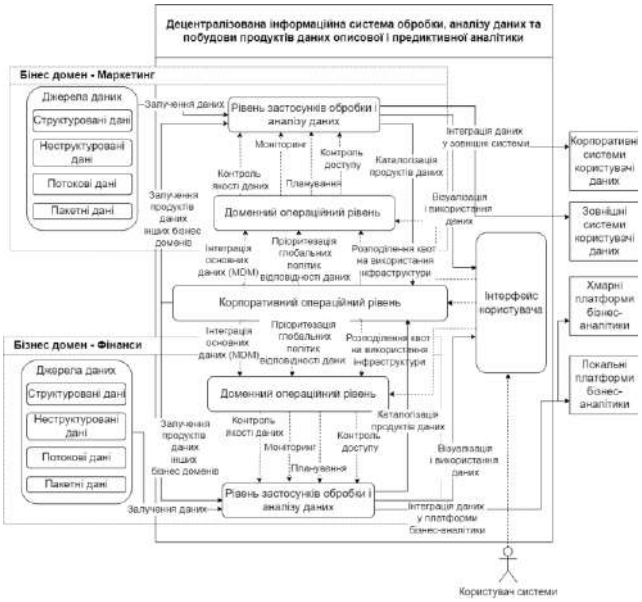

Рис. 2. Компонента діаграма рівнів децентралізованої розподіленої інформаційної системи обробки та аналізу даних

Важливим кроком під час трансформації монолітної інформаційної системи обробки і аналізу даних в децентралізовану є створення корпоративного операційного рівня. Він містить в собі програмні модулі, які забезпечуватимуть контрольоване використання продуктів даних між бізнес доменами, пріоритезуватиме глобальні правила перевірки і валідації даних, а також контролюватиме і розподілятиме квоти на використання обчислювальних ресурсів в рамках інфраструктури системи.

Корпоративний операційний рівень є централізованим компонентом в децентралізованій системі. Тому при виділенні корпоративного операційного рівня важливо зберегти необхідний рівень гнучкості і контролю для доменних підсистем, аби не звести загальну архітектуру системи до монолітної. В той же час важливо встановити чіткі рамки і правила, виконання яких є обов'язковими для підсистеми кожного домену, оскільки в інакшому випадку це може нести ризики для всієї організації. Прикладами таких правил можуть бути політики збору, зберігання, обробки і розповсюдження персональних даних, робота у відповідності до стандартів і сертифікацій, тощо.

Останнім, але не менш важливим компонентом корпоративного операційного рівня запропонованої розподіленої системи обробки і аналізу даних, є компонент інтеграції основних даних. В розподіленій системі контроль за унікальністю наборів даних стає складнішим оскільки підсистеми самодостатні і ізольовані на етапі виконання процесу обробки і аналізу даних. Компонент інтеграції основних даних мусить залишатись централізованим і оперувати на рівні всієї організації для унеможливлення дублювання наборів даних між різними доменами. При цьому компонент повинен двосторонньо інтегруватись з усіма доменами, отримувати інформації від кожного з них про нові набори даних і аналізувати чи відповідає цей набір даних критеріям основного набору даних. У випадку, якщо набір даних визнано основним, відповідальність за нього переноситься з доменної команди на централізовану команду і відповідно процес його оновлення стає корпоративним процесом.

Висновки

Процес трансформації монолітних інформаційних систем обробки даних в децентралізовані потребує значних архітектурних змін, створення нових програмних компонентів, а також залучення різних команд і експертів. Цей процес також передбачає зміну секторів відповідальності для існуючих команд експертів і створення нових команд, які будуть відповідати за корпоративний операційний рівень і координувати роботу системи загалом, даючи можливості для оперування доменним експертним командам. Подальші дослідження у цій тематиці будуть присвячені методам визначення ознак для ідентифікації наборів даних, як основних, а також методів визначення дублюючих наборів даних в рамках різних доменних підсистем.

Література

1. Anitha P, Malini M. Patil," A Review on Data Analytics for Supply Chain Management: A Case study", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.10, No.5, pp. 30- 39, 2018. DOI: 10.5815/ijieeb.2018.05.05

2. Vlasiuk, Y., Onyshchenko, V. (2023). Data Mesh as Distributed Data Platform for Large Enterprise Companies. In: Hu, Z., Dychka, I., He, M. (eds) Advances in Computer Science for Engineering and Education VI. ICCSEEA 2023. Lecture Notes on Data Engineering and Communications Technologies, vol 181. Springer, Cham. https://doi.org/10.1007/978-3-031-36118-0_17

3. Anosh Fatima, Nosheen Nazir, Muhammad Gufran Khan,"Data Cleaning In Data Warehouse: A Survey of Data Pre-processing Techniques and Tools", International Journal of Information Technology and Computer Science (IJITCS), Vol.9, No.3, pp.50-61, 2017. DOI: 10.5815/ijitcs.2017.03.06

4. Dehghani Z. How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh. 2019. / https://martinfowler.com/articles/data-monolith-to-mesh.html (дата звернення 1.05.2025)

Combined centralized-cooperative mathematical game model of UAV and UGV control

Alpert Maksym, Onyshchenko Viktoriia
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
max292009@gmail.com, v.onyshchenko@kpi.ua

Abstract. **This paper discusses the application of cooperative game-theoretic methods combined with centralized control strategies for optimizing unmanned vehicles operations. It addresses mission planning, resource allocation, and route optimization for unmanned vehicles coordination.**

Keywords: unmanned vehicles, UAV, UGV, cooperative game theory, optimization, route planning, centralized control.

INTRODUCTION

The exponential growth and complexity of UV (Unmanned Vehicles) operations, especially in tasks involving coordination between multiple units, have intensified the need for advanced optimization methods. Traditional methods of autonomous or semi-autonomous control become increasingly inadequate as mission complexity and uncertainty increase. Among various modern methodologies, game-theoretic methods have emerged as robust solutions for addressing these challenges, particularly the cooperative game theory combined with centralized control, ensuring optimal mission outcomes [1]. This paper details the development, analysis, and practical implications of integrating these methods into UV coordination tasks, specifically emphasizing search-and-rescue missions.

The UV systems have been extensively utilized in diverse operations, including civilian and commercial applications. The capability to perform sophisticated tasks autonomously, like territory monitoring, object tracking, and logistical support, places UV technology at the forefront of current and future technological advancements. However, managing these autonomous systems under constrained resources and uncertain environments requires innovative, flexible, and highly effective control strategies. This research proposes a combined centralized-cooperative mathematical game model to enhance UV management, particularly in critical missions.

METHODOLOGY

The core of the proposed method revolves around cooperative game theory. Cooperative game theory is advantageous due to its ability to consider multiple agents (UVs) collaborating towards a shared objective. This method effectively manages interactions among multiple UVs by addressing several critical operational factors, including:

- Coverage area and its optimization.
- UV movement speed and trajectory efficiency.
- Payload management under severe limitations, including human, financial, and energy resource constraints.
- Optimal route planning among high uncertainty and risks, notably large static obstacle detection using neural network approaches [2].
- Strategic selection and optimization of UV charging stations to minimize downtime and operational disruptions [3].

MATHEMATICAL MODEL DEVELOPMENT

The developed mathematical model integrates cooperative game theory principles with centralized control mechanisms. Under this model, all UV operations are orchestrated through a singular control entity, streamlining command structure and resource management. The general payoff function, denoted F_{total}, encapsulates the comprehensive mission objective, defined by maximizing operational coverage while simultaneously minimizing resource utilization:

$$F_{total} = \sum_{j=1}^{3} a_j F_j - \sum_{j=4}^{5} a_j F_j \rightarrow max \quad (1)$$

In this formula, F_j individually represents coverage, speed, payload, energy, and financial expenditure. Each parameter has an associated priority a_j, ensuring total prioritization unity:

$$\sum_{j=1}^{5} a_j = 1; 0 \leq a_j \leq 1; j = \overline{1,5} \quad (2)$$

It is assumed that all functions f_i are normalized. $x_i(t)$ is the state of the i-th control object at time t. UV prioritization is described as follows (3):

$$\sum_{i=1}^{N} p_i = 1; 0 \leq p_i \leq 1; \; i = \overline{1,N} \quad (3)$$

The specialized payoff functions for each operational component are structured comprehensively to ensure detailed optimization across the mission's various aspects.

Coverage optimization focuses on maximizing the area covered by UVs throughout the mission, taking into account the current state of each UV at any given moment to effectively address real-time demands (4).

$$F_1 = \sum_{i=1}^{N} p_i * f_{i\,coverage}\big(x_i(t)\big) \quad (4)$$

Velocity maximization aims to achieve the highest possible speed collectively for the UVs, enabling quicker response and shorter mission duration (5).

$$F_2 = \sum_{i=1}^{N} p_i * f_{i\,velocity}\left(x_i(t)\right) \quad (5)$$

Payload maximization addresses the efficient management and optimization of carried payload, crucial for accomplishing mission objectives, especially in scenarios like search-and-rescue operations where the timely and secure delivery of essential items can be mission-critical (6).

$$F_3 = \sum_{i=1}^{N} p_i * f_{i\,payload}\left(x_i(t)\right) \quad (6)$$

Moreover, the model includes energy consumption minimization, specifically formulated through a quadratic relationship with the control inputs of the UVs, where the energy function is represented as a square of the control input for each UV. This formulation effectively captures the nonlinear increase in energy consumption with aggressive maneuvering or rapid movements, thereby encouraging efficient operational patterns (7-8). $u_i(t)$ is the control function for the *i*-th control object (UV) at time *t*.

$$F_4 = \sum_{i=1}^{N} p_i * f_{i\,energy}\left(u_i(t)\right) \quad (7)$$

$$f_{i\,energy}(u_i(t)) = (\|u_i(t)\|)^2 \quad (8)$$

Similarly, financial cost minimization is modeled through an analogous quadratic relationship, reflecting the escalating financial expenses associated with heightened operational intensity or frequent adjustments during UV missions (9-10).

$$F_5 = \sum_{i=1}^{N} p_i * f_{i\,finance}\left(u_i(t)\right) \quad (9)$$

$$f_{i\,finance}(u_i(t)) = \left(u_i(t)\right)^2 \quad (10)$$

Additionally, an ideal payoff function, denoted as F_{ideal} is introduced to benchmark the efficiency of actual mission outcomes against an optimal theoretical scenario characterized by zero energy and financial expenditures (11). This ideal scenario provides a clear performance target, enabling quantitative assessment and facilitating continuous improvements in operational strategies by evaluating the achieved efficiency relative to the maximum attainable effectiveness under ideal conditions (12).

$$F_{ideal} = \sum_{j=1}^{3} a_j F_j \quad (11)$$

$$Effectiveness = \frac{F_{total}}{F_{ideal}} * 100\% \quad (12)$$

RESULTS

Extensive simulation trials were conducted to validate the proposed model's effectiveness in realistic scenarios, such as coordinated search-and-rescue operations involving multiple UV types (UAV and UGV). The simulation results yielded compelling evidence supporting centralized-cooperative methodologies (fig.1):

- Efficiency of the Centralized-Cooperative model achieved 71.12%.
- Decentralized-Cooperative models with two and five control centers attained efficiencies of 65.06% and 22.66%, respectively.

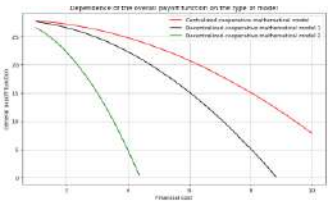

Fig. 1 Dependence of the payoff function on the type of model

PRACTICAL IMPLEMENTATION

Implementing the centralized-cooperative game-theoretic model involves strategic adjustments to UV control systems, emphasizing the following components:

- Efficiency of the Centralized-Cooperative model achieved 71.12%.
- Operator-centric control ensuring cohesive management of all UV operations through a single interface.
- Real-time monitoring and adaptive mission adjustments via advanced neural-network-driven obstacle detection.
- Optimized resource allocation across all UV units to ensure minimal downtime and maximum mission efficacy.

Advantages observed from practical implementation include reduced operational overhead, enhanced response times in critical mission scenarios, and optimized allocation of human, energy, and financial resources.

CONCLUSION

The research demonstrates that integrating cooperative game theory and centralized control significantly advances UAV operational management. By streamlining command structures, enhancing coordination efficiency, and optimizing resource usage, the proposed methodology provides tangible operational and economic benefits. The findings underscore the viability and strategic importance of this integrated approach in managing complex UV missions effectively, paving the way for broader adoption in both civilian and defense sectors.

REFERENCES

[1] M. L. Cummings, "Operator Interaction with Centralized Versus Decentralized UAV Architectures," in Handbook of Unmanned Aerial Vehicles, K. P. Valavanis and G. J. Vachtsevanos, Eds. Dordrecht: Springer, 2015, pp. 977–991. https://doi.org/10.1007/978-90-481-9707-1_117.

[2] M. Alpert and V. Onyshchenko, "Recognition of Potholes with Neural Network Using Unmanned Ground Vehicles," in Advances in Computer Science for Engineering and Education, vol. 134, pp. 209–220, 2022. https://doi.org/10.1007/978-3-031-04812-8_18.

[3] M. Alpert, "Using Game Theory to Improve Drone Operations," in Control Systems and Computers, vol. 1, pp. 57–61, 2024. https://doi.org/10.15407/csc.2024.01.05

Prospects for the Use of Morphological Filters in Rapid Image Processing Tasks

Oleksandr Rolik, Natan Smolii
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
smolii.natan@lll.kpi.ua

Abstract. **Morphological operation is a technique used in image processing to modify and enhance digital images. It involves the use of a structuring element (analogous to the kernel in spatial filtering) that is applied to each pixel in an image to alter its value based on the surrounding pixel values.**

Keywords: morphological operations, image processing, feature detection.

INTRODUCTION

Morphological operations are commonly used in image processing on stages of noise reduction or photo editing for such tasks as blurring or sharpening of images. However, primary data procession is not everything what they can be used for. Recent AI development proved that without such operations it is impossible to handle data properly. Mainly, these operations are used in image processing models, where models are used to fit values of filter kernels to training data and so, learn how to detect bigger features on the image. However, these filters are also used by humans for multi-layer image processing and since they were used there even before AI period began there existed a task of assembling a chain of filters in order to extract valuable numerical data from image.

LITERATURE REVIEW

In the article [1], the principles of convolutional neural networks (CNNs) are described in detail, along with an illustration of how morphological filters are integrated into their architecture. The study provides insights into the synergy between morphological operations and deep learning techniques for enhancing image analysis performance.

A reference book [2] demonstrates the usage of Sobel filters for gradient extraction in images through the OpenCV library. Additionally, it offers explanations on the application of custom morphological filters with user-defined convolution kernels in various image processing tasks, such as edge enhancement and object detection.

The source [3] presents the foundational principles and algorithms for constructing custom morphological filters. These filters are shown to be effective in tasks such as noise reduction, feature detection, and general enhancement of image quality, using both classical and adaptive approaches.

The study [4] provides a practical example of using morphological filters for the detection and classification of specific regions in images. It also offers guidance on employing dilation and erosion operations as efficient techniques for reducing noise in edge-dense areas, thus improving the clarity of object boundaries.

Article [5] highlights the prospects of using drones for water sampling and presents a mathematical calculations and formulas for drone positioning in the global coordinate system. It provides fundamental recommendations for controlling a drone equipped with additional equipment, taking into account dynamic loads and shifts in the center of mass.The article also contains data on disturbing influences on the system that can disable the aerial vehicle and require calculated and calibrated compensation, indicating the need for computational resources to process the mathematical model of a drone with a dynamic center of mass that is subject to significant environmental influences.

PROCESSES DESCRIPTION

Convolution

Numerous image processing techniques have been developed for object detection tasks within visual data. Among classical approaches, the Hough Transform remains a well-established method, relying on the mathematical parametrization of desired geometric shapes to detect them within an image. Despite its utility in controlled environments, the Hough Transform presents notable limitations when applied to scenes containing a large number of objects of unknown or irregular forms. Furthermore, the method often requires solving high-dimensional optimization problems, which hinders its applicability in real-time or computationally constrained scenarios. Additionally, this technique is inherently unsuitable for tasks requiring semantic or instance-level image segmentation, as it lacks the capacity to generalize beyond predefined geometric primitives.

In contrast, modern segmentation methods based on convolutional neural networks (CNNs) have demonstrated significant capabilities in pixel-level classification. These networks are capable of capturing both local contextual information and high-level structural features such as surface textures and visual similarities to known object classes within training datasets. However, segmentation architectures such as U-Net, DeepLab, or Mask R-CNN are computationally intensive

and require large volumes of annotated data for supervised training. Moreover, their resource demands far exceed those of traditional object detectors, which generally operate on region-based localization strategies and serve as precursors to full segmentation pipelines.

When considering real-time video stream processing—one of the most critical tasks in modern computer vision—a number of practical challenges emerge. Among the most prominent are the trade-offs between detection speed, accuracy, and computational efficiency. Current state-of-the-art object detection pipelines often rely on neural network architectures from the YOLO (You Only Look Once) family, which are specifically optimized for high-speed inference. These models demonstrate favorable frame-per-second (FPS) performance, making them well-suited for real-time applications; however, they tend to offer lower detection accuracy and robustness compared to other, more complex detection or segmentation-oriented neural networks, such as RetinaNet, Faster R-CNN, or semantic segmentation models.

In the context of object tracking, a different class of algorithms is typically employed. Methods such as CSRT (Channel and Spatial Reliability Tracking), KCF (Kernelized Correlation Filters), and MOSSE (Minimum Output Sum of Squared Error) offer effective tracking capabilities while maintaining relatively low computational overhead. Their lightweight nature allows them to be deployed on resource-constrained platforms, such as embedded devices or edge processors, without significant degradation in performance. This makes them a practical choice for scenarios where neural tracking approaches (e.g., deep SORT or transformer-based trackers) may be infeasible due to latency or power limitations.

Morphological operations constitute a class of image processing techniques based on the application of a convolution-like operation using a predefined structuring element. These operations are designed to modify the spatial structure or "morphology" (in analogy to its use in linguistics as the study of form) of an image. Morphological transformations are typically employed to enhance specific object shapes, extract structural features, or suppress unwanted patterns such as noise or irrelevant textures.

Depending on the selected operation—such as dilation, erosion, opening, or closing—morphological processing can emphasize or remove particular spatial configurations within the image. For example, dilation tends to expand bright regions, whereas erosion contracts them. These operations are particularly effective in binary or thresholded grayscale images where object boundaries are clearly defined.

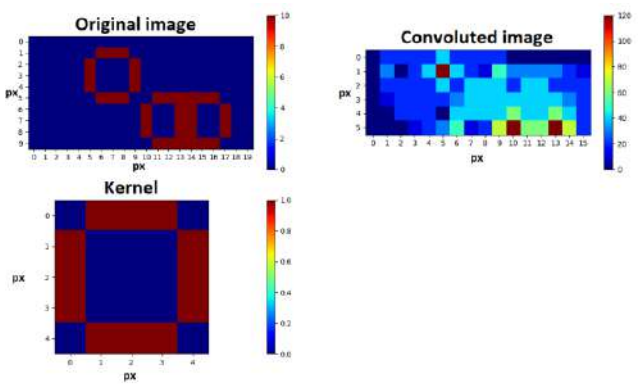

Fig. 1. Exampe of convolution operation

A practical illustration of morphological processing used to detect circular objects is shown in Fig. 1. In this example, morphological filtering with a circular structuring element facilitates the enhancement and localization of circular shapes within the input frame, demonstrating the utility of this technique in geometric pattern recognition tasks. The original input image is shown in the top-left panel. The structuring element used for morphological convolution is depicted in the bottom-left corner. The resulting convolved image is presented in the top-right panel. Color maps correspond to value intensities, and axes indicate image dimensions in pixels.

This work proposes the use of morphological transformations as a foundation for high-level image analysis, specifically for the detection of generalized, higher-order objects rather than traditional low-level features such as gradients or edges. By leveraging the structural specificity offered by morphological operators, it becomes possible to isolate abstract visual patterns relevant to task-oriented perception.

To validate the approach, a series of experiments were conducted within the Gazebo robotic simulation environment. The test scenario involved three autonomous aerial vehicles (UAVs) of the same physical model, each assigned a distinct operational role. Two of the drones were tasked with searching for and localizing a moving object within the simulated environment. The third UAV was responsible for maintaining its position directly above the localized object, effectively enabling coordinated tracking behavior.

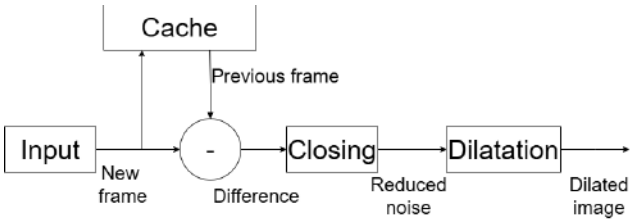

Fig. 2. Structure of used video processing pipeline

Object detection in the proposed system was performed using a morphological transformation pipeline, illustrated in Fig. 2. This pipeline enables the identification of changes between consecutive video frames through elementary frame differencing. By computing the pixel-wise difference between successive frames, the system highlights motion-induced regions which are further refined through morphological operations to suppress noise and enhance coherent structural features.

The overall architecture of the transformation chain is designed to be computationally lightweight while still providing sufficient sensitivity to dynamic changes in the scene, making it suitable for deployment in real-time video processing scenarios involving mobile platforms. Following the computation of inter-frame differences, a thresholding operation is applied to the resulting difference image to convert it into a binary format. This binarization step effectively filters out low-intensity variations caused by minor environmental movements or camera jitter—particularly relevant given the use of UAV-mounted cameras, which are prone to micro-movements during flight.

To further suppress noise and improve the coherence of detected regions, a morphological closing operation with a 5×5 convolution kernel is applied. This step helps eliminate small

holes and bridge narrow gaps in the binary image, resulting in more stable and structured representations of moving objects.

Subsequently, a dilation operation using a larger 11×11 kernel is performed to aggregate fragmented detection points that likely belong to the same object. This enhances the spatial continuity of the detected regions and facilitates the morphological consolidation of object contours, enabling more reliable tracking and classification in later stages of the pipeline.

A comparison between the original image and the results of the morphological processing pipeline is shown in Fig. 3. After the morphological operations have been applied, the resulting set of points is reduced to a single representative point, which characterizes the center of the moving object's cluster. This point is then used to approximate the position of the moving object within the image frame.

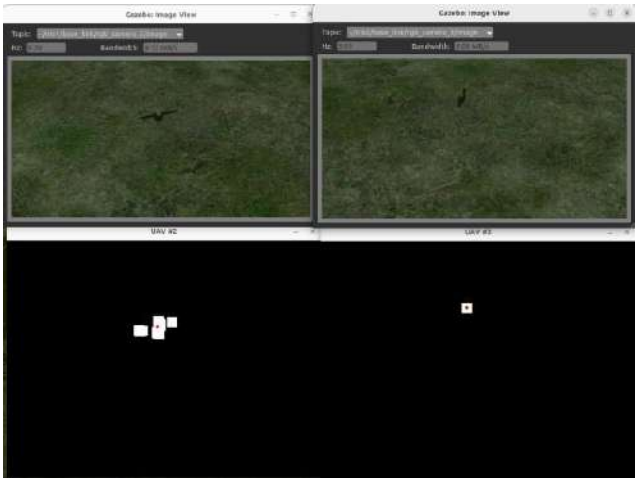

Fig. 3. Unprocessed and processed images.

While this method is not perfect and has inherent limitations—particularly in cases of occlusion or overlapping objects—it serves as a practical and computationally efficient approach for estimating the position of a moving object. Despite its simplicity, the method provides a reasonable estimation of object localization that is sufficiently robust for many real-time tracking applications, especially when computational resources are limited. Transformations for converting iamge points into 3D vectors require steps described below.

The calculation of the optical axis vector OA in the drone's coordinate system (CS) for a static hover, with tilt angles around the OX and OY axes equal to φ and θ, respectively, considering the vector OX [1.0, 0.0, 0.0] as the reference for the drone, is performed as follows:

$$R_z(\varphi) = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \quad (2)$$

$$\overrightarrow{OA} = R_z(\varphi) * R_y(\theta) * \overrightarrow{OX}. \quad (3)$$

The projection of point P on the image to a spatial vector in the camera's coordinate system uses values such as the focal length f in millimeters along the x and y axes, as well as the target image dimensions I_w and I_h, and is performed using the following formula:

$$Cam * \overrightarrow{P} = \begin{bmatrix} f_x & 0 & I_w/2 \\ 0 & f_y & I_h/2 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} P_x \\ P_y \\ 1 \end{bmatrix} = \overrightarrow{PW}. \quad (4)$$

That requires further transformation into the world coordinate system (CS) with following matrix should be used:

$$CamToWorld = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}. \quad (5)$$

For the subsequent transformation into a vector associated with the current orientation of the drone, which is typically represented in flight controllers as a quaternion, it is necessary to form the orientation quaternion for the obtained spatial vector and then rotate it to the drone's orientation quaternion. The following transformations are performed:

$$Q(\overrightarrow{PW}) = \begin{bmatrix} 0.0 \\ PW_i \\ PW_j \\ PW_k \end{bmatrix}. \quad (6)$$

Then, the resulting quaternion PQ is calculated based on the drone's orientation quaternion as follows:

$$PQ = DQ * Q(\overrightarrow{PW}) * DQ^*. \quad (7)$$

The resulting formula for transferring a point on the image to a spatial vector in the world coordinate system:

$$R_z(\varphi) * R_y(\theta) * CamToWorld * Cam * \overrightarrow{P}. \quad (8)$$

Subsequently, after obtaining two direction vectors for the moving objects, it is necessary to solve the problem of finding the two closest points that belong to two lines and determining the intermediate point that nullifies the discrepancies in coordinate estimation. The points that belong to the lines can be found using the formula A=B+K·t, where t can be calculated by performing the operations described in the following formulas:

$$d = O_1 - O_0, \quad (9)$$

$$n = V_0 \times V_1, \quad (10)$$

$$t_i = \frac{(n \times V_{1if\ i==0else\ 0}) \bullet d}{n \bullet n}, \quad (11)$$

Also, for further research, a comparison of image processing time depending on image resolution was carried out. For testing, a randomly generated black-and-white image of the specified size was used, on which the following operations were performed:

- three-stage convolution with filters of random sizes to measure the impact of convolution operations on processing time;
- exponential raising of pixel values and raising the result to the 5th power to determine the time cost of mathematical operations on data arrays;
- loop iterating through values in each row until a certain number is found, as an imitation of user-defined algorithmic image processing.

The results of the study are shown in Fig. 4:

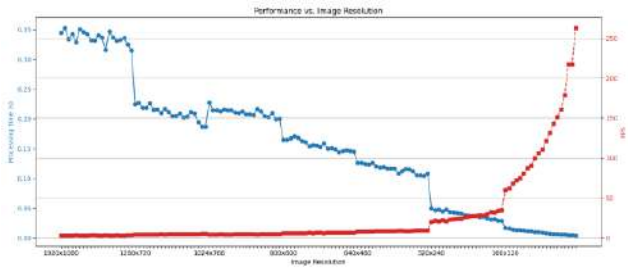

Fig. 4. Measured dependencies between processing time and image resolution.

The illustration presents two color graphs showing the time required to process an image of the specified resolution (blue graph with the scale on the left) and the expected frame rate (red graph with the scale on the right) at the output of the transformation pipeline. The frame rate graph can be used when designing systems with strict frame rate requirements, as this parameter is typically the most critical in system specifications.

It can be observed that for images processable by humans (resolutions from 1920x1080 to 640x480), the image processing time using the proposed test method decreases linearly, while an exponential drop is only observed for very small image sizes (320x240 and below), the processing of which results in significant accuracy loss. The accuracy loss is due to the fact that, with a fixed camera field of view, different image resolutions yield different levels of discretization for the captured image. As a result, when applying the transformation from formula (4), the obtained results vary significantly and affect the precision of the calculated final position using formula (11). Additionally, the graph shows a stable increase in processing time for a resolution of 1024x768, which should be considered during image processing.

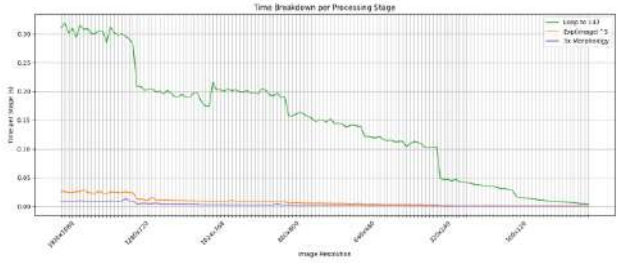

Fig. 5. Processing time splitted between three types of image processing tasks.

In Fig. 5, the time expenditures for each proposed processing stage are shown separately.

The green color indicates the time spent on the custom loop for processing pixel values, orange denotes the time taken for mathematical operations, and blue represents the time required to perform convolution operations; the blue graph shows the time expenses for mathematical operations on the image.

It can be seen that the greatest time costs are associated with the custom pixel iteration loop, indicating that the main method for optimizing image processing time is to eliminate components responsible for direct iteration from the processing program and shift to using specialized libraries for mathematical operations (NumPy, TensorFlow, SciPy, GLM), as they provide up to a tenfold reduction in processing time in some cases, which can be observed at a resolution of 1920x1080. Additionally, the analysis of this graph indicates that the increased processing time at a resolution of 1024x768 is specifically due to the use of the custom loop and cannot be avoided without a deep understanding of the programming language used to write the processing application.

Results

A thorough review of existing real-time video stream processing techniques for the detection and tracking of higher-order morphological structures has been conducted. Based on the insights from the analysis, a prototype image processing module was implemented using fundamental morphological operations. The system demonstrates the capability to extract and interpret structural patterns in video data by leveraging a mathematical framework tailored to morphological transformation outputs.

An analytical performance evaluation revealed the correlation between image resolution and processing latency. The results were presented in the form of resolution-dependent frame rate graphs, offering practical insights into system scalability. These findings led to a set of optimization recommendations.

Acknowledgment

The research was by the National Research Foundation of Ukraine under project No. 2023.04/0077 "Drone for water sampling".

References

[1] Yamashita, R., Nishio, M., Do, R.K.G. *et al.* Convolutional neural networks: an overview and application in radiology. *Insights Imaging* 9, 611–629 (2018). https://doi.org/10.1007/s13244-018-0639-9

[2] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media, 2008. [1, p. 148]

[3] P. Maragos and L. F. C. Pessoa, "Morphological filtering for image enhancement and detection," *in The Image and Video Processing Handbook*, Academic Press, 2000, pp. 1–26.

[4] F. G. B. De Natale and G. Boato, "Detecting Morphological Filtering of Binary Images," in IEEE Transactions on Information Forensics and Security, vol. 12, no. 5, pp. 1207-1217, May 2017, doi: 10.1109/TIFS.2017.2656472.

[5] Polishchuk, M., & Rolik, O. (2024). Improvement of Technological Equipment Drone for Water Sampling: Design and Modeling. FME Transactions, 52(2), 237–245. doi: 10.5937/fme2402237P

Method of Horizontal Pod Autoscaling in Kubernetes to Omit Overregulation

Oleksandr Rolik, Volodymyr Volkov
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
o.rolik@kpi.ua, ask4ua@gmail.com

Abstract. This work describes the method of omitting the over-regulation effect that occurs under certain conditions by horizontal pod autoscaling microservices in the container application orchestration system Kubernetes. The effect was initially observed only for long-term HTTP WebSocket sessions, where it led to excessive use of computing resources, which reduced the efficiency of IT infrastructure management, and caused service failure. It was found that the overregulation effect is reproduced not only for connections with long-term HTTP sessions, such as HTTP WebSocket, but also for shorter-term REST HTTP sessions in case of increased delay in the metric collection cycle used for horizontal pod autoscaling. It is assumed that this effect happens due to the approach of implementing horizontal scaling controllers similar to the principles of proportional regulators in systems with negative feedback from the theory of automation and control. It is proposed to extend one of the methods used for optimizing the proportional controller to the problem consisting of reducing the time delay between scaling metrics collecting and upscale applied by the controller in Kubernetes. The applied method demonstrated its effectiveness, therefore, within the same methodology, an experiment was conducted on using the proportional-integral-differential controller for automatic horizontal scaling of pods. The results obtained showed why the proportional-integral-differential controller is not widespread among the overviewed Kubernetes solutions for horizontal automatic scaling. An assumption was made about the limitations of studying the downscaling process in Kubernetes due to the need to consider the quality of service when stopping pods and the need to collect indicator metrics using quality-of-service object management tools such as ISTIO.

Keywords: Kubernetes, Microservices, Horizontal Pod Autoscaling, Proportional Regulation, Cloud Computing.

Метод горизонтального автомасштабування подів у кубернетес для запобігання перегулювання

Олександр Ролік, Володимир Волков
КПІ імені Ігоря Сікорського
м.Київ, Україна
o.rolik@kpi.ua, ask4ua@gmail.com

Анотація. **В роботі представлено метод запобігання ефекту надмірного регулювання, що за певних умов виникає при автоматичному горизонтальному масштабуванні мікросервісів в системі оркестрації контейнерних застосунків *Kubernetes*. Ефект виявлено, спочатку, лише на тривалих *HTTP* сесіях *WebSocket*, на яких він призвів і як до надмірного використання обчислювальних ресурсів, що знизило ефективність управління *IT*-інфраструктурою, так і спричинило відмову сервісу. Виявлено, що ефект надмірного регулювання відтворюється не тільки для з'єднань із довгостроковими *HTTP*-сесіями як-то *HTTP WebSocket*, так і для більш короткотривалих *REST HTTP* сесій за умови збільшення затримки в циклі збору метрик, що використовувались для горизонтального автомасштабування. Зроблено припущення, що це пов'язано з підходом реалізації контролерів горизонтального масштабування схожим до принципів роботи пропорційного регулятора у системах з негативним зворотнім зв'язком з теорії автоматики і управління. Запропоновано вирішити проблему одним з методів оптимізації пропорційного регулятору, а саме зменшення часової затримки управління автоматичного масштабування в *Kubernetes*. Застосований метод продемонстрував ефективність, тому у межах тієї ж методології був проведений експеримент з застосування пропорційно-інтегрально-диференціального регулятору для автоматичного горизонтального масштабування подів. Отримані результати показали, чому пропорційно-інтегрально-диференціальний**

регулятор не поширений серед розглянутих рішень *Kubernetes* для горизонтального автоматичного масштабування, і зроблено припущення про обмеження дослідження зворотного до масштабування процесу згортання кількості реплік подів в *Kubernetes* через необхідність врахування якості обслуговування при зупинці подів і необхідності збору метрик індикаторів засобами управління об'єктами якості сервісу як то *ISTIO*.

Ключові слова: Kubernetes, мікросервіси, горизонтальне автомасштабування, пропорційне регулювання, хмарні обчислення.

Вступ

Kubernetes є найвідомішою платформою для керування контейнерними додатками [1]. Існує три типи автомасштабування у Kubernetes: Horizontal Pod Autoscaler (HPA), Vertical Pod Autoscaler (VPA) і Cluster Autoscaler (CA) [2].

Під час масштабування вгору HPA створює нові репліки подів для спільного розподілення робочих навантажень, не впливаючи на сессії, які вже створені для існуючих подів [2]. Це дозволяє динамічно регулювати кількість реплік подів на основі спостережуваного використання ЦП або інших ресурсів без необхідності перезапуску вже працюючих поди і збросу поточних сесій. І це робить HPA одним з найкращих рішень автомасштабування у контексті підтримки рівня якості обслуговування користувача, не скидаючи існуючі сеанси під час зростаючого навантаження, принаймні під час процесу масштабування вгору.

Однак [3] стверджує, що під час керування масштабуванням за допомогою HPA в середовищі Kubernetes кількість реплік може почати коливатися за деяких обставин, тобто виникає ефект, що називається тремтінням або плесканням. Це призводить до неефективного використання ресурсів через надмірну кількість подів, а також до порушення якості послуг, що надаються додатками, встановленими на цих подах. У [7] описано приклад сценарію подібного перегулювання для довготривалих TCP сессій на WebSocket.

Як нативний HPA контролер у Kubernetes [3], так і KEDA [6], розроблений CNCF [3], визначають таргетну кількість реплік мікросервісу із пропорційним масштабуванням на основі деякої метрики, на яку посилаються, перетворену на ціле значення або отриману як ціле – тобто є типовими пропорційними регуляторами. Нативний контролер Kubernetes HPA [3] має механізм обмеження швидкості масштаування від часу. І це єдиний доступний на даний момент механізм, описаний як метод оминання надмірного регулювання [3]. Інший відомий механізм підстройки пропорційного регулятора: налаштування часових параметрів – але його не згадано в офіційній документації [3], вірогідно, через несприйняття розробниками Kubernetes HPA контролеру як пропорційного регулятору – хоча така можливість і існує.

Це породжує проблему неповноти описаних методів оптимізації HPA та уникнення його надмірного регулювання, як типову проблему пропорційного регулятора [4]. І про необхідність дослідження методу налаштування HPA як пропорційного регулятора, щоб довести, що ефект надмірного регулювання відтворюється також для короткочасних сеансів, таких як HTTP REST, а не лише для WebSocket [7].

Результати дослідження

З урахуванням формули (1) описаної у [7] між доступними часовими параметрами для налаштування регулятору було вибрано t_r (час для відновлення показників, визначено за допомогою опції контролера «період синхронізації метрик HPA») як найлегший для впливу в лабораторії. З таким припущенням ми отримуємо:

$$N_d(t) = \left\lceil \frac{W_c(t - t_{up})}{R_c(t)} \cdot \frac{M_n - \Delta M(t_r)}{M_p} \right\rceil, \text{ for upscaling } M_n - \Delta M(t_r) > M_p. \quad (1)$$

Для Kubernetes зміна цього параметру вимагає перезапуску кластера. Але це значно спрощує сам експеримент без необхідності налаштування програми і не вимагає додавання штучних затримок для відтворення корпусу в кластерах з невеликим фактичним навантаженням (доступно для студентських експериментальних стендів). Збільшення часу отримання метрики tr (1) дорівнює зміні часу реакції, що також робить його критерієм можливості розбалансування традиційного пропорційного регулятора.

У наступних експериментах використовувався додаток, що імітує обробку суміші викликів REST HTTP GET POST (імітація запитів користувача). Набір запитів-навантаження однаковий у всіх експериментах. Навантаження робиться нерівномірним протягом усього часу експерименту, щоб досягти ефекту зростаючого сплеску затримки обслуговування. Додаток запущено в одно потоковому режимі з архітектурним обмеженням виконання в 1 ЦП (vCPU). Усі модулі були розміщені для виконання на одному вузлі (ноді), щоб виключити різницю затримки маршрутизації трафіку. Кількість вільних віртуальних ЦП (vCPU) була набагато більшою, ніж фактично використана програмою, щоб виключити це обмеження. Додаток було розгорнуто у двох репліках, що дозволило перевірити балансування на початковому стані, але зробило марною для аналізу частину діаграм початкового завантаження (перші приблизно 10 секунд експериментів). У всіх випадках використовувалися наступні критерії для масштабування програми: середній сплеск використання vCPU до 80% для всіх запущених екземплярів.

На рис. 1 зображено 2 хвилини експерименту для випадку, коли горизонтальний модуль автомасштабування встановлено на затримку отримання метрик у 20 секунд, що не є значенням за замовчуванням.

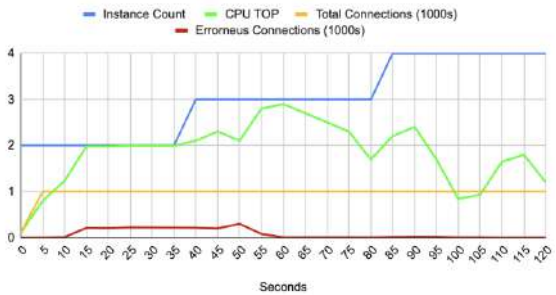

Рис. 1. Кількість подів за часом експерименту з навантаженням для 20-секундного оновлення метрики

Блакитна лінія фактичної кількості модулів має показувати функцію цілого із округленням у 80% від середнього максимуму ЦП для поду, але починаючі з 80ї секнуди демонструє фактичне перегулювання – збільшення кількості реплік подів понад необхідної для задоволення запиту.

Для наступного випадку, описаного на рис. 2, час реакції зменшено до 15 секунд, що дозволяє досягти фактичного падіння навантаження на 75-й секунді та не викликати надмірну реакцію, що спричиняє надмірне регулювання.

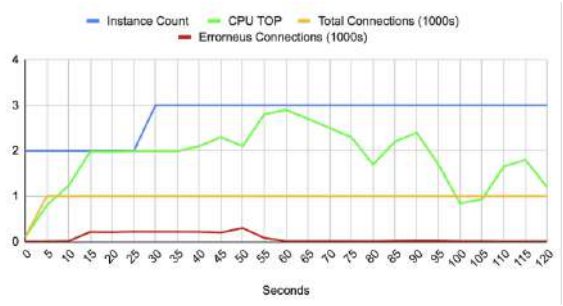

Рис. 2. Кількість подів за часом експерименту з навантаженням для 15-секундного оновлення метрики

У пропорційних системах керування, таких як Kubernetes HPA, часовий лаг відіграє вирішальну роль у якості керування [3]. Значний проміжок часу між виявленням відхилення від заданого значення та початком коригувальних дій може поставити під загрозу здатність системи підтримувати стабільність і бажані рівні продуктивності. Тривале відставання може призвести до недостатнього контролю, коли коригувальна дія затримується, що призводить до погіршення продуктивності або виснаження ресурсів. І навпаки, у міру того, як часовий лаг стає коротшим, система стає більш чутливою, але це також може призвести до надмірного контролю.

На рисунку 3 стандартний модуль замінено на такий самий з ПІД регулятором і затримкою обробки метрик у 1 секунду. Також на рисунку 3 можна звернути увагу на все ж наявність сплеску помилок з 80 до 85 секунди, відсутні на рисунках 1 і 2.

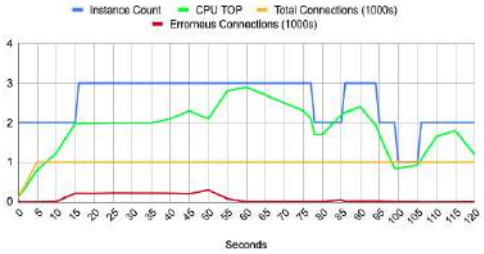

Рис. 3. Кількість модулів за часом експерименту з навантаженням для 1-секундного відновлення метрики під PID

Огляд результатів

У сфері наукових досліджень щодо оптимізації контейнерних середовищ припущення про горизонтальне автомасштабування Kubernetes (HPA) на основі зібраних показників можна порівняти з використанням пропорційного регулятора. Подібно до того, як пропорційний регулятор динамічно регулює свій вихід у відповідь на зміни в сигналі помилки, HPA динамічно масштабує кількість реплік модулів на основі спостережуваних показників, таких як використання ЦП або користувальницькі показники продуктивності. Основна діаграма представлена на рис. 1.

Це аналогічно принципам систем керування зі зворотним зв'язком у техніці, де механізми пропорційного керування спрямовані на підтримку бажаної заданої точки шляхом безперервного моніторингу та регулювання параметрів системи. У контексті Kubernetes HPA діє як автоматизований цикл зворотного зв'язку, гарантуючи, що розподіл ресурсів програми відповідає поточному рівню попиту. Співвідносячи зібрані показники з рішеннями щодо масштабування, HPA оптимізує використання ресурсів, покращує чуйність системи та сприяє ефективній роботі контейнерних робочих навантажень. Таким чином, це приклад застосування принципів теорії управління в сучасних хмарних обчислювальних середовищах.

Дуже близькі результати експериментів для ПІД-оптимізованого регулятора на рис. 1 і оптимізованого за часом регулятора на рис. 3. Вони показують, що часовий проміжок між збором метрик і застосуванням мінімізації може бути достатнім для досягнення напівоптимального рівня регулювання автомасштабування для деяких умов. Це також може бути поясненням того, чому більшість поширених рішень автомасштабування не за замовчуванням, таких як KEDA та стандартний автомасштабатор Kubernetes, використовують лише просту логіку пропорційного регулятора.

Висновки

Приведені дослідження показали, що горизонтальне автомасштабування в Kubernetes може відтворювати ефект коливань навіть для короткочасних викликів HTTP REST API, а не лише для довгострокових WebSocket сессій як було виявлено у [6]. Підтверджено, що логіка HPA сильно залежить від часового проміжку між збором показників і застосуванням рішення про масштабування майже так само, як і пропорційний регулятор. Доведено метод налаштування затримки HPA шляхом мінімізації часу отримання метрики. Дефолтне HPA Kubernetes імплементується пропорційним контролером із логікою негативного зворотного зв'язку, і він міг би не відтворювати коливання та досягати точності масштабування за умови наявності вбудованого ПІД-алгоритму. Але дослідження показали також, що налаштування часової затримки може бути достатнім для досягнення стабільного процесу масштабування без перегулювання, і пояснює, чому найчастіше контролери з автомасштабування, такі як KEDA, все ще використовують лише просту логіку пропорційного регулятора.

Література

1. Статті з журналів (у тому числі електронних):

[1] E. Truyen, D.V. Landuyt, D. Preuveneers, B. Lagaisse W. Joosen "A Comprehensive Feature Comparison Study of Open-Source Container Orchestration Frameworks", Appl. Sci. 2019, 9(5), pp. 35–43; doi: 10.3390/app9050931.

[2] D.V. Martins "Scaling of Applications in Containers". Master's thesis, 2023. pp. 13–15.

[3] T. -T., Nguyen, Y. -J. Yeom, T. Kim, D. -H. Park, and S. Kim, "Horizontal Pod Autoscaling in Kubernetes for Elastic Container Orchestration". Sensors, 20(16), 4621. 2020 pp. 13-15; doi: 10.3390/s20164621.

[4] S. Singh, C. H. Muntean and S. Gupta, "Boosting Microservice Resilience: An Evaluation of Istio's Impact on Kubernetes Clusters Under Chaos", 2024, 9th International Conference on Fog and Mobile Edge Computing (FMEC), Malmö, Sweden, 2024, pp. 245–252, doi: 10.1109/FMEC62297.2024.10710237.

[5] K. Johan, Å. Richard, M. Murray. "Feedback Control Systems", 1963, p. 295 ISBN-13: 978-0-691-13576-2.

[6] D.K. Alqahtani, A.N. Toosi, "Container Orchestration in Heterogeneous Edge Computing Environments. InResource Management in Distributed Systems", 2024 pp. 151–168. doi: 10.1007/978-981-97-2644-8_8, ISBN: 978-981-97-2643-1.

[7] O. Rolik and V. Volkov, "Method of horizontal pod scaling in Kubernetes to omit overregulation", Inf. Comput. and Intell. syst. j., no. 5, pp. 55–67, Dec. 2024

Task allocation in hierarchical IoT systems based on osmotic computing

Nahaiko Dmytro, Rolik Oleksandr
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
d.nahaiko@kpi.ua, o.rolik@kpi.ua

Abstract. **A task allocation method for IoT systems with a three-tier architecture – cloud, fog, and edge computing – is proposed. The method is based on the concept of osmotic computing and determines the optimal execution environment for tasks by considering their characteristics and requirements and the current state of computing resources. A hierarchical control model is developed, in which each level includes computing nodes and a management system responsible for task allocation, resource state monitoring, and decision-making regarding the placement, migration, and removal of microelements.**

Keywords*: *information systems and technologies, internet of things (IoT), cloud computing, fog computing, edge computing, osmotic computing, task allocation.

Розподіл задач в ієрархічних IoT-системах на основі осмотичних обчислень

Нагайко Дмитро, Ролік Олександр
КПІ імені Ігоря Сікорського
м. Київ, Україна
d.nahaiko@kpi.ua, o.rolik@kpi.ua

Анотація. **Запропоновано метод розподілу задач в IoT-системах, які мають трирівневу архітектуру – хмарних, туманних, крайових обчислень. Метод побудовано на концепції осмотичних обчислень, а для визначення оптимального середовища виконання задач він враховує як характеристики та вимоги самих задач, так і стан обчислювальних ресурсів. Розроблено ієрархічну модель керування, коли кожен рівень містить обчислювальні вузли та систему управління, що здійснює розподіл задач, моніторинг стану ресурсів та прийняття рішень щодо розміщення, міграції та видалення мікроелементів.**

Ключові слова: інформаційні системи та технології, інтернет речей (IoT), хмарні обчислення, туманні обчислення, крайові обчислення, осмотичні обчислення, розподіл задач.

Вступ

Інтернет речей (IoT) застосовується в різних сферах людської діяльності, надаючи інструменти для автоматизації, моніторингу й керування різноманітними процесами в реальному часі.

Активний розвиток та впровадження IoT-технологій призвели до стрімкого зростання кількості пристроїв, підключених до IoT: лише у 2024 р. їхній парк перевищив 18 млрд, що на 13% більше ніж у 2023 р., а корпоративні витрати на IoT-інфраструктуру сягнули 298 млрд дол. США [1]. Ця тенденція супроводжується збільшенням обсягів даних, які генеруються IoT-пристроями, та зростанням обчислювального навантаження на IT-інфраструктуру, що ускладнює підтримання гарантованого рівня якості обслуговування (QoS). Окрім цього, висока гетерогенність та динамічність, якими характеризуються IoT-середовища [2], вимагають відповідного рівня адаптивності IoT-системи до змін в реальному часі.

Використання туманних та крайових обчислень частково вирішує зазначені проблеми, розширяючи обчислювальні потужності системи до краю мережі, тим самим наближаючи процеси збору, аналізу та оброблення даних ближче до джерела цих даних, що сприяє мінімізації затримок передавання даних, зменшує загальний час виконання задач та знижує навантаження у хмарному середовищі [3].

Однак застосування туманних та крайових технологій створюють нові проблеми, які пов’язані з нерівномірним навантаженням, обмеженими ресурсами на периферії та потребою реагувати на динамічні підключення або відключення та вихід з ладу обчислювальних вузлів.

Для вирішення цих проблем в [4] запропоновано концепцію осмотичних обчислень, яка запозичена з хімічного процесу осмосу і полягає в автономному та динамічному управлінні обчислювальними ресурсами. Завдяки безперервному вертикальному балансуванню навантаження між хмарним, туманним та крайовим середовищами, осмотичні

обчислення забезпечують адаптацію системи до змін в реальному часі, підтримуючи рівномірний розподіл навантаження між різними обчислювальними середовищами [4, 5].

Однак, з урахуванням концепції осмотичних обчислень, актуальним завданням залишається управління розподілом задач в умовах невизначеності, динамічності та гетерогенності IoT-середовища, забезпечуючи при цьому надання послуг з відповідним рівнем QoS.

Особливості загальної архітектури IoT-систем

В роботі розглядається трирівнева архітектура розподілених IoT-систем з поділом на хмарне, туманне та крайове середовища, в якій управління задачами та ресурсами базується на концепції осмотичних обчислень (рис. 1).

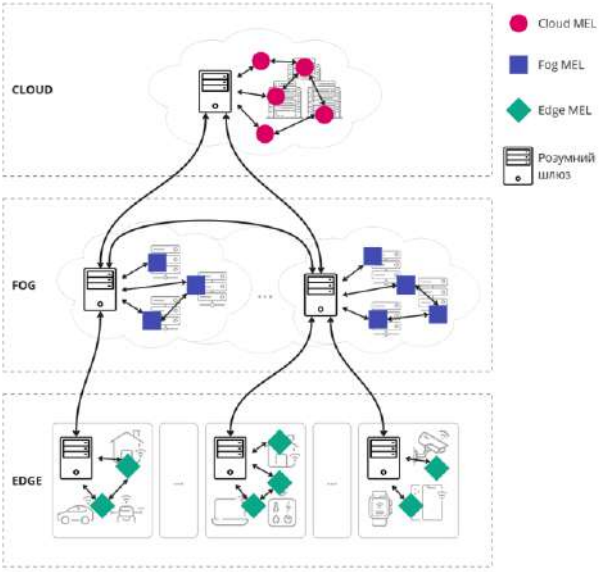

Рис. 1. Загальна структура трирівневої IoT-системи на основі осмотичних обчислень

Перший рівень – Edge, який знаходиться найнижче в ієрархічній структурі, є середовищем крайових обчислень. Цей рівень розташований найближче до джерел даних, якими є кінцеві IoT-пристрої, частина з яких мають власні обчислювальні потужності, що дозволяє їм виконувати прості задачі з низькою затримкою.

Другий рівень – середовище туманних обчислень (Fog), містить проміжні обчислювальні ресурси на рівні мікро- або регіональних дата-центрів. Завдяки розширеним обчислювальним можливостям Fog-рівень здатний обробляти більші об'єми даних та виконувати складніші обчислення порівняно з Edge-рівнем. Fog-рівень може одночасно обслуговувати декілька доменів Edge-рівня.

Третій, найвищий рівень – це хмарне середовище (Cloud), яке розташоване найдалі від кінцевих пристроїв та побудоване на основі центрів оброблення даних з великими обчислювальними потужностями. Cloud-рівень забезпечує аналіз, обробку та збереження великих масивів даних з високим рівнем відмовостійкості. Cloud-рівень може одночасно обслуговувати декілька доменів Fog-рівня.

Взаємодія між обчислювальними середовищами реалізується як ієрархічно організований обмін даними між рівнями IoT-системи.

Ідея осмотичних обчислень запозичена з хімічного процесу осмосу, де розчинник переходить із ділянки з нижчою концентрацією розчиненої речовини у ділянку з вищою концентрацією розчиненої речовини через напівпроникну мембрану, завдяки чому вирівнюється концентрація по обидва боки мембрани. В осмотичних обчисленнях в ролі розчинника виступають мікроелементи (MicroElements, MELs), які можуть мігрувати між різними середовищами (хмара, туман, край) через програмно-визначену мембрану (Software-Defined Membrane, SDMem).

MELs поділяються на мікросервіси (MicroServices, MS), які надають функціональні можливості, та мікродані (MicroData, MD), які передаються між компонентами IoT-системи.

Програмно-визначена мембрана регулює переміщення MELs відповідно до різних політик управління, враховуючи доступність обчислювальних ресурсів, вимоги до якості обслуговування та поточний стан системи [5, 6].

В архітектурі IoT-систем на рис.1 реалізована трирівнева ієрархічна модель керування. Кожен рівень містить обчислювальні вузли та систему управління, яка на основі аналізу даних моніторингу стану ресурсів приймає рішення щодо розміщення, міграції та видалення MELs й виконує розподіл задач (рис. 2).

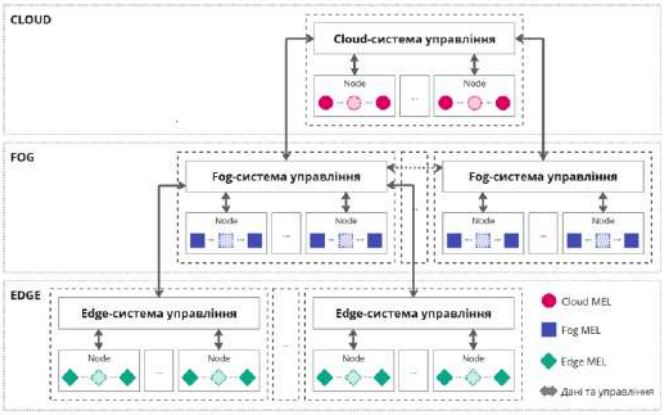

Рис. 2. Загальна модель розподілу задач та управління ресурсами в IoT-системі

На обчислювальному вузлі може бути розгорнуто один або декілька MELs залежно від наявних потужностей, а кожен MEL може виконувати одну або декілька задач.

Система управління нижче розташованого рівня підпорядковується вищій за ієрархією системі управління. Так Edge-система управління обслуговує обчислювальні вузли, що взаємодіють з кінцевими пристроями IoT-систем. Fog-система управління координує edge-домени – групи обчислювальних вузлів, що підпорядковуються одній системі управління, – та обчислювальні вузли свого домену. Центральна система управління, яка розташована на хмарному рівні, має інформацію про загальний стан IoT системи, забезпечуючи узгоджену роботу всіх рівнів.

Розглянута модель керування реалізує комбінацію централізованого та децентралізованого підходів до управління. Глобальне управління та прийняття рішень здійснюється центральною системою управління. В той же час децентралізоване управління здійснюється системами управління відповідних рівнів, які приймають локальні рішення і здійснюють управління в межах свого домену. Поєднання цих підходів дозволяє підвищити відмовостійкість, масштабованість та адаптивність системи.

Метод розподілу задач в IoT-системі

У розглянутій трирівневій архітектурі розподіленої IoT-системи на основі осмотичних обчислень фактичне середовище виконання задач побудовано на основі MEL. MEL ро-

згортається на обчислювальному вузлі $N_i \in \mathrm{N}_L$ з ресурсами $R_i \in \mathrm{R}_L$, де N_L – множина обчислювальних вузлів з сумарними ресурсами R_L на рівні $L \in \mathrm{L}$, $\mathrm{L} = \{Edge, Fog, Cloud\}$, $i = \overline{1, n_L}$, де n_L – кількість доступних вузлів на рівні L. При цьому обчислювальні вузли в межах одного рівня можуть мати різні ресурси $R_i \neq R_m$, $R_i \in \mathrm{R}_L, R_m \in \mathrm{R}_L$ для деяких $i \neq m$, де $i = \overline{1, n_L}, m = \overline{1, n_L}$.

У запропонованому методі розподіл задач $T_j \in \mathrm{T}$, $j = \overline{1, J}$ здійснюється за алгоритмом 1.

Алгоритм 1. Розподіл задачі в ієрархічній IoT-системі

Input: Task T_j, set of nodes $\mathrm{N} = \bigcup_{L \in \mathrm{L}} \mathrm{N}_L$ with resources $\mathrm{R} = \bigcup_{L \in \mathrm{L}} \mathrm{R}_L$, $\mathrm{L} = \{Edge, Fog, Cloud\}$

Output: Assignment of task T_j to a node or queue/reject the task

```
L ← level where T_j was initialized
while L ∈ L do
    if isLevelSuitableForTask(T_j, L) then
        for each N_i from N_L do
            if isNodeAvailableForTask(T_j, R_i) then
                assign T_j to N_i node
                return
            end if
        end for
        for each N_i from N_L do
            if canReleaseResourcesForTask(T_j, N_i) then
                release resources on N_i node
                if isNodeAvailableForTask(T_j, R_i) then
                    assign T_j to N_i node
                    return
                end if
            end if
        end for
    end if
    L ← L' ∈ L \{L}
end while
queue or reject T_j
```

Реєстрація задачі $T_j \in \mathrm{T}$, $j = \overline{1, J}$ відбувається в локальній системі управління того рівня $L \in \mathrm{L}$, на якому вона була ініційована. Кожна задача описується вектором параметрів $P_j = \{ p_j^k \mid k = \overline{1, K} \}$, де p_j^k – значення k-го параметра задачі T_j. До таких параметрів можуть належати: пріоритет задачі, вимоги до затримок, обчислювальна складність тощо. Кількість та типи параметрів можуть відрізнятися в залежності від вимог та потреб конкретної IoT-системи.

Після ініціації задачі здійснюється аналіз придатності поточного рівня L для виконання задачі T_j: $f_L(T_j, L) \rightarrow f_L(P_j, L)$. Якщо поточний рівень задовольняє вимоги виконання, система управління рівня переходить до перевірки наявності необхідних обчислювальних ресурсів у такий спосіб: для кожного обчислювального вузла $N_i \in \mathrm{N}_L$ з ресурсами $R_i \in \mathrm{R}_L$ здійснюється оцінка за деякою функцією $f_R(T_j, R_i)$, що характеризує можливість розміщення задачі T_j на вузлі N_i.

У випадку достатньої кількості ресурсів задача призначається відповідному MEL для виконання. Якщо ресурсів недостатньо, система управління рівня L здійснює аналіз доцільності вивільнення ресурсів, оцінюючи пріоритети поточних задач та потенційні втрати від їх переміщення. При позитивному результаті цього аналізу виконується вивільнення ресурсів, після чого знову перевіряється наявність необхідних ресурсів.

Якщо поточний рівень L не відповідає вимогам або неможливо вивільнити достатньо ресурсів, система управління рівня перевіряє чи залишилися інші нерозглянуті рівні в ієрархії $L' \in \mathrm{L} \setminus \{L\}$, куди можна розподілити задачу. При наявності альтернативних варіантів задача перенаправляється на вищий або нижчий рівень в залежності від параметрів P_j самої задачі, після чого процес аналізу повторюється. Якщо доступних рівнів не залишилось, задача поміщається в чергу очікування або відхиляється в залежності від політик, які реалізовано у системі управління.

Розглянутий метод розподілу задач спрямований зменшити загальний час виконання задач за рахунок їх раціонального розміщення, обираючи середовище для виконання з урахуванням параметрів задач та поточного навантаження.

Висновки

В роботі розроблено загальну модель розподілу задач та управління ресурсами в IoT-системах на основі осмотичних обчислень. Розглянута трирівнева архітектура розподіленої IoT-системи спрямована забезпечити ефективне використання обчислювальних ресурсів крайового, туманного та хмарного рівнів, регулювати розподіл навантаження між ними та підвищити адаптивність системи до динамічних змін середовища. Запропоновано метод розподілу задач, який враховує як характеристики та вимоги самих задач, так і стан обчислювальних ресурсів для визначення раціонального середовища виконання.

Подальші кроки досліджень будуть спрямовані на розробку методів управління MELs для ефективного використання ресурсів, балансування навантаження та зменшення часу виконання задач, а також дослідження застосування апарату нечіткої логіки з метою забезпечити адаптивність IoT-системи в умовах динамічних змін та невизначеності середовища.

Література

1. Cole C. IoT 2024 in review: The 10 most relevant IoT developments of the year. *IoT Analytics*. 15.01.2025. URL: https://iot-analytics.com/iot-2024-review (access date: 27.04.2025).

2. Self-adaptive architectures in IoT systems: a systematic literature review / I. Alfonso et al. *Journal of Internet Services and Applications*. 2021. Vol. 12. P. 1–28. URL: https://doi.org/10.1186/s13174-021-00145-8.

3. Rolik O., Telenyk S., Zharikov E. IoT and Cloud Computing: The Architecture of Microcloud-Based IoT Infrastructure Management System. *Securing the Internet of Things: Concepts, Methodologies, Tools, and Applications*. Hershey, PA: IGI Global, 2020. C. 1157–1185. URL: https://doi.org/10.4018/978-1-5225-9866-4.ch052.

4. Osmotic Computing: A New Paradigm for Edge/Cloud Integration. *IEEE Cloud Computing*. 2016. Vol. 3, iss. 7. P. 76–83. URL: https://doi.org/10.1109/MCC.2016.124.

5. A Systematic Review on Osmotic Computing / B. Neha et al. *ACM Transactions on Internet of Things*. 2022. Vol. 3, iss. 2. P. 1–30. URL: https://doi.org/10.1145/3488247.

6. Software Defined Membrane: Policy-Driven Edge and Internet of Things Security / M. Villari et al. *IEEE Cloud Computing*. 2017. Vol. 4, iss. 4. P. 92–99. URL: https://doi.org/10.1109/MCC.2017.3791014.

Problems of ensuring fault tolerance of cloud systems based on the estimations of reliability metrics

Yurii Tymoshyn, Bohdan Bachkala
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
y.timoshin@gmail.com, bbachkala@gmail.com

Abstract. This paper addresses the problems of ensuring fault tolerance in Infrastructure as a Service (IaaS) cloud systems based on the estimations of reliability metrics. It proposes a fault-tolerant cloud infrastructure architecture. The analysis covers the impact of Recovery Point Objective (RPO) on potential data loss and Recovery Time Objective (RTO) on permissible service downtime. Results provide guidance for choosing optimal backup strategies.

Keywords: Disaster Recovery, Cloud Computing, RPO, RTO, Fault tolerance, Metrics evaluation

Проблеми забезпечення відмовостійкості хмарних систем на основі оцінок метрик надійності

Тимошин Юрій, Бачкала Богдан
КПІ імені Ігоря Сікорського
м. Київ, Україна
y.timoshin@gmail.com, bbachkala@gmail.com

Анотація. Робота присвячена проблемі забезпечення відмовостійкості хмарних сервісів моделі IaaS (Infrastructure as a Service). Запропоновано архітектуру відмовостійкої хмарної інфраструктури. Досліджено вплив метрик RPO (Recovery Point Objective) на потенційні втрати даних та RTO (Recovery Time Objective) на допустимий час простою. Надано рекомендації щодо вибору оптимальних стратегій резервування.

Ключові слова: Аварійне відновлення, Хмарні обчислення, RPO, RTO, відмовостійкість, оцінювання метрик

Вступ

У сучасних хмарних обчислювальних системах критично важливо забезпечувати безперервність роботи сервісів навіть за наявності відмов. Як свідчать останні дослідження, тривалі відмови критичних сервісів призводять до значних фінансових втрат та підривають довіру користувачів [1]. Для середнього та великого бізнесу швидке відновлення IT-інфраструктури після відмови і повернення до працездатного стану бізнес-додатків та цифрових сервісів є ключовим чинником конкурентоспроможності.

Для кількісної оцінки здатності системи зберігати працездатність в умовах відмов застосовують метрики надійності, серед яких основними є RTO – цільовий час відновлення, та RPO – цільова точка відновлення [2]. Значення цих метрик визначають вимоги до архітектури відмовостійкої інфраструктури та безпосередньо впливають на вибір технологій резервування, реплікації та процедур відновлення даних.

Водночас для комплексної оцінки якості надання хмарних послуг важливо враховувати не лише метрики надійності, але й показники доступності, що визначають, наскільки стабільно система забезпечує користувачам доступ до її функціоналу. Основними показниками доступності є SLA (Service Level Agreement) – угода про рівень обслуговування, що фіксує зобов'язання постачальника щодо якості сервісу, та SLO (Service Level Objective) – конкретні цільові значення доступності, які дозволяють контролювати відповідність системи встановленим стандартам [3].

Метою роботи є аналіз та моделювання впливу метрик надійності (RTO, RPO) на показники доступності (SLA, SLO). Отримані результати дослідження дозволять розробити рекомендації щодо вибору оптимальних стратегій ре-

зервування у хмарних системах моделі IaaS, які забезпечують баланс між витратами на інфраструктуру та очікуваннями користувачів.

Архітектура відмовостійкої хмарної інфраструктури

Відмовостійка хмарна інфраструктура будується за принципом багаторівневої архітектури з резервуванням на всіх критично важливих рівнях, що забезпечує безперервність роботи сервісів навіть у випадку відмов окремих компонентів [4].

На рис. 1 показано схему реалізації відмовостійкої хмарної IaaS-архітектури, що використовувалася як модель дослідження. Запити від веб-користувачів або сервісів спочатку надходять до хмарного сервісу доменних імен (DNS, Domain Name System), який спрямовує трафік на основний регіон. У основному регіоні балансувальник навантаження, розгорнутий у декількох зонах доступності (AZ, Availability Zone), розподіляє запити між віртуальними машинами (ВМ) у групі автоматичного масштабування. У разі повної відмови основного регіону DNS автоматично перенаправляє запити до резервного регіону відновлення (DR, Disaster Recovery), мінімізуючи час простою.

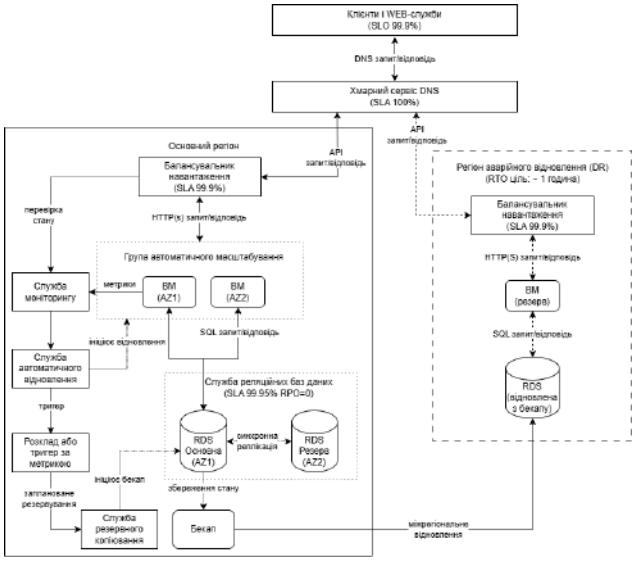

Рис. 1. Структурна схема відмовостійкої хмарної інфраструктури

На мережевому рівні балансувальник навантаження виконує функцію розподілу запитів між кількома екземплярами сервісу та слідкує за їх станом. За допомогою вбудованих перевірок стану балансувальник автоматично припиняє надсилати запити на будь-який екземпляр, що не відповідає або перевищує допустимий час відгуку, перенаправляючи запити до інших, справних вузлів [5]. Такий підхід гарантує, що відмова окремого вузла не вплине на доступність сервісу загалом.

Група автоматичного масштабування забезпечує резерв обчислювальних ресурсів і адаптацію до зміни навантаження. Система автоматично додає нові екземпляри ВМ при зростанні інтенсивності запитів і вилучає зайві екземпляри при спаді навантаження. Таке горизонтальне масштабування дозволяє підтримувати заданий рівень продуктивності та реагувати на пікові навантаження без втрати доступності.

Сервіс резервного копіювання відповідає за захист даних і відновлення у випадку відмов. Резервні копії створюються регулярно за розкладом відповідно до визначеної політики (наприклад, щогодини або щодоби) і зберігаються у віддаленому сховищі. За потреби дані з резервних копій використовуються для відновлення системи. Поєднання постійної реплікації даних між регіонами з регулярним резервним копіюванням дозволяє мінімізувати втрати даних і забезпечити повне відновлення системи після масштабних відмов. Зазначені підходи (синхронна реплікація, дублювання компонентів) відповідають основним принципам забезпечення відмовостійкості, описаним у сучасній літературі [4,5].

Запропонована архітектура закладає основу для подальшого моделювання взаємозв'язків між показниками надійності та доступності. У наступному розділі досліджено, як вибір значень RTO і RPO впливає на цільові показники якості обслуговування (SLO) та ризики втрати даних.

Моделювання впливу метрик надійності на доступність системи

Цільова точка відновлення даних (RPO) визначає, об'єм даних який система може потенційно втратити при відмовах, і безпосередньо залежить від частоти створення резервних копій. В цьому розділі розглядаються дві стратегії резервування: лінійна стратегія з фіксованим інтервалом між копіями та експоненціальна стратегія, що базується на випадкових інтервалах створення резервних копій.

Лінійна стратегія резервування передбачає створення резервних копій з фіксованим інтервалом *T*. У цьому випадку максимальний обсяг втрати даних дорівнює інтервалу між копіями, а ймовірність перевищення допустимого порогу RPO лінійно зменшується зі зменшенням інтервалу *T*. Імовірність того, що втрачений обсяг даних перевищить RPO, визначається за формулою (1):

$$P\{втрата\ даних\ >\ RPO\} = \left(1 - \frac{RPO}{T}\right) \times 100\% \quad (1)$$

де RPO – допустимий час втрати даних; T – середній інтервал часу між резервними копіями.

З формули (1) видно, що для лінійної стратегії резервування ризик перевищення RPO лінійно спадає від 100% (коли $RPO \to 0$, тобто частота резервного копіювання не покриває допустимий час втрати даних) до 0% при $RPO = T$. Якщо інтервал між копіями не перевищує допустимий час втрати даних, ймовірність втратити більше даних, ніж дозволяє RPO, стає нульовою. В цьому випадку частота резервування повністю покриває допустиме «вікно» втрати даних. Така стратегія є оптимальною для систем з суворими вимогами до цілісності та збереження даних, наприклад, для фінансових платформ, де навіть мінімальна втрата даних є неприпустимою через значні репутаційні та фінансові ризики.

Експоненціальна стратегія базується на ймовірнісних моделях, де інтервали між створенням резервних копій мають експоненціальний розподіл. При цьому ймовірність перевищення допустимого RPO експоненціально зменшується зі скороченням середнього інтервалу резервування, але ніколи не досягає нуля, залишаючи ненульовий ризик втрати даних. Імовірність втрати даних для цієї стратегії визначається за формулою (2):

$$P\{втрата\ даних\ >\ RPO\} = \left(e^{-\frac{RPO}{T}}\right) \times 100\% \quad (2)$$

де RPO – допустимий час втрати даних; T – середній інтервал часу між резервними копіями.

Навіть за незначної середньої тривалості інтервалу *T* відповідно до формули (2) існує невелика, але відмінна від нуля ймовірність того, що інтервал між копіями виявиться довшим за допустиме значення RPO. Ця стратегія є менш

вимогливою до ресурсів і може бути доцільною для систем, де допускається мінімальний ризик втрати даних, наприклад, для некритичних аналітичних платформ.

На рис. 2 наведено графік залежності відносного ризику втрати даних від співвідношення між допустимою втратою даних та частотою резервного копіювання для різних стратегій резервування.

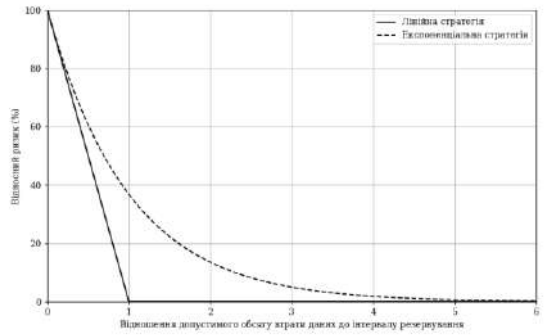

Рис. 2. Відносний ризик втрати даних для різних стратегій резервування

Рис. 2 демонструє відмінності між двома стратегіями: лінійна стратегія здатна забезпечити відсутність ризику втрати даних, тоді як експоненціальна стратегія допускає залишковий ризик. Мінімізація RPO є необхідною умовою забезпечення цілісності даних і важливим компонентом загальної стратегії забезпечення відмовостійкості системи.

Іншим ключовим показником надійності є цільовий час відновлення (RTO), що визначає максимально допустиму тривалість простою системи після відмови. RTO безпосередньо впливає на цільові показники доступності системи (SLO). Фактичний рівень доступності системи визначається за формулою (3):

$$SLO = \left(1 - \frac{N \times RTO}{T_{період}}\right) \times 100\% \qquad (3)$$

де N – кількість відмов на рік; RTO – середній час відновлення (годин); $T_{період}$ – тривалість розглянутого періоду.

Із наведеної залежності (3) видно, що для систем із високою частотою відмов критично важливим є зменшення RTO. Водночас, для систем із низькою інтенсивністю відмов допустимий час відновлення може бути більшим без суттєвого впливу на показники доступності.

На рис. 3. наведено графік залежності цільового рівня обслуговування від цільового часу відновлення для трьох різних частот відмов: 1, 5 та 10 відмов на рік.

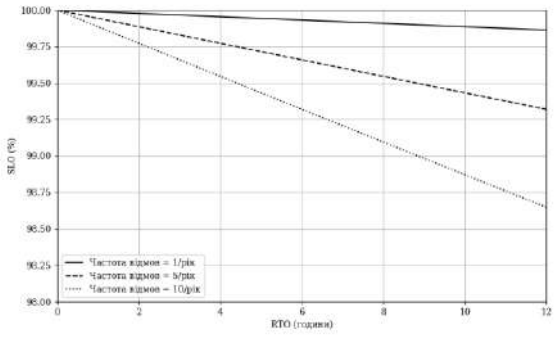

Рис. 3. Залежність SLO від RTO та частоти відмов

Рис. 3 демонструє, що для систем із низькою частотою відмов (1 на рік) SLO залишається стабільно високим навіть при тривалому RTO. Це означає, що поодинокі відмови суттєво не впливають на загальний показник доступності за умови помірного часу відновлення. Водночас, для систем з високою частотою відмов (10 на рік) швидке відновлення є критичним для підтримки високого рівня доступності. Така залежність підкреслює необхідність балансу між надійністю системи (зменшення частоти відмов) і швидкістю відновлення (зменшення RTO). Для підтримки систем із високою частотою відмов необхідно інвестувати в автоматизовані процедури відновлення, тоді як для систем із рідкісними відмовами можна дозволити більший RTO без значного впливу на доступність.

Висновки

В роботі досліджено проблему забезпечення відмовостійкості у хмарних обчислювальних системах. Встановлено, що вибір цільової точки відновлення (RPO) істотно впливає на ризик втрати даних. Лінійна стратегія резервування гарантує прогнозоване зменшення ризику втрати даних, але потребує значних ресурсних інвестицій. Натомість експоненціальна стратегія є більш економічною з точки зору ресурсів, але допускає наявність залишкового ризику через стохастичність інтервалів створення копій. Аналіз залежності цільових показників рівня обслуговування (SLO) від часу відновлення (RTO) та частоти відмов показав необхідність чіткого планування цільових показників доступності та відповідних інвестицій у засоби автоматизованого відновлення та підвищення надійності компонентів.

Отримані результати можуть бути використані компаніями та SRE-інженерами (Site Reliability Engineering) при плануванні та виборі параметрів RPO та RTO і побудови архітектур аварійного відновлення (DR) для своїх сервісів. Це дозволить визначити частоту резервного копіювання, рівень реплікації та необхідні засоби автоматичного відновлення, щоб забезпечити задані показники доступності без надлишкових витрат.

Література

1. Uptime Institute. 2022 Outage Analysis: Increasing downtime costs and consequences [Електронний ресурс]. – 2022. – Режим доступу: https://uptimeinstitute.com/resources/research-and-reports (дата звернення: 02.05.2025).

2. Wilson M. Establishing RPO and RTO Targets for Cloud Applications [Електронний ресурс] // AWS Blog. – 2022. – Режим доступу: https://aws.amazon.com/blogs/ (дата звернення: 02.05.2025).

3. Google Cloud. Disaster recovery planning guide – RTO, RPO and SLA/SLO [Електронний ресурс]. – 2023. – Режим доступу: https://cloud.google.com/architecture/disaster-recovery-planning-guide (дата звернення: 02.05.2025).

4. Hasan M., Goraya M. S. Fault tolerance in cloud computing environment: A systematic survey // Computers in Industry. – 2018. – Т. 99. – С. 156–172. – DOI: 10.1016/j.compind.2018.03.027.

5. Gandham P., Damodaram R., Randhawa S. Preparing for the Unexpected Outages for Your Mission-Critical Cloud Infrastructure with Confidence // International Journal of Computer Trends and Technology. – 2024. – Т. 72, № 8. – С. 134–141. – DOI: 10.14445/22312803/IJCTT-V72I8P12

The Impact of AI-Powered Development Tools on Architecture and Testing of High-Load Information Systems

Kornaha Yaroslav, Oleksii Andrii
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

This paper investigates the impact of artificial intelligence-based development tools, such as Cursor, on the architecture and testing processes of high-load information systems. The study shows that the use of AI assistants facilitates modular code structures, accelerates test generation, and improves system resilience. Key architectural patterns that enhance compatibility with AI-supported development workflows are analyzed, along with the quality and relevance of automatically generated tests and their influence on system sustainability.

***Keywords*: AI-assisted development, high-load systems, software architecture, automated testing, system resilience, Cursor, test generation**

Вплив інструментів з підтримкою штучного інтелекту на архітектуру та тестування високонавантажених інформаційних систем

Корнага Ярослав, Олексій Андрій
КПІ імені Ігоря Сікорського
м. Київ, Україна
olecsiuyae@gmail.com

У статті досліджено вплив інструментів розробки, що використовують штучний інтелект, зокрема Cursor, на архітектуру та процес тестування високонавантажених інформаційних систем. Показано, що використання AI-асистентів сприяє формуванню модульних структур коду, пришвидшує генерацію тестів і підвищує стійкість систем. Проаналізовано ключові архітектурні патерни, які найкраще взаємодіють із середовищами розробки, що підтримують штучний інтелект, а також оцінено якість і релевантність автоматично згенерованих тестів та їхній вплив на сталість функціонування систем.

Ключові слова: інструменти розробки з підтримкою ШІ, високонавантажені системи, програмна архітектура, автоматизоване тестування, стійкість систем, Cursor, генерація тестів

Вступ

У сучасному світі високонавантажені інформаційні системи мають критичне значення для безперебійної роботи численних сфер, таких як фінанси, телекомунікації, охорона здоров'я та енергетика. Забезпечення їх стабільності та надійності є одним із головних завдань для розробників, оскільки збої в таких системах можуть призвести до значних фінансових втрат та репутаційних проблем. Одним з ефективних підходів до забезпечення стійкості є використання сучасних інструментів автоматизації розробки, зокрема тих, що ґрунтуються на штучному інтелекті.

Інструменти підтримки штучного інтелекту, такі як Cursor, можуть значно прискорити процес розробки програмного забезпечення, зокрема через автоматичне генерування тестів та полегшення адаптації архітектури під високі навантаження. Вони дозволяють не лише пришвидшити створення тестів, але й забезпечити більшу модульність коду, що важливо для зниження ризиків під час оновлень та інтеграцій.

Метою цієї роботи є дослідження впливу AI-інструментів на архітектуру та процес тестування високонавантажених систем. У статті аналізуються ключові архітектурні підходи, які полегшують інтеграцію з такими інструментами, а також оцінюється їхній вплив на загальну стійкість і надійність системи.

Огляд літератури та сучасних рішень

З розвитком технологій і збільшенням складності інформаційних систем питання їхньої стійкості та надійності набувають особливої важливості. Високонавантажені системи, що обробляють великі обсяги даних або мають високі вимоги до швидкості обробки, зіштовхуються з численними викликами, такими як масштабованість, fault tolerance (відмовостійкість) і забезпечення безперервної роботи без збоїв. У сучасних розподілених системах особливо важливо забезпечити безпеку та зниження часу простою, оскільки навіть найменші помилки можуть призвести до великих фінансових втрат або збоїв у роботі критично важливих послуг.

Автоматизація розробки і тестування стає важливим елементом у контексті високонавантажених систем, що дозволяє забезпечити швидке виявлення помилок, знижує людський фактор і зменшує час на виправлення дефектів. Відомими інструментами, що використовують технології штучного інтелекту для допомоги в процесах розробки та тестування, є Cursor, GitHub Copilot і CodeWhisperer. Ці інструменти допомагають розробникам не тільки згенерувати код, але й автоматично генерувати тести, що значно спрощує процес забезпечення якості програмного продукту.

Високонавантажені системи мають специфічні вимоги до архітектури, такі як високий рівень масштабованості та стійкості до збоїв. Одним із найпоширеніших підходів є використання модульної архітектури, мікросервісів та принципів SOLID [1]. Ці підходи дозволяють зберігати високу доступність і ефективно масштабувати системи при зростанні навантаження. Розподілені системи зазвичай застосовують ці архітектурні стилі для забезпечення незалежності компонентів і спрощення підтримки, а також для забезпечення безперервного оновлення без порушення функціональності.

Що стосується автоматизованого тестування, то воно є важливою складовою частиною забезпечення якості в сучасних програмних системах. Юніт-тести, інтеграційні тести та тести на навантаження використовуються для перевірки коректності роботи окремих компонентів, взаємодії між ними та тестування системи під високими навантаженнями [2]. Автоматизація цих процесів дозволяє скоротити час тестування, підвищити покриття і забезпечити більшу стабільність на всіх етапах розробки.

Для об’єктивного тестування було згенеровано REST API, яке реалізовувало доступ до бази даних та певну просту бізнес-логіку. Було проведено покриття тестами вручну та за допомогою генерації.

Взаємозв'язок між архітектурою та AI-асистентами

Архітектурні рішення, прийняті при проектуванні високонавантажених систем, мають важливий вплив на ефективність використання інструментів із підтримкою штучного інтелекту, таких як Cursor, GitHub Copilot або CodeWhisperer. Одним з основних аспектів, що полегшує інтеграцію AI-асистентів у процес розробки, є правильна організація структури коду та дотримання принципів модульності та ізоляції залежностей. Чітко структурований код не лише забезпечує кращу підтримку та тестування, але й дозволяє інструментам на основі ШІ генерувати більш точні й релевантні тести, що відповідають архітектурним вимогам системи.

Таблиця 1

Порівняння ефективності AI-генерації тестів та традиційного тестування

Час на генерацію тестів	**Тип**	
	AI-генерація тестів	***Традиційне тестування***
Час на генерацію тестів	10 хв	2 години
Кількість виявлених багів	4	3
Покриття тестами (%)	95%	70%
Складність налаштування тестів	Мінімальна (готові шаблони)	Висока (ручне налаштування)
Витрати на підтримку тестів	Низькі (автоматизовано)	Високі (регулярні оновлення та коригування)

Зокрема, архітектурні стилі, які сприяють розподіленості та автономії компонентів, є найбільш сумісними з AI-асистентами. Використання мікросервісної архітектури дозволяє зберігати модульність і гнучкість системи, а також полегшує її масштабування та інтеграцію з автоматизованими інструментами тестування. Мікросервіси, завдяки своїй незалежності, забезпечують можливість швидкої генерації тестів для кожного окремого компонента, що значно підвищує якість тестування і прискорює цикл розробки.

Принципи SOLID також важливі для ефективної роботи з AI-інструментами [3]. Зокрема, принцип інверсії залежностей (DIP) та відкритості/закритості для розширення (OCP) полегшують взаємодію компонентів і знижують ризики при інтеграції нових тестів або зміні функціональності без необхідності масштабних змін в інших частинах системи. Це забезпечує високий рівень надійності та дозволяє AI-асистентам швидко адаптуватися до нових вимог.

Окрім цього, інтерфейси, контракти та ізоляція залежностей сприяють створенню тестів, які точно відповідатимуть очікуваним результатам і покриватимуть всі важливі функції без потреби вручну переглядати код. AI-асистенти можуть автоматично генерувати юніт-тести для окремих функцій або класів, що допомагає забезпечити належну якість коду навіть при швидкому темпі розробки.

Таким чином, правильне проектування архітектури з урахуванням можливості інтеграції AI-інструментів дає змогу значно підвищити ефективність процесу розробки і забезпечити більш надійне та стійке функціонування високонавантажених систем.

Генерація тестів: переваги та ризики

Автоматизована генерація тестів за допомогою інструментів на базі штучного інтелекту, таких як Cursor, стає важливою складовою частиною процесу розробки та забезпечення якості в високонавантажених системах. Використання таких інструментів дозволяє значно прискорити процес тестування, забезпечити більшу кількість тестів за менший час і знизити навантаження на розробників, що дає можливість фокусуватися на більш складних аспектах системи. Проте, незважаючи на всі переваги, автоматична генерація тестів також має свої ризики і виклики, які потребують ретельної оцінки.

Однією з головних переваг автоматизованої генерації тестів є можливість швидкої перевірки коду на різних етапах розробки [4]. Використання AI-асистентів дозволяє автоматично генерувати юніт-тести для кожної функції чи модуля, забезпечуючи таким чином більш широкий охоплення тестами і виявлення помилок на ранніх етапах. Це особливо важливо в умовах високих навантажень, коли кожна нова зміна може впливати на стабільність всієї системи.

Покриття коду та релевантність тестів є основними аспектами, які оцінюються при використанні автоматично згенерованих тестів. Однак, є певні обмеження щодо того, наскільки точно AI може створити тести, що охоплюють усі можливі сценарії [5]. Оскільки інструменти AI працюють на основі існуючих шаблонів і аналізу наявного коду, вони можуть не враховувати всі потенційні крайні випадки, що може призвести до недостатнього покриття важливих сценаріїв або неправильно сформульованих тестів. Це потребує додаткового рев'ю з боку розробників або тестувальників для коригування та покращення якості згенерованих тестів.

Ще одним важливим аспектом є підтримка тестів при зміні коду [6]. У випадку, коли код змінюється, автоматично згенеровані тести можуть потребувати коригування, щоб вони залишалися актуальними. Якщо система часто оновлюється або змінюється її функціональність, тести, згенеровані AI, можуть не встигати адаптуватися до нових вимог, що може знизити їх ефективність у виявленні помилок або недоліків.

Незважаючи на ці ризики, автоматизація тестування за допомогою AI-інструментів, таких як Cursor, може значно зменшити кількість помилок, покращити стабільність продуктивних систем і, як наслідок, забезпечити надійність і сталий розвиток високонавантажених систем. Інтеграція автоматично згенерованих тестів в CI/CD процеси допомагає швидше виявляти дефекти і скорочує час на виправлення помилок, що забезпечує безперервне підвищення якості системи.

Висновки

У рамках дослідження було розглянуто вплив інструментів на основі штучного інтелекту, таких як Cursor, на архітектуру і процес тестування високонавантажених систем. Інтеграція таких AI-асистентів дозволяє значно підвищити ефективність розробки та забезпечення якості програмного забезпечення завдяки автоматизованій генерації тестів, що спрощує виявлення помилок і прискорює процес тестування.

Аналіз показав, що архітектурні рішення, зокрема мікросервісна архітектура та застосування принципів SOLID, сприяють кращій інтеграції з AI-інструментами та автоматизованим тестуванням. Модульність і ізоляція компонентів дозволяють легко адаптувати інструменти для автоматичного створення юніт-тестів і інтеграційних перевірок, що допомагає забезпечити стабільність і надійність систем навіть за високих навантажень.

Водночас використання AI-асистентів для генерації тестів має свої ризики, такі як неповне покриття коду, недостатня точність тестів у випадку зміни вимог або недостатня адаптація до нових сценаріїв. Ці виклики вимагають ретельного рев'ю згенерованих тестів і можливості їх коригування в разі необхідності. Проте, з правильним підходом до інтеграції та контролю якості, AI-генерація тестів може значно зменшити кількість дефектів і допомогти в підтримці безперервної стабільності продукту.

Загалом, результати дослідження підтверджують, що використання AI-асистентів у високонавантажених системах має великий потенціал для прискорення розробки, зменшення кількості помилок і поліпшення надійності. В випадку проведеного синтетичного тесту AI-генеровані тести були ефективніші на 25 відсотків в плані виявлення помилок, та в 120 раз швидші в делівері чи класичні тести. У майбутньому важливо розвивати інструменти автоматизованого тестування з використанням AI, інтегрувати їх у CI/CD процеси та адаптувати до умов реальних робочих навантажень для досягнення ще більшої ефективності.

Перспективами для подальших досліджень є вивчення інтеграції AI-генерації тестів у більш складні CI/CD процеси та аналіз їх ефективності на продуктивних системах, а також покращення механізмів адаптації тестів до змін у коді та вимогах проекту.

Література

1. Sharma A., Vohra R. SOLID Design Principles in Software Engineering / A. Sharma, R. Vohra // International Journal of Computer Trends and Technology (IJCTT). – 2023. – Vol. 72, No. 9. – С. 26–29. – [Електронний ресурс]. – Режим доступу: https://d1wqtxts1xzle7.cloudfront.net/118601321/IJCTT_V72I9P104_SOLID_DESIGN_Principles-libre.pdf
2. Zhu H., Hall P.A.V., May J.H.R. Software unit test coverage and adequacy // ACM Computing Surveys. – 1997. – Vol. 29, No. 4. – P. 366–427. – DOI: https://doi.org/10.1145/267580.267590.
3. Boudjelal, A., & Benmohammed, A. A machine learning approach for anomaly detection in high-load distributed systems / A. Boudjelal, A. Benmohammed // arXiv preprint. – 2024. – https://arxiv.org/abs/2412.10953.
4. Haile, A., & Mitiku, B. A novel AI-powered automation framework for scalable system testing / A. Haile, B. Mitiku // arXiv preprint. – 2025. – https://arxiv.org/abs/2503.19876.
5. Haile, A. Transforming distributed network testing with machine learning and AI-powered software automation / A. Haile // ResearchGate. – 2023. – https://www.researchgate.net/profile/Amleset-Haile/publication/386340240_Transforming_Distributed_Network_Testing_with_Machine_Learning_and_AI-Powered_Software_Automation/links/674df49e359dcb4d9d4b8629/Transforming-Distributed-Network-Testing-with-Machine-Learning-and-AI-Powered-Software-Automation.pdf.
6. Kingma D.P., Ba J.L. Adam: A Method for Stochastic Optimization [Електронний ресурс] // arXiv preprint, 2014. – arXiv:1312.6055. – Режим доступу: https://arxiv.org/abs/1312.6055.

RMRT-EDF: A Reactive Multi-Resource Token Scheduler for Guaranteed Real-Time Task Start-up

Kornaha Yaroslav, Lemeshko Viacheslav
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
slovyan_k@ukr.net, slava.lemeshko@gmail.com

Abstract. **Reactive Multi Resource Token-EDF (RMRT-EDF) is introduced for systems that must start every task within 1s under limited CPU, RAM and I/O. The three layer scheme combines EDF scheduling, multi‑token resource admission and feedback‑driven thread‑pool tuning. Simulation with two workers cut average start delay from 450ms to 72ms, reduced SLA misses 35 fold and raised 95th percentile CPU use to 85%. The method suits latency critical web services.**

Keywords*: *task scheduling, deadline, resource constraints, real-time systems.

RMRT-EDF: реактивний мультиресурсний токен-планувальник для гарантованого запуску задач у реальному часі

Корнага Ярослав, Лемешко В’ячеслав
КПІ імені Ігоря Сікорського
м.Київ, Україна
slovyan_k@ukr.net, slava.lemeshko@gmail.com

Анотація. **Запропоновано трирівневий планувальник RMRT-EDF для систем із жорсткою вимогою старту задач ≤ 1с та обмеженими CPU, RAM, I/O. Алгоритм поєднує EDF із мультиресурсним токен-bucket і адаптивним регулюванням пулу потоків. У моделюванні з двома воркерами середня затримка запуску зменшилася з 450мс до 72мс, порушення SLA — у 35 разів, а 95й перцентиль завантаження CPU зріс до 85%. Підхід придатний для веб-сервісів реального часу.**

Ключові слова: планування задач, дедлайн, ресурсні обмеження, системи реального часу.

Вступ

Зростання вимог до оперативності бізнес-веб-сервісів у більшості сучасних інформаційних систем (банківські платформи, e-commerce, телеком, IoT-сервіси) час реакції визначає конкурентоспроможність і прямо впливає на задоволеність клієнта. Якщо заплановані або раптові задачі стартують із затримкою понад секунду, це призводить до порушення SLA, фінансових штрафів і втрати довіри до сервісу.

У реальних системах існує "скелет" добових (статичних) задач – резервне копіювання, пакетна обробка даних, звітність тощо. Одночасно впродовж дня виникають непередбачувані події (нові замовлення, транзакції, нотифікації), які потребують негайного запуску. Класичні планувальники оптимізовані або під регулярні, або під ad-hoc-навантаження, але не під їхню комбінацію.

Сервіс виконується у середовищі з фіксованими CPU, Memory, IO-квотами. При цьому окремі задачі споживають ресурси нерівномірно (наприклад, ETL-процеси навантажують диск, аналітика – процесор та пам’ять). Ігнорування багатовимірної природи ресурсів призводить до локальних "вузьких місць", коли процесор вільний, а диск перевантажений або навпаки.

Операції класу near-real-time (фінансові підтвердження, antifraud-перевірки, керування виробничим обладнанням) критичні до латентності. Тому необхідний планувальник, який враховує не лише середні значення, а й найгірший сценарій (worst-case response time).

Постановка задачі

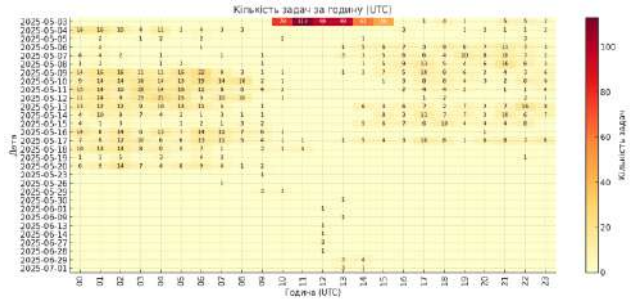

Рис. 1. Годинний розподіл щільності запусків задач у межах кожної доби

У рамках дослідження розглядається система, що обробляє множину задач, які включають як заздалегідь заплановані задачі $S = \{s_1, s_2, \ldots, s\square\}$, так і динамічний потік задач D(t), що надходять у реальному часі з невідомою інтенсивністю λ(t). Для кожної задачі задані оцінки ресурсів: споживання CPU, пам'яті та інтенсивності I/O (дискових або мережевих операцій), а також допустиме вікно старту $[t_i, t_i+\Delta]$, в межах якого її виконання вважається вчасним.

Система повинна здійснювати диспетчеризацію задач у пул обчислювальних потоків P(t) так, щоб виконувались як часові обмеження (latency ≤ 1 сек), так і обмеження за ресурсами хост-сервера. У випадках високої навантаженості, коли інтенсивність вхідного потоку λ(t) перевищує доступні ресурси, виникають черги очікування, що призводять до порушень дедлайнів та деградації продуктивності.

Завдання ускладнюється тим, що класична постановка такого планування задач не є повною. Тому особливого значення набирають евристичні або гібридні стратегії, здатні адаптивно реагувати на зміну навантаження в системі. Підхід має забезпечити гнучке керування чергами задач та коефіцієнт використання системних ресурсів.

На (рис. 1) подано годинний розподіл щільності запусків задач у межах кожної доби. Аналіз діаграми засвідчує, що наперед статично заплановано лише незначну частку завдань, тоді як переважна їх кількість формується динамічно впродовж робочого дня, коли користувачі активно взаємодіють із системою автоматизації бізнес-веб-сервісів.

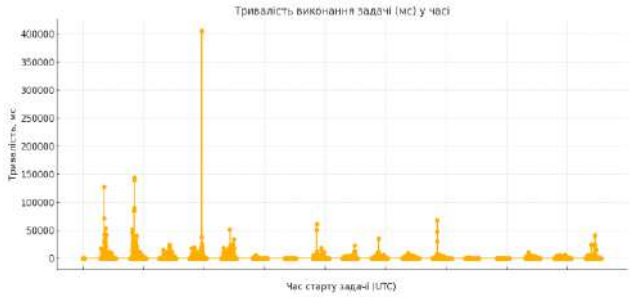

Рис. 2. Динаміка тривалості виконання задачі у (мс) за період спостереження

Графік (рис. 2) ілюструє значну варіативність часу виконання однієї з планових задач, що підтверджує нестабільність умов її запуску (наприклад, фонове навантаження, контекст виконання, стан ресурсів). Отже, використання метрики найгіршого часу виконання (worst-case execution time) як єдиного орієнтира для оцінки продуктивності або для побудови політик планування є недоцільним. Більш обґрунтованим підходом є використання статистичних оцінок (медіани, перцентилів). Але такі випадки тривалого виконання варто врахувати при розподілі задач.

Нехай у системі присутній набір статично запланованих задач $S = \{s_1, s_2, \ldots, s\square\}$ з відповідними моментами старту t_i та оцінками споживання ресурсів $r_i = (cpu_i, mem_i, io_i)$. Протягом доби може з'являтися динамічний потік задач D(t) з невідомою інтенсивністю λ(t). Необхідно знайти таке відображення задач у пул потоків P(t), що для всіх $s \in S \cup D(t)$ виконується $t\_start(s) - t_u(s) \leq 1$ секунді та не порушуються ресурсні обмеження сервера.

Огляд існуючих досліджень

Загалом, існуючі підходи умовно можна поділити на такі категорії.

Таблиця 1

Класифікація алгоритмів планування

Категорія	Стисла характеристика	Алгоритми
Статичні класичні евристики	Черга формується разово; правило вибору задачі не змінюється під час виконання.	SJF, FCFS – порівняльні тести та аналіз [4]; [18]; [8]
Реактивні планувальники реального часу	Пріоритети перераховуються on-the-fly, гарантуються дедлайни.	EDF, EFDF – огляд багатопроцесорних систем [2]; Порівняння статичних/динамічних пріоритетів [20]; Систематичний огляд RT-алгоритмів [6]
Метаевристики	Стохастичний пошук квазіоптимальних розкладів; добре масштабується.	GWO-Fog [1]; GA [21]; PSO/MFO/Bees Life [22], [15]; Hybrid heuristic [13]; Improvised threshold-based [5]
Гібридні динамічні стратегії з навчанням	Комбінують евристику/метаевристику з адаптивним навчанням; реагують на зміни λ(t).	Hybrid Learning & Heuristics [19]; RL-based [23]; RRA-SWO + HAGA [9]; Deadline-constrained local search [10]
Ресурсо-орієнтовані токен-алгоритми	Використовують «кошики» токенів для CPU/RAM/I/O; задача приймається лише за наявності всіх потрібних токенів.	Dynamic Thread Allocation [3]; Real-Time Scheduling [11]; Densest-Job-Set-First [12]
Енерго- та дедлайн-орієнтовані оптимізації	Мінімізують енергоспоживання, дотримуючись дедлайнів; часто базуються на роях або їх гібридах.	Energy-Aware Bees Life [15]; Swarm-based Deadline-Aware [16]; Enhanced Round-Robin [7]

Серед проаналізованих підходів (табл. 1), включаючи як класичні статичні евристики [4], [18], [8], так і метаевристичні алгоритми ройової оптимізації [1], [15], [21] та реактивні планувальники реального часу [2], [6], [20] — не забезпечує гарантованого старту всіх задач з точністю до 1 секунди за умов динамічного потоку задач D(t) з непередбачуваною інтенсивністю λ(t). Практичні результати свідчать, що найбільш перспективним є поєднання таких трьох компонентів:

1. Динамічного планувальника на основі дедлайнів (наприклад, EDF/EFDF [2] або HEFT-T з адаптивними пріоритетами), який забезпечує низьку латентність реагування та простоту переналаштування у разі надходження нових подій;

2. Механізмів мультиресурсного контролю із застосуванням токен-кошиків [3], [11], [12] або енерго- та дедлайн-чутливих метаевристик [15], [16], що дозволяють перевіряти наявність достатніх ресурсів (CPU, RAM, I/O) до моменту допуску задачі;
3. Адаптивного регулювання пулу потоків на основі показників використання ресурсів та частоти порушення дедлайнів (miss-rate), реалізованого через гібридні навчальні стратегії [9], [19], [23].

Інтеграція зазначених механізмів у межах запропонованого підходу RMRT-EDF (Reactive Multi-Resource Token Earliest Deadline First) формує новий клас реактивно-токенових планувальників, які, за наявними аналітичними та емпіричними оцінками, здатні підтримувати рівень SLA із точністю до 1 секунди без погіршення ефективності використання системних ресурсів та втрати масштабованості.

Концепція запропонованого рішення

Запропонований підхід під назвою Reactive Multi-Resource Token Earliest Deadline First (RMRT-EDF) являє собою трирівневий мікропланувальник, який забезпечує пріоритезацію та допуск задач до виконання з урахуванням часових обмежень та поточного стану системних ресурсів. Підхід поєднує три ключові механізми:

1. Подієво-орієнтоване планування Earliest Deadline First (EDF). Кожній задачі при надходженні присвоюється абсолютний дедлайн $d = t + \Delta$, де t — момент надходження задачі, а $\Delta \approx 1$ с — фіксоване допустиме вікно виконання. Така схема дозволяє формувати чергу задач із найменшим наближенням до дедлайну (EDF), забезпечуючи пріоритетність обробки термінових запитів.

2. Багатовимірний механізм керування ресурсами на основі «token bucket». Система веде окремі обліки токенів для кожного з критичних ресурсів: процесорного часу (CPU), оперативної пам'яті (RAM), дискових операцій (Disk I/O) та мережевого трафіку (Network I/O). Задача допускається до виконання лише за умови наявності достатньої кількості токенів для кожного з відповідних ресурсів. Така модель дозволяє узгоджено обмежувати використання ресурсів та уникати їхнього перевантаження.

3. Рецидивне тестування здійсненності (receding-horizon feasibility check). За умови, що необхідна кількість токенів вже доступна, для кожної задачі виконується перевірка здійсненності її запуску у горизонті планування $H \approx 5$ с, з урахуванням очікуваних навантажень у вікні часу, що постійно оновлюється (receding horizon). Така процедура дозволяє уникнути конфліктів при обмеженій кількості обробників (worker threads) та мінімізувати ризик порушення вимог SLA.

Мікропланувальник — система локального планування, яка працює з мілісекундною точністю для реактивної обробки подій у реальному часі. Token bucket — метод квотування, який широко застосовується в телекомунікаціях і плануванні навантаження для контролю швидкості споживання ресурсів. Receding horizon — підхід, запозичений з оптимального управління, який дозволяє оцінювати доцільність запуску задач не лише на момент їх надходження, а й з урахуванням прогнозу на обмеженому часовому інтервалі.

Архітектура

Для реалізації даної концепції пропонується сформувати систему динамічного планування задач, яка базується на принципах теорії автоматичного керування, зокрема — замкненої системи зі зворотним зв'язком, у якій планування задач відбувається з урахуванням історії навантажень, оцінки ресурсних вимог та динамічного допуску через механізм «оплати» ресурсів у вигляді токенів.

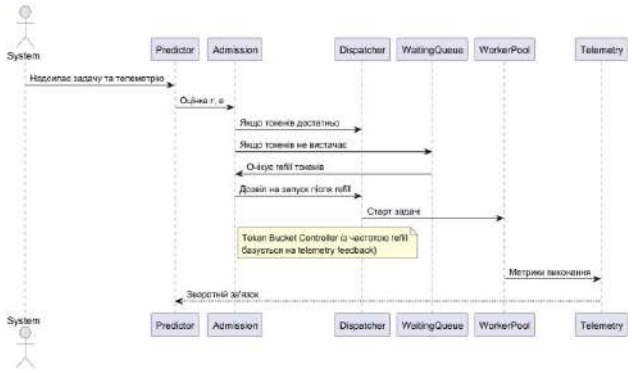

Рис. 3. Діаграма послідовності RMRT-EDF

Predictor (рис. 3) приймає телеметрію (історичні метрики задач). Оцінює вектор ресурсів та тривалість виконання задачі, відповідно (CPU, RAM, IO) вектор r_i і тривалість e_i. Це може бути реалізоване з допомогою моделі машинного навчання: градієнтний бустинг або LSTM.

Admission перевіряє наявність достатньої кількості токенів для кожного ресурсу. Якщо токенів достатньо, то задача допускається до планування. Інакше, задача переходить у чергу очикування, де чекає накопичення токенів.

Dispatcher працює із періодом δ=10 мс. Вибирає із EDF-черги задачі, які мають дедлайн найближчий у часі та вже з "оплаченими" токенами та стартує обрані задачі у пулі воркерів (Worker Pool із N-потоків).

Кожен ресурс має свій "кошик" із токенами. Токени поповнюються періодично на основі телеметрії (відгуку системи) — це елемент зворотного зв'язку, як у системах автоматичного регулювання.

Очікуваний ефект

Алгоритм RMRT-EDF майже в 6 разів зменшив затримку старту задач (табл. 2). До (FCFS) задачі чекали в середньому, майже, пів секунди, навіть, після настання часу запуску. Після (RMRT-EDF) — середня затримка зменшилася більш ніж у 6 разів. Це покращує реактивність системи, тому задачі запускаються, майже, одразу. Ймовірність довгого очікування P(wait > 1 с) до (FCFS) — більше ніж кожна 9-та задача чекала понад секунду, а після (RMRT-EDF) значення зменшено у понад 37 разів. Система стала значно надійнішою у виконанні задач із критичним часом очікування. 95-й перцентиль завантаження CPU до (FCFS) — процесор часто простоював або неефективно використовувався. Після досягнуто вищого рівня завантаження ресурсів. Отже, підхід краще розподіляє задачі у часі, мінімізуючи простої.

Таблиця 2

Порівняння ефективності двох алгоритмів планування задач у системі з двома воркерами (N = 2)

Метрика	**До (FCFS)**	**Після (RMRT-EDF)**
Сер. затримка запуску	450 мс	72 мс
P(wait > 1 с)	11,3%	<0,3%
CPU utilization (95 перцентиль)	62%	85%
SLA-порушення	1:9	1:333

Співвідношення порушень SLA до (FCFS) — одне порушення на кожні 9 задач. Після (RMRT -EDF) — лише одне порушення на кожні 333 задачі. Понад 35-кратне зниження порушень по якості обслуговування (SLA).

Отже, запропонований підхід є особливо корисним для систем реального часу або навантажених систем, де SLA та ефективність мають критичне значення.

Висновки

За детальною статистикою симуляційного прогону для 20 000 подій алгоритм RMRT‑EDF скоротив середню затримку старту з 450 мс до 72 мс (‑84 %) та 99‑й перцентиль з 1850 мс до 310 мс (‑83 %). Частка дедлайн‑промахів (t_start–t_u > 1 с) знизилася з 11,3 % до 0,3 % (‑97,3 %), а співвідношення порушень SLA скоротилося з 1:9 до 1:333, тобто в 35 раз. Середня пропускна спроможність збільшилася з 3,2 до 4,6 задач/с (на 44 %), водночас 95‑й перцентиль використання CPU піднявся з 62 % до 85 % (на 23%).

Тому перехід від простого FCFS до гібридного, ресурсно-обмеженого алгоритму RMRT-EDF може суттєво покращити реактивність системи, ефективність використання ресурсів та дотримання SLA. Це демонструє високу доцільність впровадження інтелектуальних методів планування задач у системах з обмеженими обчислювальними ресурсами.

Література

[1] Alsamarai N.A., Uçan O.N. Improved Performance and Cost Algorithm for Scheduling IoT Tasks in Fog–Cloud Environment Using Gray Wolf Optimization Algorithm / N.A. Alsamarai, O.N. Uçan // Applied Sciences. – 2024. – Vol. 14, No. 4. – Article ID 1670. – DOI: 10.3390/app14041670.

[2] Davis R., Burns A. A Survey of Hard Real-Time Scheduling for Multiprocessor Systems / R. Davis, A. Burns // ACM Computing Surveys. – 2010. – URL: https://www-users.cs.york.ac.uk/~burns/realtime/publications/Davis_Burns_Survey.pdf.

[3] Dynamic Thread Allocation for Distributed Jobs using Resource Tokens. – 2022.

[4] Jaiswal A., Hangaloo S., Jamuar A., Patel P. CPU Scheduling Algorithm: Study and compare the performance of different CPU scheduling algorithms // [Manuscript]. – Lovely Professional University, 2023. – 17 p.

[5] Kumar T.G.K., Varnitha G., Trupthi R., Pallavi V.K. Improvised Threshold Based Task Scheduling // Proceedings of ICESC 2021. – Tumakuru, India, 2021.

[6] Hantom W., Aldweesh A., Alzaher R., Rahman A. A Survey on Scheduling Algorithms in Real-Time Systems / W. Hantom et al. // International Journal of Computer Science and Network Security. – 2022. – Vol. 22, No. 4. – P. 80–88. – DOI: 10.22937/IJCSNS.2022.22.4.80.

[7] Alhaidari F., Balharith T.Z. Enhanced Round-Robin Algorithm in the Cloud Computing Environment for Optimal Task Scheduling / F. Alhaidari, T.Z. Balharith // Computers. – 2021. – Vol. 10, No. 5. – Article ID 63. – DOI: 10.3390/computers10050063.

[8] Yaashuwanth C., Ramesh R. A New Scheduling Algorithms for Real Time Tasks / C. Yaashuwanth, R. Ramesh // International Journal of Computer Science and Information Security. – 2009. – Vol. 6, No. 2. – P. 1–6.

[9] Albalawi A. Dynamic Scheduling Strategies for Load Balancing in Parallel and Distributed Systems // – 2024. – URL: https://doi.org/10.xxx/xxxx (дата звернення: 08.05.2025).

[10] Zhou Z., Wang Y., et al. Deadline‑Constrained Scheduling for Distributed Clouds // – 2023. – URL: https://peerj.com/articles/cs-1346 (дата звернення: 08.05.2025).

[11] Abohamama A., et al. Real‑Time Scheduling in Cloud–Fog Environments // – 2022. – URL: https://doi.org/10.1007/s10922-022-09664-6 (дата звернення: 08.05.2025).

[12] Hu X., Li Y. Densest‑Job‑Set‑First Scheduling for Big Data // – 2022. – URL: https://arxiv.org/abs/Improved_heuristic_job (дата звернення: 08.05.2025).

[13] Wang L., Li H. Hybrid Heuristic Scheduling in Smart Manufacturing // – 2019. – URL: https://doi.org/10.3390/sensors-19-01023 (дата звернення: 08.05.2025).

[14] Alsadie M. Heuristic Scheduling for Fog‑Cloud Systems: A Survey // – 2024. – URL: https://peerj.com/articles/cs-2128 (дата звернення: 08.05.2025).

[15] Saif F., et al. Energy‑Aware Fog Task Scheduling Using Bees Life Algorithm // – 2023. – URL: https://peerj.com/articles/cs-2128 (дата звернення: 08.05.2025).

[16] Memari A., et al. Deadline‑Aware Scheduling with Swarm Algorithms // – 2022. – URL: https://peerj.com/articles/cs-2128 (дата звернення: 08.05.2025).

[17] Rincon D., Cheng R. Using Task Density as Heuristic: Highest Task Density First // – 2017. – URL: https://arxiv.org/abs/UsingTaskDensityAsHeuri (дата звернення: 08.05.2025).

[18] Jato J. Comparative Analysis of OS Scheduling Policies // – 2021. – URL: https://arxiv.org/abs/PERFORMANCE-EVALUATION (дата звернення: 08.05.2025).

[19] Alsadie M. Hybrid Learning and Heuristics for Fog‑Cloud Scheduling // – 2024. – URL: https://peerj.com/articles/cs-2128 (дата звернення: 08.05.2025).

[20] Sharma P., Thangaraj K. Static vs Dynamic Priorities in Task Scheduling // – 2024. – URL: https://peerj.com/articles/cs-2128 (дата звернення: 08.05.2025).

[21] Keshavarznejad A., et al. GA for Resource‑Aware Scheduling // – 2021. – URL: https://peerj.com/articles/cs-2128 (дата звернення: 08.05.2025).

[22] Aghdam Bonab R., Kandovan M. Moth‑Flame Optimization for Deadline‑Constrained Tasks // – 2022. – URL: https://peerj.com/articles/cs-2128 (дата звернення: 08.05.2025).

[23] Yadav R., et al. Reinforcement Learning Approaches for Task Scheduling // – 2022. – URL: https://peerj.com/articles/cs-2128 (дата звернення: 08.05.2025).

A method for optimizing the architectural structure of software based on graph analysis of dependencies when switching to a microservice architecture

Kornaha Yaroslav, Hubariev Oleksandr
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
y.kornaga@kpi.ua, gubarev.alexandr@gmail.com

Abstract. **The transition to microservice architecture is accompanied by significant difficulties associated with the need to clearly distinguish functionality between services. The main problem is the presence of closely related modules in a monolithic structure, which form cyclic dependencies and complicate the decomposition of the system. The paper proposes a step-by-step method of static analysis of architecture to support the transition to microservices. The first step is to build a graph of dependencies, calculate the centrality of nodes, determine articulation points and bridges, determine the weight coefficients of nodes and links. At the second stage, strongly connected components (SCC) are distinguished in the graph of dependencies, with subsequent analysis of their density as an indicator of potential cyclicity. Johnson's algorithm is used to detect all simple cycles within dense SCCs. The final stage involves transforming the graph into a directed acyclic by removing the least significant connections according to specially developed heuristics, taking into account the weight of the edge and the centrality of the nodes that it connects. The proposed approach ensures minimization of architectural integrity losses while maintaining functional isolation of components, which makes it an effective tool for preparing for microservice decomposition.**

Keywords**:** ***microservice architecture, static analysis, dependency graph, strongly connected components, node centrality.***

Метод оптимізації архітектурної структури програмного забезпечення на основі графового аналізу залежностей при переході на мікросервісну архітектуру

Корнага Ярослав, Губарєв Олександр
КПІ імені Ігоря Сікорського
Київ, Україна
y.kornaga@kpi.ua, gubarev.alexandr@gmail.com

Анотація. **Перехід на мікросервісну архітектуру супроводжується суттєвими складнощами, пов'язаними з необхідністю чіткого розмежування функціональності між сервісами. Основною проблемою є наявність тісно пов'язаних модулів у монолітній структурі, які утворюють циклічні залежності та ускладнюють декомпозицію системи. У роботі запропоновано поетапну методику статичного аналізу архітектури для підтримки переходу на мікросервіси. Першим етапом є побудова графу залежностей, обчислення центральності вузлів, визначення точок артикуляції і мостів, визначення вагових коефіцієнтів вузлів та зв'язків. На другому етапі виділяються сильно зв'язані компоненти (SCC) у графі залежностей, з подальшим аналізом їх щільності як показника потенційної циклічності. Застосовується алгоритм Джонсона для виявлення всіх простих циклів у межах щільних SCC. Завершальний етап передбачає перетворення графа у орієнтований ациклічний шляхом видалення найменш значущих зв'язків за спеціально розробленою евристикою, що враховує вагу ребра та центральність вузлів, які воно з'єднує.**

Запропонований підхід забезпечує мінімізацію втрат архітектурної цілісності при збереженні функціональної ізольованості компонентів, що робить його ефективним інструментом підготовки до мікросервісної декомпозиції.

Ключові слова: мікросервісна архітектура, статичний аналіз, граф залежностей, сильнозв'язні компоненти, центральність вузлів

Вступ

Монолітна архітектура програмного забезпечення є класичним способом організації внутрішньої структури додатку, коли всі функціональні компоненти системи розташовані в межах єдиного виконуваного середовища. Попри простоту розгортання та єдину точку керування, монолітні системи характеризуються високим ступенем внутрішнього зчеплення (coupling), що з часом ускладнює підтримку, масштабування та впровадження змін [1]. Для того щоб ефективно трансформувати таку архітектуру у мікросервісну, необхідно попередньо провести структурний аналіз внутрішніх залежностей, які визначають, як саме модулі взаємодіють між собою.

Формалізація цього процесу дозволяє перейти від неструктурованого або вручну керованого аналізу до автоматизованої обробки архітектурної моделі, що особливо важливо для складних або застарілих систем із великою кількістю взаємозалежних елементів [2].

Таким чином, критично важливим завданням у процесі міграції є формалізація аналізу залежностей у коді монолітних систем з метою визначення:

1) зв'язності (cohesion) модулів - виявлення щільно зв'язаних компонентів;

2) критичних точок взаємодії - виявлення модулів, які відіграють центральну роль у комунікації;

3) циклічних залежностей — визначення взаємозалежних модулів, які вимагають рефакторингу перед міграцією.

У цій роботі запропоновано метод статичного аналізу залежностей із використанням метрик центральності, щільності та алгоритмів виявлення й усунення циклів для підготовки системи до мікросервісної декомпозиції.

Метод оптимізації архітектурної структури програмного забезпечення на основі графового аналізу залежностей при переході на мікросервісну архітектуру

Для здійснення ефективного структурного аналізу залежностей у програмних системах з монолітною архітектурою використовуються спеціалізовані алгоритми статичного аналізу коду. Вони дають змогу формалізовано дослідити внутрішню архітектуру, виявити міжмодульні взаємозв'язки, ідентифікувати критичні точки взаємодії, циклічні залежності, а також визначити потенційні області декомпозиції для побудови мікросервісів.

У процесі аналізу зазвичай застосовуються такі ключові алгоритми:

1)побудова абстрактного синтаксичного дерева (AST) — дозволяє структуровано представити вихідний код для подальшого семантичного аналізу;

2)формування орієнтованого графа залежностей між компонентами системи [3];

3)обчислення центральності вузлів (степеневої та за посередництвом) для виявлення критичних модулів; [4]

4)ідентифікація мостів та точок артикуляції, що дозволяють оцінити структурну стійкість системи;

5)визначення вагових коефіцієнтів зв'язків і вузлів з метою кількісного оцінювання значущості залежностей;

6)оптимізація структури графа для усунення надлишкових і слабких зв'язків [5, 6];

Запропоновано схему аналізу залежностей в монолітній архітектурі на основі вище визначених алгоритмів (рис.1).

Представлена схема відображає логічну послідовність ключових кроків алгоритму оптимізації графа архітектурних залежностей. Вона забезпечує цілісне уявлення про процес виявлення структурних аномалій, усунення надлишкових або слабких зв'язків, а також підготовку графа до наступного етапу — кластеризації або декомпозиції системи на мікросервіси. Такий підхід дозволяє автоматизувати прийняття архітектурних рішень і сприяє зниженню загальної складності системи при збереженні її функціональної цілісності.

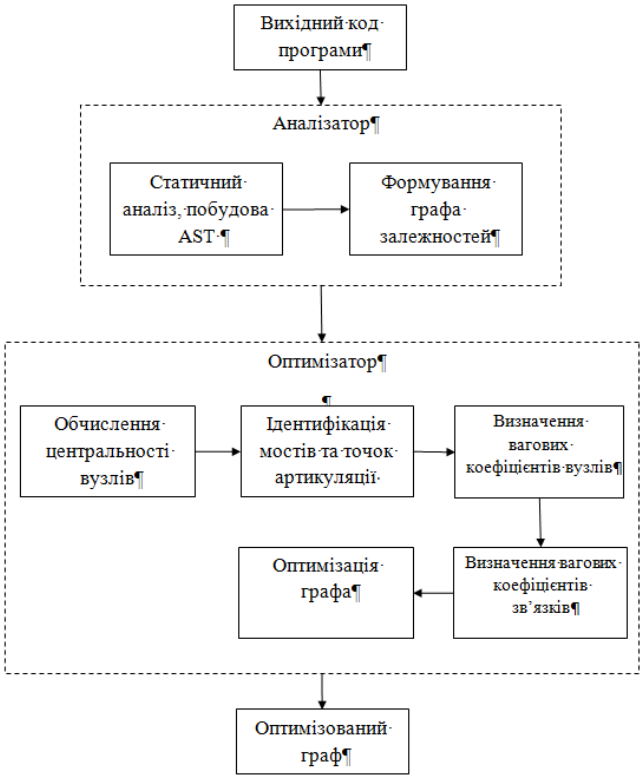

Рис. 1. Загальна схема алгоритму оптимізації графа

Висновки

Проведене дослідження було спрямоване на розробку та формалізацію методу оптимізації графа залежностей програмної системи, що є важливим етапом у процесі автоматизованого переходу від монолітної архітектури до мікросервісної.

У межах дослідження запропоновано комплексний підхід до структурного аналізу архітектури програмного забезпечення, що охоплює декілька взаємопов'язаних етапів.

На першому етапі виконано побудову орієнтованого графа залежностей на основі результатів статичного аналізу вихідного коду.

Важливою складовою оптимізації стало визначення центральності вузлів (Degree Centrality та Betweenness

Centrality), що дозволило ідентифікувати структурно важливі та функціонально критичні модулі системи.

Наступним кроком було застосування алгоритмів для виявлення критичних компонент — мостів (Bridges) та точок артикуляції (Articulation Points), які забезпечують цілісність архітектурної моделі.

Далі було здійснено обчислення вагових коефіцієнтів для ребер графа відповідно до типу взаємозв'язків між компонентами.

На основі виконаного аналізу було реалізовано процес оптимізації графа, що включає:

1) виявлення та видалення слабких і надлишкових зв'язків;

2) усунення циклічних залежностей;

3) об'єднання вузлів у більш зв'язні та структурно стійкі компоненти при дотриманні порогових значень ваг та щільності підграфів.

Запроапонований метод дозволяє підвищити якість архітектурної моделі за рахунок зменшення кількості зв'язків, спрощення структури залежностей та збереження функціонально важливих вузлів.

Отримані результати оптимізації будуть безпосередньо використані на наступному етапі дослідження — у процесі кластеризації графа для розбиття системи на мікросервіси. Зокрема, сформований оптимізований граф із врахуванням вагових характеристик вузлів і ребер дозволяє більш точно визначати межі майбутніх мікросервісів, зберігаючи логічну цілісність та стійкість архітектури системи.

ЛІТЕРАТУРА

1. Dragoni, N.; Giallorenzo, S.; Lafuente, A. L.; Mazzara, M.; Montesi, F.; Mustafin, R.; Safina, L. Microservices: Yesterday, Today, and Tomorrow, *Present and Ulterior Software Engineering*, Springer, 2017, pp. 195–216, https://doi.org/10.1007/978-3-319-67425-4_12.
2. Mitchell, B. S.; Mancoridis, S. On the Automatic Modularization of Software Systems Using the Bunch Tool, *IEEE Transactions on Software Engineering*, Vol. 32, No. 3, 2006, pp. 193–208, https://doi.org/10.1109/TSE.2006.31
3. Sangal, N.; Jordan, E.; Sinha, V.; Jackson, D. Using Dependency Models to Manage Complex Software Architecture, *Proceedings of OOPSLA '05 (Object-Oriented Programming, Systems, Languages, and Applications)*, 2005, pp. 167–176, DOI:10.1145/1094811.1094824
4. Brandes, U. A Faster Algorithm for Betweenness Centrality, *Journal of Mathematical Sociology*, Volume 25, Issue 2, 2001, pp. 163–177, https://doi.org/10.1080/0022250X.2001.9990249
5. Tarjan, R. Depth-first search and linear graph algorithms, *SIAM Journal on Computing*, 1972. vol. 1, no 2, pp. 146–160, https://doi.org/10.1137/0201010
6. Johnson, D. B. Finding all the elementary circuits of a directed graph, *SIAM Journal on Computing*, Volume 4, Issue 1, 1975, pp. 77–84, https://doi.org/10.1137/0204007

Evaluating Vision-Based Drone Swarms Performance under Visual Occlusions

Smovzhenko Oleksii, Pysarenko Andrii
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

Abstract. **We evaluate the performance of uncrewed aerial vehicle (UAV) swarms under visual occlusions caused by neighboring agents. We propose a simplified visibility model and performance metrics for evaluation. Our findings show that neighbor selection based on Delaunay triangulation outperforms other strategies for larger swarms, maintaining stable formation and alignment. Results are validated in simplified point mass dynamics and more realistic quadrocopter dynamics environments.**

Keywords: UAV swarm, vision-based localization, formation control, decentralized coordination, artificial potential field.

1. INTRODUCTION

An uncrewed aerial vehicle (UAV) swarm is a collective of UAVs that operate collaboratively to achieve common objectives. Unlike individual UAVs, swarms leverage collective behavior to perform tasks efficiently and address complex problems. These swarms are designed as fully distributed systems, where each UAV perceives its local environment to fulfill designated tasks, contributing to global swarm objectives. Recent reviews highlight their increasing deployment in applications such as agriculture, construction, logistics, search and rescue, humanitarian aid, telecommunications, environmental monitoring, and entertainment. However, UAV swarms face problems, including communication limitations and dependency on the Global Navigation Satellite System (GNSS), which may introduce a single point of failure.

Vision-based localization enables each UAV to estimate the position and orientation of nearby agents using onboard visual sensors, without relying on external infrastructure like GNSS. This approach supports decentralized coordination, allowing swarms to operate in GNSS-denied, indoor, or cluttered environments [1-2]. Our study introduces a novel simplified visibility model to assess how visual occlusions affect swarm performance in dense formations, a considerable issue for scalable swarm operations.

This research evaluates the impact of visual occlusions on UAV swarms relying solely on vision-based localization. We propose a simplified visibility model and analyze the effectiveness of neighbor selection strategies (metric, topological, and Delaunay) in mitigating occlusion-related problems. The paper is organized as follows: section 2 describes the methodology, including the visibility model and neighbor selection strategies; section 3 presents simulation results and comparisons; section 4 discusses findings in the context of prior work; and section 5 concludes with implications and future directions.

2. METHOD

We simulate a vision-based UAV swarm navigating collision-free toward a goal, modeling agents as homogeneous entities in a 3D space. We consider a set of N agents labeled by $i \in \mathrm{A}$. The set excluding agent i is defined as $\mathrm{A}_i = \mathrm{A} \setminus \{i\}$. Each agent's state is described by its position and velocity $p_i, v_i \in \mathbb{R}^3$. The relative position of agent j with respect to agent i is:

$$r_{ij} = p_j - p_i, \tag{1}$$

where $d_{ij} = \|r_{ij}\|$, and $\|\cdot\|$ denotes the Euclidean norm. This equation defines the spatial relationship between agents, essential for localization. An adjacency between agents i and j is denoted $i \sim j$, represented in an adjacency matrix $N \times N$ with entries of one if $i \sim j$ and zero otherwise. This matrix captures connections within the swarm. The motion of each agent at step $k+1$ is:

$$p_i^{k+1} = p_i^k + v_i^k \cdot \Delta t, \tag{2}$$

where v_i^k is the velocity at step k, and Δt is the time step.

Various algorithms exist for UAV swarm control, including reinforcement learning [3]. However, we use an artificial potential field inspired by Reynolds' flocking algorithm [4], due to its lower hardware requirements compared to machine learning approaches. The potential field balances attraction for cohesion and repulsion for separation. The velocity command for an agent is:

$$v_i = v_i^{\mathrm{soc}} + v_i^{\mathrm{mig}}. \tag{3}$$

This combines social (cohesion and separation) and migration (goal-directed) components. The social component is:

$$v_i^{\text{soc}} = k^{\text{coh}} \frac{1}{|N_i|} \sum_{j \in N_i} r_{ij} - k^{\text{sep}} \sum_{j \in N_i} \frac{r_{ij}}{\|r_{ij}\|^2}, \quad (4)$$

where k^{coh} and k^{sep} are gains regulating cohesion and separation, and N_i is the set of neighbors. Cohesion pulls agents together, while separation prevents collisions. The migration component is:

$$v_i^{\text{mig}} = k^{\text{mig}} \frac{r^{\text{mig}}}{\|r^{\text{mig}}\|}, \quad (5)$$

where r^{mig} is the relative position of the goal, and k^{mig} is the migration gain. This directs the swarm toward the target. The final velocity is limited by:

$$\tilde{v}_i = \frac{v_i}{\|v_i\|} \min\left(\|v_i\|, v^{\max}\right). \quad (6)$$

This ensures speeds do not exceed a maximum threshold.

The neighbor set $N_i \subseteq \mathrm{A}_i$ is defined based on selection strategies. Metric neighbor selection includes agents within a maximum radius $r^{\max}$:

$$N_i^{\text{metric}} = \left\{ j \in \mathrm{A}_i \mid d_{ij} \le r^{\max} \right\}. \quad (7)$$

Vision-based neighbor selection further restricts this to agents within $r^{\max}$ and not occluded by others. Treating agents as spheres, occlusion is simplified to a planar check among three agents' centers:

$$N_i^{\text{visual}} = \left\{ \begin{array}{l} j \ne k \in N_i^{\text{metric}} \mid \\ \left(\theta_{ij} + \theta_{ik} > \alpha_{ijk} \wedge d_{ik} < d_{ij} \right) \end{array} \right\}, \quad (8)$$

where θ_{ij} and θ_{ik} are angular half-sizes of agents j and k from agent i's center, and α_{ijk} is their angular separation. This model captures visual perception constraints.

Delaunay selection uses Delaunay triangulation, where agent i is a neighbor of j if they share a ridge:

$$N_i^{\text{del}} = \left\{ j \in \mathrm{P} \mid \text{edge}(i, j) \text{ exists in } \mathrm{T}(\mathrm{P}) \right\}, \quad (9)$$

where P is the set of agent positions, and T is the triangulation. This typically yields up to 12 neighbors in 3D space. Topological selection includes the n nearest neighbors:

$$N_i^{\text{topo}} = \left\{ n - \arg\min d_{ij}, j \in \mathrm{A}_i \right\}. \quad (10)$$

We assess swarm performance using three metrics computed at each time step. The minimum nearest neighbor distance, indicating collision avoidance, is:

$$d^{\min} = \min_{i \ne j} d_{ij}. \quad (11)$$

A collision occurs if $d^{\min} < 2r$, where r is the agent radius. Alignment, measuring velocity vector correlation, is:

$$\phi^{\text{align}} = \frac{1}{N(N-1)} \sum_{i \ne j} \frac{v_i \cdot v_j}{\|v_i\| \|v_j\|}. \quad (12)$$

A value of one indicates perfect alignment, zero indicates disorder. The union metric, assessing swarm connectivity, is:

$$\phi^{\text{union}} = 1 - \frac{n^{\text{comp}} - 1}{N - 1}, \quad (13)$$

where n^{comp} is the number of connected components in the adjacency matrix. A value of zero indicates complete fragmentation.

We conducted 10 migration experiments to evaluate swarm performance across neighbor selection strategies. For topological selection, $n = 12$, matching the average Delaunay neighbor count. Agents are spawned randomly in a cube with predefined density, assigned a constant migration velocity $v^{\text{mig}} = [1, 0, 0]$ along the horizontal axis. Performance metrics are averaged over the last 25% of steps. Parameters are listed in Table 1.

We combine metric, topological, and Delaunay strategies with vision-based selection to evaluate vision-based localization. Two simulation environments are used: a point mass environment for statistical analysis (up to 150 agents) and Gazebo with PX4 for realistic quadrocopter dynamics (up to 70 agents, due to hardware limits). Gazebo's lockstep feature synchronizes velocity commands. The Gazebo environment validates point mass results in a realistic setting.

TABLE 1

EXPERIMENTAL PARAMETERS

Description	Notation	Value
Agent radius	r	0.25 m
Perception radius	$r^{\max}$	10 m
Maximum neighbors	n	12
Maximum speed	$v^{\max}$	1 m/s
Separation gain	k^{sep}	1 m/s
Cohesion gain	k^{coh}	2 m/s
Migration gain	k^{mig}	0.5 m/s
Time delta	Δt	0.1 s
Simulation duration	T	120 s

3. RESULTS

We evaluated swarm performance across sizes $N \in \{10, 30, 50, 70, 90, 110, 130, 150\}$ and neighbor selection strategies in a point mass simulation to identify the most effective strategy. Results are shown in Fig. 1.

Purely vision-based selection degrades as swarm size increases, with reduced average minimum distance and alignment. This can lead to collisions and route deviations, increasing battery usage. For swarms larger than 30 agents,

visual occlusions and frequent neighbor changes cause trajectory instability. Vision+metric and vision+topological strategies maintain stable alignment and neighbor counts but suffer from reduced union, indicating fragmentation. Vision+metric shows fragmentation from 10 agents, while vision+topological shows it from 30 agents. Vision+Delaunay provides consistent performance for swarms of 50+ agents, maintaining stable minimum distance, alignment, and neighbor count with perfect union. However, for swarms with fewer than 30 agents, it exhibits lower alignment due to insufficient neighbors. Overall, vision+Delaunay excels for larger swarms, despite slightly lower alignment.

We validated the vision+Delaunay strategy in Gazebo with quadrocopter dynamics for $N \in \{10, 30, 50, 70\}$. Results are shown in Fig. 2. The close agreement between environments confirms that vision+Delaunay enables effective formation control in realistic settings.

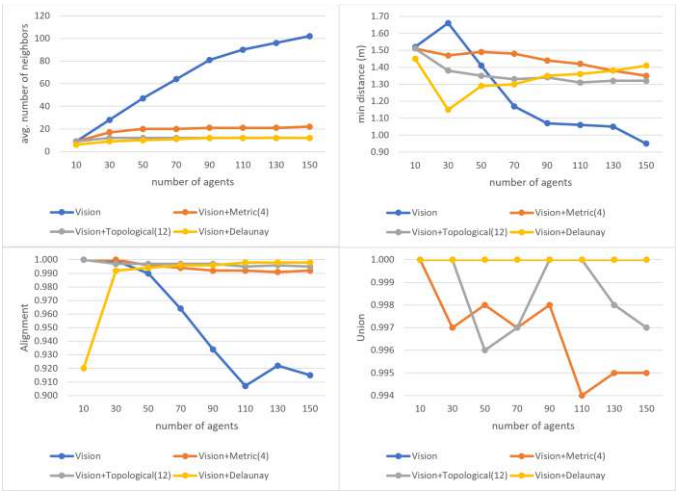

Fig. 1. Performance metrics (minimum distance, alignment, union, neighbor count) for different neighbor selection strategies across swarm sizes in point mass simulations

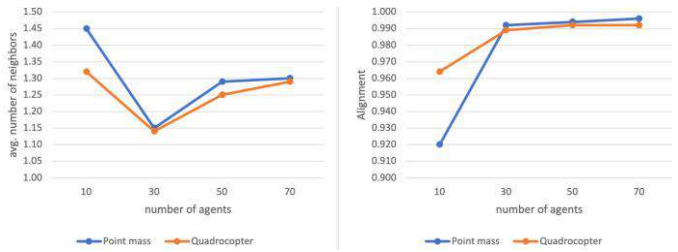

Fig. 2. Comparison of performance metrics (minimum distance, alignment, union) between point mass and quadrocopter dynamics for vision+Delaunay across swarm sizes

4. Discussion

The vision+Delaunay strategy outperforms others for large swarms, as it mitigates occlusion effects by prioritizing neighbors forming triangulation ridges, ensuring robust connectivity [2]. Vision+metric and vision+topological strategies, while simpler, lead to fragmentation, as seen in reduced union metrics, consistent with findings in [1]. The poor performance of vision+Delaunay for small swarms (less than 30 agents) likely stems from insufficient neighbor counts, limiting alignment, a problem also noted in [2] for sparse formations. Compared to prior work, our simplified visibility model reduces computational complexity by approximating occlusions in a planar framework, unlike the more complex models in [1]. However, the model assumes spherical agents, which may not capture real-world UAV shapes, and the artificial potential field requires careful parameter tuning, as noted in [4].

5. Conclusion

We evaluated UAV swarm performance under visual occlusions using a novel simplified visibility model, focusing on vision-based localization. Simulations across swarm sizes reveal that vision+metric and vision+topological strategies risk fragmentation, while vision+Delaunay enhances performance for swarms with 50+ agents, maintaining connectivity and stability, though it underperforms for smaller swarms. The model's planar occlusion check and spherical agent assumption may limit its applicability to complex UAV shapes, and the artificial potential field's sensitivity to tuning poses problems for real-world deployment. These findings suggest vision+Delaunay is well-suited for large-scale applications like search and rescue in cluttered environments. Future work could explore reinforcement learning [3] to improve scalability and address occlusion problems, or refine the visibility model to account for non-spherical agents.

References

[1] F. Schilling, F. Schiano, and D. Floreano, "Vision-Based drone flocking in outdoor environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2954–2961, Feb. 2021

[2] F. Schilling, E. Soria, and D. Floreano, "On the Scalability of Vision-Based Drone Swarms in the Presence of Occlusions," *IEEE Access*, vol. 10, pp. 28133–28146, Jan. 2022

[3] Y. Albrekht and A. Pysarenko, "Exploring the power of heterogeneous UAV swarms through reinforcement learning," *Technology Audit and Production Reserves*, vol. 6, no. 2(74), pp. 6–10, Dec. 2023

[4] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 25–34, Aug. 1987

Multi-UAV mission planning models

Koval Vadym, Zhdanova Oleva, Popenko Volodymyr
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

Abstract. **This paper focuses on the three variations of the UAV route planning problem, taking into account constraints on flight range and recognition radius. A novel approach is proposed that reduces more complex problem instances to parallel solutions of a basic problem without service points. This modular structure enhances computational efficiency and improves routing accuracy by enabling flexible distribution of targets among segments or individual UAVs.**

Keywords: unmanned aerial vehicles (UAVs), multi-UAV mission, path optimization, target inspection, route smoothing.

INTRODUCTION

The UAV is an autonomous drone that can fly missions independently of a human pilot [1]. In modern monitoring and reconnaissance tasks performed using unmanned aerial vehicles (UAVs), efficient planning of inspection routes plays a crucial role. Depending on the technical limitations of UAVs – such as flight range, recognition radius, and the availability of service points – it becomes necessary to formalize and solve various routing problem scenarios. Planning an acceptable trajectory, known as the UAV path planning problem [2], is necessary to enable any mission. An effective option for the use of UAVs is the solution of tasks by a group of UAVs acting as a team [3].

The paper considers three meaningful formulations of the UAV route planning problem with increasing complexity: from a single-UAV problem without intermediate service points (referred to as the basic problem) to a multi-UAV problem with multiple service points. The idea of constructing efficient routes for the basic problem is presented. The decomposition-based approach is proposed, which reduces more complex problems to a series of basic subproblems. This enables both improved computational efficiency and greater flexibility in constructing more accurate routes.

PROBLEM 1 (SINGLE UAV, NO SERVICE POINTS)

This problem represents the simplest case: it does not involve the use of intermediate service points.

Given: n – number of targets; $T = \{T_1, \ldots, T_j, \ldots, T_n\}$ – set of targets; (x_j, y_j) – coordinates of the target T_j $(j = 1, \ldots, n)$; one UAV; d – maximum flight range (km) of the UAV without recharging; r_j – maximum recognition range (km) of target T_j by the UAV camera; S – UAV's take-off point; (x_s, y_s) – coordinates of the UAV's take-off point; F – UAV's landing point; (x_F, y_F) – coordinates of the UAV's landing point. The UAV follows a route that begins at a starting point S, covers a number of targets, and ends at a final destination F. During the flight, the UAV inspects a target T_j if it is within range – that is, at a distance not exceeding the maximum recognition radius for that target r_j. The UAV's flight endurance allows it to travel a limited distance d without requiring energy replenishment. The objective is to determine a route that enables the UAV to inspect the maximum possible number of targets from the given set T.

Example of problem 1

Consider a scenario with $n = 12, k = 4$. Figure 1 shows the layout of the targets and the corresponding recognitions.

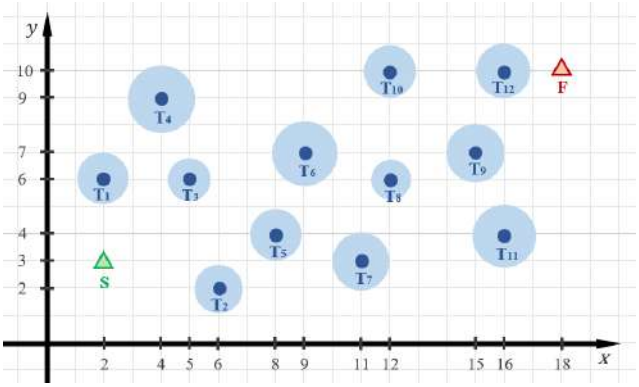

Fig. 1. The layout of the targets and the corresponding recognition zones

In the classical approach to UAV route planning, it is assumed that the vehicle flies directly through the center of each target to be inspected. While this trajectory ensures full coverage, it is often redundant in terms of overall route length.

In the classical route construction approach, each breakpoint corresponds to the center of one of the targets, meaning the

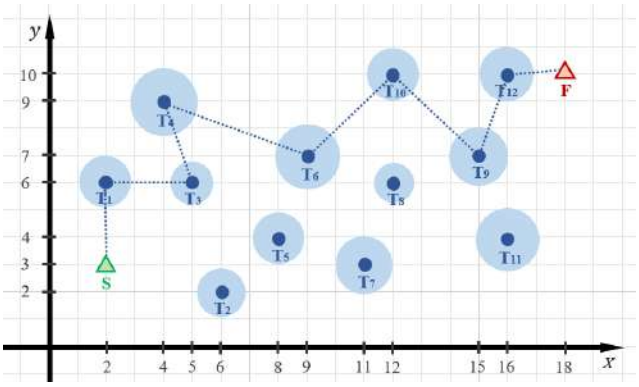

Fig. 2. UAV route passing through the centers of the targets

UAV flies directly over the geometric center of the inspected target. Figure 2 illustrates the example of the classical route.

A smoothed route is a polyline or curve that does not pass through the target centers but instead approaches them within the allowable recognition radius. In other words, a target is considered recognized if the distance from any point along the route to the target does not exceed a given threshold.

This approach allows for:

– reducing the total route length by avoiding unnecessary detours;

– maintaining inspection efficiency without decreasing the number of covered targets;

– providing greater flexibility in route design, especially in cases with flight distance constraints between service points.

So, the length of the constructed routes can be significantly reduced by smoothing them – i.e., by forming trajectories in such a way that the UAV flies near the target, not necessarily through its center, but within the permissible recognition zone. Figure 3 includes an example of the smoothed route for the UAV; for clarity, the corresponding unsmoothed trajectories are also displayed.

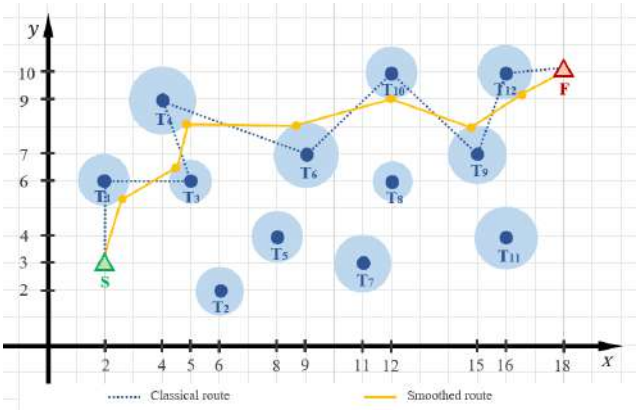

Fig. 3. Classical and smoothed routes of UAV

When smoothing UAV flight routes, it is important to recognize that this method is not always optimal for constructing sub-routes. In certain cases – depending on the movement direction from the current point to the next two (i.e., the shape of the resulting polyline) – it may be more effective to use the classical sub-route format, where the UAV flies directly over the center of the target, rather than implementing a smoothed path that only touches the edge of the target area within the recognition range of the onboard camera.

PROBLEM 2 (SINGLE UAV AND SERVICE POINTS)

This task is a generalized case of the previous one.

Given: n – number of targets; $T = \{T_1, ..., T_j, ..., T_n\}$ – set of targets; (x_j, y_j) – coordinates of the target $T_j (j = 1, ..., n)$; one UAV; d – maximum flight range (km) of the UAV without recharging; r_j – maximum recognition range (km) of target T_j by the UAV camera; S – UAV's take-off point; (x_S, y_S) – coordinates of the UAV's take-off point; F – UAV's landing point; (x_F, y_F) – coordinates of the UAV's landing point. k – number of service points for the UAV; $R = \{R_1, ..., R_k\}$ – set of intermediate landing points.

The UAV follows a route that starts at the starting point S, covers certain targets, and ends at the finishing point F. During the flight, the UAV inspects the target T_j, which is within range, i.e., at a distance not exceeding the maximum recognition radius r_j. The maximum flight resource of the UAV allows it to cover a distance of up to d kilometers without recharging. Thus, the distance between any two consecutive route points (including the starting, intermediate, and finishing points) must not exceed d. The UAV has k service points. These intermediate points $R_1, ..., R_k$ can be used for technical maintenance or resource replenishment and are part of the set of permissible landing points on the route. Each point may be visited more than once. The task is to determine the UAV routes that allow inspecting all targets from the set T in the minimum time.

Example of problem 2

Consider the task with parameters: $n = 12, k = 4$. Figure 4 illustrates the inspection routes of certain targets by a single UAV using service points.

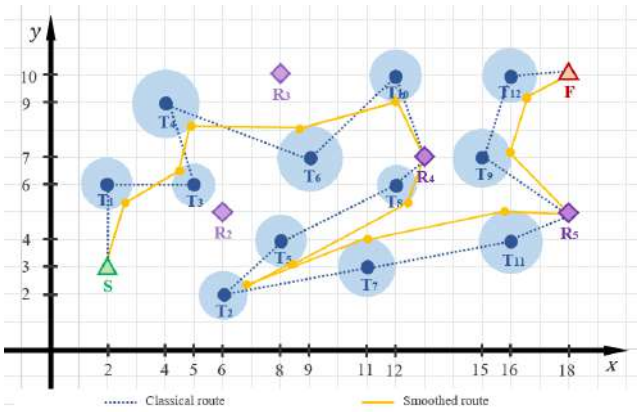

Fig. 4. Route variants: through target centers vs. smoothed path with service stops (with stops at intermediate service points)

PROBLEM 3 (MULTIPLE UAVS AND SERVICE POINTS)

Given: n – number of targets; $T = \{T_1, ..., T_j, ..., T_n\}$ – set of targets; (x_j, y_j) – coordinates of the target $T_j (j = 1, ..., n)$; m – number of UAVs; $U = \{U_1, ..., U_i, ..., U_m\}$ – set of UAVs; d_i – maximum flight range (km) of the UAV U_i without recharging the flight resource $(i = 1, ..., m)$; r_{ij} – maximum recognition range (km) of target T_j by the camera installed on the UAV U_i $(i = 1, ..., m; j = 1, ..., n)$; S_i – take-off point of UAV $U_i (i = 1, ..., m)$; (x_i^S, y_i^S) – coordinates of the take-off point of the UAV $U_i (i = 1, ..., m)$; F_i – landing point of UAV $U_i (i = 1, ..., m)$; (x_i^F, y_i^F) – coordinates of the landing point of the UAV $U_i (i = 1, ..., m)$; k_i – number of service points for UAV $U_i (i = 1, ..., m)$; $R_i = \{R_1^i, ..., R_{k_i}^i\}$ – set of intermediate landing points for the UAV $U_i (i = 1, ..., m)$.

Each UAV U_i moves along a route that starts at the starting point S_i, covers certain targets, and ends at the finishing point F_i. During the flight, the UAV inspects a target T_j that is within its range, meaning the target is within the maximum recognition radius r_j set for the UAV-target pair. The maximum flight range of the UAV U_i allows it to travel a distance d_i without recharging that does not exceed a certain limit. Therefore, the distance between any two consecutive points on the route (including the starting, intermediate, and finishing points) must not exceed this limit d_i.

Each UAV U_i is assigned k_i of service points. These intermediate points $R_1^i, ..., R_{k_i}^i$ can be used for maintenance or

resource replenishment and are part of the set of permissible landing points along the route. Each point may be visited more than once. The task is to determine the UAV routes that allow inspecting all the targets in the set T in the minimum time.

Example of problem 3

To illustrate the above, consider a task with the parameters: $n = 12, m = 2, k_1 = 3, k_2 = 3$.

It is assumed that $r_{1j} = r_{2j}$, $j = 1,...,12$, meaning that the cameras installed on both UAVs have the same recognition range characteristics. Figure 5 illustrates the inspection routes of certain targets by two UAVs using service points.

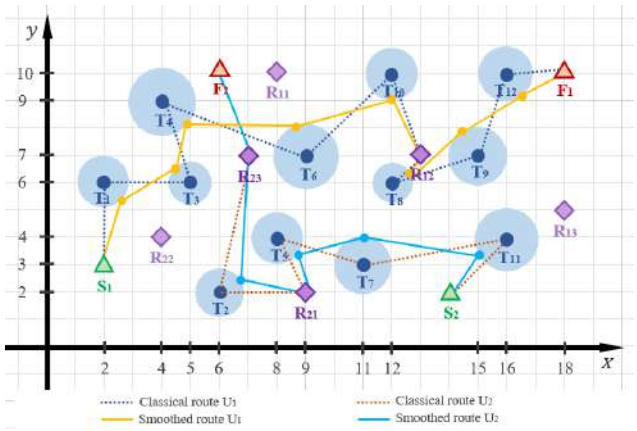

Fig. 5. Classical and smoothed routes for two-UAVs mission

General algorithms description

Task 1 is a fundamental (basic block) for the algorithms solving problems 2 and 3. The idea of the algorithm for solving problem 2 is to treat each permissible subsequence of route points (including service points) as a separate implementation of task 1, and then combine the obtained partial solutions. The idea of the algorithm for solving problem 3 is based on solving the corresponding task 2 for each UAV. At the same time, the process of route construction can be performed in parallel, which not only increases computational efficiency but also allows flexible decisions regarding the distribution of targets among UAVs. This, in turn, promotes the formation of more accurate and balanced routes, as each target can be assigned to the UAV for which its inclusion is the most beneficial in terms of route length and coverage. In general, for this type of task such algorithms as ant colony optimization (MMACO) and artificial Bee Colony (ABC) are considered among the most appropriate [4].

Mathematical model of Problem 1

Variables (quantities to be determined)

$M = (v_0, v_1, ..., v_k)$ is the UAV route, where $v_0 = S$, $v_k = F$, $v_i \in R^2$ – the coordination of the route breakpoints.

Constraints

1) Target coverage constraint: considering that a target can be surveyed not only at the waypoints but from any point along a route segment (i.e., between two waypoints), if the distance from the target to the segment does not exceed the recognition radius, the constraint is as follows: for each target T_j to be considered surveyed, there must exist at least one segment of the route $[v_p, v_{p+1}]$, $p = 0,...,k$, such that the distance from T_j to the segment does not exceed the recognition radius of the target r_j:

$$z_j = 1, \text{if } \exists p \in \{0,...,k-1\} \text{ that } D(T_j, [v_p, v_{p+1}]) \le r_j, \quad (1)$$

where $D(T_j, [v_p, v_{p+1}])$ is the shortest Euclidean distance from a point T_j to a line segment $[v_p, v_{p+1}]$.

2) Route length constraint: the total length of the route must not exceed a given maximum distance d:

$$\sum_{p=0}^{k-1} |v_p, v_{p+1}| \le d, \quad (2)$$

where $|v_p, v_{p+1}|$ is the length of the segment $[v_p, v_{p+1}]$.

3) Binary constraint for target survey: $z_j \in \{0,1\}$, $j = 1,...,n$.

4) Waypoint coordinate constraint: the coordinates of the waypoints $v_1,...,v_{k-1}$ can be arbitrary, selected from a set of permissible points, or determined in a specific manner.

Auxiliary problem

Calculating the distance from a point $T_j = (x_j, y_j)$ to a segment $[A, B]$, where $A = (x_A, y_A)$, $B = (x_B, y_B)$ involves two steps. First, a projection parameter is calculated:

$$t = \frac{(x_j - x_A)(x_B - x_A) + (y_j - y_A)(y_B - y_A)}{(x_B - x_A)^2 + (y_B - y_A)^2}. \quad (3)$$

Then the shortest distance from the point to the segment is determined using that parameter:

$$D(T_j, [v_p, v_{p+1}]) = \begin{cases} \sqrt{(x_j - x_A)^2 + (y_j - y_A)^2}, & \text{if } t < 0, \\ \sqrt{(x_j - x_B)^2 + (y_j - y_B)^2}, & \text{if } t > 1, \\ \sqrt{(x_j - x_t)^2 + (y_j - y_t)^2}, & \text{if } 0 \le t \le 1. \end{cases} \quad (4)$$

Conclusions

Three problem formulations are proposed. The complex problems (1 and 2) are reduced to a series of independent subtasks of type 3. In the case of problem 2, the route is presented as a sequence of segments between permissible points (start, service points, finish), each of which is solved as a separate task 3. Problem 1 is reduced to the parallel solution of problem 2 for each UAV. This approach ensures modularity and allows for an efficient distribution of targets among UAVs, taking into account coverage and route length. The work formulates a mathematical model for task 3, which considers the ability to recognize a target not only at the route points but also along its segments. The proposed hierarchical task reduction scheme allows for solving UAV route planning problems, taking into account practical constraints. Task 3 serves as a universal component that enables the effective construction of more complex routes and the scalability of the approach to a larger number of devices.

References

[1] A. Barnawi, K. Kumar, N. Kumar, N. Thakur, B. Alzahrani, and A. Almansour, "Unmanned aerial vehicle (UAV) path planning for area segmentation in intelligent landmine detection systems," Sensors, vol. 23, no. 16, p. 7264, Aug. 2023.

[2] S. Ghambari, M. Golabi, L. Jourdan, J. Lepagnot, and L. Idoumghar, "UAV path planning techniques: A survey," RAIRO - Operations Research, vol. 58, no. 4, pp. 2951–2989, Mar. 2024.

[3] L. Hulianytskyi and O. Rybalchenko, "Optimization of decisions when planning a UAV group mission with alternative depots," in *Proc. III Int. Sci. Symp. "Intelligent Solutions" (IntSol-2023)*, Kyiv, Ukraine, Sep. 27–28, 2023, *CEUR Workshop Proc.*, vol. 3538, pp. 245–256, 2023.

[4] M. Rahman, N. I. Sarkar, and R. Lutui, "A survey on multi-UAV path planning: Classification, algorithms, open research problems, and future directions," Drones, vol. 9, no. 4, p. 263, Mar. 2025

Model and Method for Forming Service Packages for Infocommunication Providers

Zhdanova Olena, Bukasov Maksym, Tsymbal Swiatoslav,
Chymshyr Viacheslaw, Omelchenko Rostyslav
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
omelchenkorv@gmail.com

Abstract. **The problem of forming service packages for infocommunications providers is considered. Its formal formulation is proposed, aimed at increasing the efficiency of IT companies and providers. A specific feature of the model is the orientation towards the modern concept of ensuring mutual benefit of IT companies and infocommunications providers. The choice of methods that form the basis for the implementation of the service package formation system is justified. A combined method of forming service packages is proposed. The results of an experimental study of methods for forming service packages are presented.**

Keywords*: *IT company, infocommunications provider, service life cycle, service package, model, method, information system.

Модель і метод формування пакетів сервісів для провайдерів інфокомунікацій

Жданова Олена, Букасов Максим, Цимбал Святослав,
Чимшир В'ячеслав, Омельченко Ростислав
КПІ імені Ігоря Сікорського
м.Київ, Україна
omelchenkorv@gmail.com

Анотація. **Розглянуто проблему формування пакетів сервісів для провайдерів інфокомунікацій. Запропоновано її формальну постановка, спрямовану на підвищення ефективності діяльності ІТ-компаній і провайдерів. Специфічною особливістю моделі є орієнтація на сучасну концепцію забезпечення взаємної вигоди ІТ-компаній і провайдерів інфокомунікацій. Обгрунтовано вибір методів, покладених в основу реалізації системи формування пакетів сервісів. Запропоновано комбінований метод формування пакетів сервісів. Наведено результати експериментального дослідження методів формування пакетів сервісів.**

Ключові слова: ІТ-компанія, провайдер інфокомунікацій, життєвий цикл сервісів, пакет сервісів, модель, метод, інформаційна система.

Вступ

З розвитком інформаційних технологій відбуваються зміни у всіх галузях діяльності людини. Одним із напрямів змін є широке впровадження сервісного підходу. Сформувався новий науковий напрям Service Science, спрямований на наукове розв'язання проблем, які стримують поширення сервісного підходу [1, 2]. Особливо важливі проблеми виникають при впровадженні сервісного підходу в галузі інфокомунікацій, оскільки тут потрібні рішення, які охоплюють бізнесовий, виробничий і технологічний рівні інформаційних систем провайдерів, які забезпечують комплексну підтримку бізнес-процесів їх діяльності [3 – 5]. І хоч сервісний підхід має довгу історію впровадження в галузі інфокомунікацій, залишаються багато проблем, які вимагають ефективних розв'язків.

1 Формування пакетів сервісів

Діяльність провайдерів інфокомунікацій є об'єктом уваги багатьох ІТ-компаній, які розробляють інформаційні технології для підтримки бізнес-процесів провайдерів. Загалом в інформаційних системах провайдерів інфокомунікацій інтегровані різноманітні рішення, які забезпечують підтримку понад сотні бізнес-процесів. Загалом створення інформаційних систем для провайдерів інфокомунікацій, які підтримують усі бізнес-процеси становить складну науково-практичну проблему. Сьогодні її вирішення базується на концепціях життєвого циклу сервісів [6], E2E [7], платформних рішень [8], ISTM [9], передбачає широке використання технологій штучного інтелекту, насамперед

LLM, RAG-систем, інтелектуальних агентів і проектування та реалізацію в рамках сучасних методологій створення ІС у комплексних середовищах із застосуванням ефективних інструментів. Для ефективної підтримки кожного з понад сотні бізнес-процесів треба сформулювати і вирішити відповідну локальну проблему, розробивши моделі і методи і створивши на їх основі відповідні рішення. Усі рішення локальних проблем необхідно інтегрувати, щоб забезпечити підтримку діяльності провайдера згідно його цілей.

Однією з найцікавіших проблем, яка очікує ефективного розв'язання в умовах змін, які відбуваються в галузі, є проблема формування пакетів сервісів для провайдерів інфокомунікацій. Загалом ця проблема має дві сфери застосування – формування пакетів сервісів, які ІТ-компанія надає провайдерам інфокомунікацій, і формування пакетів сервісів, які провайдер інфокомунікацій надає своїм клієнтам. У доповіді зазначена проблема розглядається у першому аспекті.

2 Огляд існуючих рішень

Проблема формування пакетів сервісів для провайдерів інфокомунікацій досить давно є предметом уваги дослідників. Тут напрацьовано багато різних методів. У першу чергу, варто згадати статистичні методи [10] і методи, побудовані на основі кластеризації [11]. Як перший, так і другий методи побудовані на основі накопиченого досвіду надання сервісів, поєднаних в пакети. Ці методи широко використовуються в інших галузях, але виявилися корисними і для формування пакетів сервісів у галузі інфокомунікацій.

Щоб результативно врахувати специфіку задачі формування пакетів сервісів у галузі інфокомунікацій, доцільно розширити множину методів, які можна використовувати для її розв'язання. Для врахування сучасних тенденцій до надання взаємовигідних пакетів сервісів потрібно побудувати моделі математичного програмування, які включають цільові функції і обмеження стосовно формування пакетів сервісів. Тоді виникне потреба використовувати інші методи, зокрема ефективні методи математичного програмування, евристичні методи.

У літературі можна знайти моделі і методи розв'язання задачі формування пакетів сервісів у галузі інфокомунікацій, наприклад [12]. Але специфічні особливості задачі формування пакетів сервісів у різних умовах вимагають нових прийомів їх врахування у моделях і нових методів розв'язання. Зокрема, у доповіді буде досліджено можливість використання комбінації евристичних алгоритмів для швидкого розв'язання задачі формування пакетів сервісів у відомій постановці.

3 Формальна постановка задачі формування пакетів сервісів

Задача формування пакетів сервісів розглядається з точки зору ІТ-компанії, яка надає множину сервісів $S = \{S_1, \dots, S_i, \dots S_k\}$, де k – кількість сервісів, провайдерам інфокомунікацій з множини $P = \{P_1, \dots, P_j, \dots, P_m\}$, де m – кількість провайдерів. Взаємозв'язки сервісів визначає множина R_i, $i = 1, \dots, k$ підмножин вигляду $\{S_1^i, \dots, S_{n_i}^i\}$, $S_g^i \in S$, $g \in [1, n_i]$ таких, що надання сервісу S_g^i залежить від кожного з n_i сервісів $S_1^i, \dots, S_{n_i}^i$ (тобто у своїй сукупності ці сервіси становлять собою множину взаємозалежних сервісів).

Традиційним для бізнесової діяльності ІТ-компанії у цій галузі є ведення матриці преференційних цін за надання сервісів $\mathbf{D} = \| d_{ij} \|$, де d_{ij} – базова ціна ІТ-компанії за надання сервісу S_i провайдеру P_j, $i = 1, \dots, k$, $j = 1, \dots, m$.

ІТ-компанія за надання сервісу S_i провайдеру P_j може надавати знижки r_{ij} до преференційної ціни d_{ij}, $i = 1, \dots, k$, $j = 1, \dots, m$ ($0 \le r_{ij} < 1$).

Для врахування складу сервісів пакетів ІТ-компанії вводяться булеві змінні v_{ij}:

$$v_{ij} = \begin{cases} 1, \text{ якщо сервіс } S_i \text{ входить до портфеля провайдера } j, \\ 0, \text{ якщо сервіс } S_i \text{ не входить до портфеля провайдера } j. \end{cases}$$

Задача полягає у тому, щоб для кожного провайдера знайти пакет сервісів, які надає ІТ-компанія, та відповідні знижки до преференційної ціни, що забезпечують максимальну сумарну вигоду як для ІТ-компанії, так і для кожного з її провайдерів. Відповідні цільові функції мають вигляд:

1) максимізація сумарного доходу ІТ-компанії від надання пакетів сервісів:

$$W = \sum_{j=1}^{k} \sum_{i=1}^{m} d_{ij}(1 - r_{ij}) v_{ij}, \quad (1)$$

2) максимізація сумарного доходу кожного провайдера P_j ($j = 1, \dots, m$) від надання клієнтам сервісів, запропонованих ІТ-компанією:

$$Q_j = \sum_{i=1}^{m} (p_{ij} - d_{ij}(1 - r_{ij})) v_{ij}, \quad (2)$$

де p_{ij} – дохід провайдера P_j від надання сервісу S_i своїм клієнтам.

Наведені вирази для цільових функцій оцінки якості розв'язків з точки зору ІТ-компанії та провайдерів дозволяють врахувати сучасну концепцію ведення бізнесу – забезпечення взаємної вигоди ІТ-компанії та кожного з провайдерів.

Мають місце наступні обмеження:

1) обмеження на ресурси ІТ-компанії, необхідні для надання усіх сервісів, включених до пакетів усіх провайдерів:

$$\sum_{i=1}^{k} \sum_{j=1}^{m} \beta_{ijl} v_{ij} \le T_l, \; l = 1, \dots, L, \quad (3)$$

де β_{ijl} – кількість ресурсу l, що використовується для надання сервісу S_i провайдеру P_j, T_l – загальна кількість ресурсу l ІТ-компанії (у загальному випадку одиниць ресурсу l за плановий період), L – кількість видів обмежених ресурсів;

2) обмеження на відносну цінність сервісів для провайдера, які гарантують що кожен сервіс, що надається провайдером, є для нього економічно доцільним, тобто забезпечує прийнятний рівень прибутковості:

$$s_{ij} v_{ij} \le \frac{p_{ij}}{b_{ij}}, \; i = 1, \dots, k, \; j = 1, \dots, m, \quad (4)$$

де p_{ij} – дохід провайдера P_j, пов'язаний з наданням сервісу S_i, b_{ij} – витрати провайдера P_j, пов'язані з наданням сервісу S_i, s_{ij} – мінімальна відносна цінність сервісу S_i, що є прийнятною для провайдера P_j;

3) показники SLA надаваних сервісів:

$$a_{ijg} v_{ij} \le \alpha_{ig}, \; i = 1, \dots, k, \; j = 1, \dots, m, \; g = 1, \dots, G, \quad (5)$$

де a_{ijg} – значення показника SLA g сервісу S_i, що надається провайдеру P_j, α_{ig} – значення показника SLA g сервісу S_i, яке визначене в умовах про надання цього сервісу своїм клієнтам, G – кількість показників SLA сервісів;

4) обмеження міжсервісної залежності:

$$v_{ij} v_{lj} = \rho_{il}, \; i = 1, \dots, k, \; j = 1, \dots, m, \; l = 1, \dots, k, \quad (6)$$

де параметр ρ_{il} задає зв'язок між сервісами S_i та S_l у випадках, коли надання одного сервісу вимагає наявності іншого: $\rho_{il}=1$, якщо сервісу S_i у пакеті провайдера потрібен сервіс S_l, $\rho_{il}=0$ в протилежному випадку.

У реальних умовах залежності (6) можуть мати складнішу структуру, що враховує багаторівневі або умовні зв'язки між групами сервісів.

Загалом вигляд цільових функцій і обмежень зумовлений специфікою бізнесу ІТ-компанії та провайдерів інфокомунікацій. Усі процеси діяльності ІТ-компаній і провайдерів, зокрема пов'язані з формуванням пакетів сервісів, спрямовані на досягнення власних бізнесових цілей. Це пояснює наявність двох цільових функцій, одна з яких використовується для оцінки пакета сервісів з точки зору кожного провайдера окремо.

Сьогодні на прийняття рішень при веденні бізнесової діяльності впливає концепція взаємної вигоди. Це вимагає врахування вигоди як ІТ-компанії, яка надає сервіси, так і провайдера, який отримує сервіси. Цим пояснюється вигляд цільових функцій і введені параметри, які дозволяють досягти взаємну вигоду за допомогою системи знижок.

У доповіді з відомих систем знижок, які ІТ-компанії можуть надавати провайдерам – на певні сервіси, на кожен додатковий сервіс, який додається до пакета, на кожну додаткову групу, яка додається до пакета, в залежності від кількості сервісів у пакеті та ін. – розглянуто тільки знижки на кожен сервіс, який надається провайдерам.

Стосовно обмежень, що накладаються на розв'язки. Ресурсні обмеження встановлюються для ІТ-компанії і забезпечують ресурсну реалістичність формованих пакетів. Вони дають змогу врахувати обмеження за кожним видом ресурсу, необхідним для підтримки надання всіх сервісів, включених до пакетів для всіх провайдерів. Загалом ідеться про людські, обчислювальні, комунікаційні та технологічні ресурси, які надаються за різними моделями.

Другу групу складають обмеження з боку провайдерів, які поділяються на економічні та якісні. Вони залежать від типу сервісів, які надає провайдер, і специфіки його клієнтів. У межах цієї моделі враховано:

– серед економічних обмежень – обмеження на відносну цінність сервісів (4),

– серед якісних – обмеження на показники SLA (5).

Останні є критично важливими для провайдера, оскільки дозволяють гарантувати якість сервісів, що надаються його клієнтам. Включення SLA-показників до угоди між провайдером і клієнтом фіксує очікуваний рівень якості послуг. Відповідно, ці обмеження мають бути відображені у формальній постановці задачі формування пакетів сервісів, щоб забезпечити відповідність очікуванням клієнтів і знизити ризики невиконання договірних зобов'язань.

Третю групу складають технологічні обмеження, до яких передовсім належать обмеження міжсервісної залежності. Вони призначені для забезпечення наявності у пакеті для кожного сервісу всіх сервісів, від яких він залежить. Ці обмеження пов'язані з тим, що надання окремих сервісів може вимагати підтримки деяких їх функцій від інших сервісів. Формально це обмеження полягає у тому, що надання, наприклад, сервісу S_i може вимагати розміщення на тих же ресурсах іншого сервісу, наприклад, сервісу S_t, який повинен виконувати певні операції для підтримки сервісу S_i. У загальному випадку можна говорити, що сервіс може залежати від підмножини сервісів з каталогу сервісів.

Звичайно для визначення обмежень міжсервісної залежності при формуванні пакетів сервісів для кожного провайдера використовується розбиття сервісів на групи взаємопов'язаних сервісів, як це показано в роботі [12]. Тоді обмеження цієї групи вимагають для кожного сервісу, який включається до пакету, включення всієї групи сервісів, до якої цей сервіс належить. Варто зауважити, що множини R_i, $i \in [1, k]$ формуються на основі цих обмежень.

Опрацювавши формальну постановку задачі формування пакетів сервісів, можемо переходити до вибору відповідних методів її розв'язання, які будуть реалізовані в системі формування пакетів сервісів.

4 Комбінований метод розв'язання задачі формування пакетів сервісів

Важливими для вибору методу розв'язання сформульованої вище задачі є специфічні її особливості. По-перше, формальна постановка містить змінні двох видів – v_{ij} та r_{ij}, які визначають структуру пакетів і знижки. По друге, сформульовані критерії максимізації цільової функції вигоди для ІТ-компанії (1) і провайдерів (2) враховують надання знижок. По третє, сформульовані обмеження (3) – (6) кількох груп обумовлюють необхідність для провідних ІТ-компаній і великих провайдерів розв'язання задач великої розмірності.

У таких умовах логічним виходом буде використання комбінованих методів розв'язання багатокритеріальних задач, які інтегрують у новий спосіб відомі прийоми: побудови на основі багатокритеріальної задачі однокритеріальних підзадач для ІТ-компанії і провайдерів; використання для розв'язання визначених підзадач спеціальних алгоритмів, які враховують їх специфічні особливості; визначення спільної частини у розв'язках підзадач для ІТ-компанії і провайдерів; використання спеціальних евристичних процедур для доповнення знайденої спільної частини до повного найкращого рішення початкової багатокритеріальної задачі з врахуванням системи знижок.

У доповіді пропонується варіант комбінованого методу розв'язання задачі формування пакетів сервісів, який поєднує описані вище підходи. Спочатку уточнимо згадані у описі ідеї методу підзадачі.

А. Підзадача для ІТ-компанії:

– критерій: максимізація цільової функції (1);

– обмеження: (3) – (6).

Б. Підзадача для провайдерів:

– критерій: максимізація цільової функції (2);

– обмеження (3) – (6).

Залишається лише визначити порядок застосування вибраних прийомів. Спочатку на першому етапі розв'язуємо дві однокритеріальні задачі, представлені описаними вище підзадачами А та Б. Їх розв'язками будуть пакети сервісів, найвигідніші для ІТ-компанії і для провайдерів відповідно, тобто пакети які забезпечать максимальний дохід для ІТ-компанії і провайдерів відповідно.

Потім на другому етапі для кожного з отриманих розв'язків підзадач для ІТ-компанії (для провайдерів) формуємо розв'язок відповідної симетричної підзадачі для провайдерів (для ІТ-компанії).

На третьому етапі застосовується процедура поліпшення отриманих на попередньому етапі спільних частин розв'язків підзадачі А (Б) та Б (А).

Відповідно четвертий етап полягає у виборі кращого з двох поліпшених розв'язків.

Нижче наведено опис варіанту комбінованого методу розв'язання задачі формування пакетів сервісів.

Етап 1. Розв'язання підзадач А, Б за допомогою ймовірнісно-жадібного алгоритму.

Підетап 1.1. Використання ймовірнісно-жадібного алгоритму для розв'язання підзадачі А і отримання найвигіднішого для ІТ-компанії пакету сервісів за умови виконання усіх обмежень (ресурсних, провайдера, міжсервісної залежності).

Підетап 1.2. Використання ймовірнісно-жадібного алгоритму для розв'язання підзадачі Б і отримання для кожного провайдера P_j (j = 1, ... , m) найвигіднішого пакету сервісів за умови виконання усіх обмежень (ресурсних, провайдера, міжсервісної залежності).

Етап 2. Визначення на основі розв'язків підзадач А, Б розв'язків симетричних підзадач Б і А.

Підетап 2.1. Пошук на основі отриманого на етапі 1 розв'язку підзадачі для ІТ-компанії розв'язку відповідної підзадачі для провайдерів.

Підетап 2.2. Пошук на основі отриманих на етапі 2 розв'язків підзадачі для провайдерів розв'язку відповідної підзадачі для ІТ-компанії.

Етап 3. Використання евристичної процедури для поліпшення отриманої на етапі 2 спільної частини у парах розв'язків підзадачі А і її симетричної підзадачі та підзадачі Б і її симетричної підзадачі з врахуванням системи взаємовигідних для обох сторін знижок.

Підетап 3.1. Поліпшення на основі евристичної процедури визначеної на підетапі 1 етапу 2 спільної частини у розв'язках підзадачі А для ІТ-компанії і симетричної задачі для провайдерів P_j (j = 1, ... , m).

Підетап 3.2. Поліпшення на основі евристичної процедури визначеної на підетапі 2.2 спільної частини у розв'язках підзадачі Б для провайдерів P_j (j = 1, ... , m) і симетричної задачі для ІТ-компанії.

Етап 4. Вибір кращого з розв'язків підетапів 3.1 і 3.2.

Для виділених підзадач ІТ-компанії і провайдерів ефективним рішенням виявився ймовірнісно-жадібний алгоритм. Він враховує спільні особливості підзадач А та Б, а також наявність багатьох провайдерів, для кожного з яких треба розв'язати підзадачу Б, забезпечуючи дотримання спільних для всіх провайдерів обмежень. Для виділення спільної частини у розв'язках виділених підзадач ІТ-компанії і провайдерів запропонована досить ясна процедура, яка враховує структури розв'язків і обмежень. А для поліпшення спільної частини у розв'язках виділених підзадач ІТ-компанії і провайдерів запропонована евристична процедура з врахуванням системи знижок.

5. Алгоритми реалізації комбінованого методу розв'язання задачі формування пакетів сервісів

5.1. Ймовірнісно-жадібний алгоритм розв'язання підзадачі А етапу 1

Тут і далі під сервісом будемо розуміти групу взаємопов'язаних сервісів. Також припустимо, що лімітуючим є лише один ресурс ІТ-компанії (тобто в обмеженнях (3) L=1, і у величин β_{ijl} можна опустити третій індекс, тобто $\beta_{ij1} = \beta_{ij}$). Таке спрощення є доцільним у зв'язку з тим, що у більшості практичних ситуацій саме один тип ресурсу (наприклад, людський, обчислювальний або часовий) є критичним та обмежує можливість надання сервісів, решта ресурсів або є надлишковими, або не досягають граничних значень. Це дозволяє сконцентрувати увагу на ключових аспектах задачі формування пакетів сервісів без втрати загальності моделі. Також припустимо, що обмеження (4)-(5) виконуються.

Для швидкого отримання близьких до оптимального розв'язків підзадач у евристичному алгоритмі розв'язання підзадач етапу 1 втілені кращі риси жадібного і ймовірнісного підходів. Від першого алгоритм успадкував ідею пріоритетності призначень $S_i - P_j$ з високою ефективністю використання ресурсу. Оскільки цінність кожної пари «сервіс – провайдер» (θ_{ij}) прямо залежить від співвідношення доходу до витрат ресурсу, що відповідає класичному жадібному критерію "максимум вигоди на одиницю витрат", кращі (ефективніші) призначення мають вищі шанси на вибір.

Від другого алгоритм успадкував ідею використання ймовірнісної стратегії для уникнення пастки локальних екстремумів. Евристичний алгоритм використовує ймовірнісний вибір, на відміну від детермінованих жадібних алгоритмів, які завжди обирають найкращу за критерієм пару. Такий ймовірнісний вибір, за умови реалізації якого шанси кожного допустимого призначення пропорційні його цінності θ_{ij}, і дозволяє уникнути локальних екстремумів. Дійсно, такий підхід додає елемент дослідження простору розв'язків. А оскільки алгоритм запускається багаторазово, з'являється можливість локалізувати кращі комбінації за рахунок повторного випадкового вибору.

Спочатку для кожної пари «сервіс – провайдер» обчислюється так звана цінність одиниці ресурсу, яка визначається як відношення преференційної ціни надання сервісу до обсягу ресурсу, необхідного для його підтримки. З урахуванням знижки це розраховується за формулою:

$$\theta_{ij} := \frac{d_{ij}(1 - r_{ij})}{\beta_{ij}}.$$

На кожному кроці алгоритму:

– формується множина дозволених призначень – це ті пари «сервіс S_i – провайдер P_j», для яких провайдер ще не отримав даний сервіс, і для яких залишкового (вільного) ресурсу достатньо для його надання;

– з-поміж дозволених пар одна вибирається випадковим чином, з ймовірністю, пропорційною її цінності.

5.2. Ймовірнісно-жадібний алгоритм розв'язання підзадачі Б етапу 1

Ідея ймовірнісно-жадібного алгоритму розв'язання підзадачі для провайдерів подібна до ідеї ймовірнісно-жадібного алгоритму розв'язання підзадачі А, але усі дії виконуються з точки зору вигоди для провайдерів. Для цього передбачена наявність циклу по провайдерам. Спочатку для кожної пари «сервіс S_i – провайдер P_j» обчислюється так звана цінність одиниці ресурсу, яка визначається як відношення доходу провайдера сервісу до обсягу ресурсу, необхідного ІТ-компанії для його підтримки. З урахуванням знижки це розраховується за формулою:

$$\theta_{ij} := \frac{p_{ij} - d_{ij}(1 - r_{ij})}{\beta_{ij}}.$$

На кожному кроці алгоритму:

– обирається провайдер $P_{j'}$ (наприклад, по циклу або з найменшим поточним сумарним доходом);

– для нього формується множина дозволених призначень – це ті пари «сервіс S_i – провайдер P_j» для яких залишкового (вільного) ресурсу ІТ-компанії достатньо для його надання;

– з-поміж дозволених пар одна вибирається випадковим чином, з ймовірністю, пропорційною її цінності θ_{ij}.

5.3. Алгоритми розв'язання підзадач етапу 2

Ідея алгоритму пошуку на основі отриманого на етапі 1 розв'язку підзадачі для ІТ-компанії (підзадачі для провайдерів) розв'язку відповідної підзадачі для провайдерів (підзадачі для ІТ-компанії) очевидна. Він полягає в наступному:

– для кожної включеної у пакет сервісів пари «сервіс S_i, провайдер P_j» розв'язку підзадачі для ІТ-компанії (підзадачі для провайдерів) включаємо у пакет сервісів пару «сервіс S_i, провайдер P_j» розв'язку підзадачі для провайдерів (підзадачі для ІТ-компанії);

– обчислюємо значення цільової функції підзадачі для провайдерів (підзадачі для ІТ-компанії).

5.4 Алгоритм розв'язання підзадач етапу 3

На цьому етапі використовується алгоритм, побудований на основі ідеї поступок. Попередньо впорядковуємо у порядку зростання співвідношення значень Q_j^A і Q_j^B цільової функції (2) розв'язку підзадач А, Б, отриманих для кожного провайдера. Варто нагадати, що значення Q_j^B цільової функції (2) для кожного провайдера ми отримуємо в результаті застосування ймовірнісно-жадібного алгоритму до підзадачі Б етапу 1. Значення Q_j^A цільової функції (2) для кожного провайдера ми отримуємо в результаті застосування алгоритму для підетапу 2.1.

Ідея основної частини алгоритму досить перспективна і проста. Для кожного провайдера у задачі підетапу 3.1 ми застосовуємо процедуру надання знижок, а у задачі підетапу 3.2 – процедуру відмови від знижок для кожного включеного у пакет сервісу. Тобто, поліпшення розв'язку можливе лише за умови поступки ІТ-компанії за надання сервісів і відповідно поступки провайдера, який погоджується на вищу ціну заради зростання сумарної вигоди провайдера і ІТ-компанії. Важливо, що пошук кращого розв'язку здійснюється по всім сервісам ІТ-компанії, які вона надає кожному провайдеру.

6 Результати експериментального дослідження

Проведене експериментальне дослідження запропонованого варіанту комбінованого методу продемонструвало працездатність розроблених алгоритмів і їх здатність надати ефективний розв'язок задачі формування пакетів сервісів. Розроблені алгоритми покладені в основу реалізації системи формування пакетів сервісів.

На рисунку 1 показано результати розв'язання задачі формування пакетів сервісів на прикладі довільно вибраних трьох провайдерів і 18 інфокомунікаційних сервісів за допомогою створеної інформаційної системи формування пакетів сервісів, в якій реалізований запропонований у доповіді варіант комбінованого методу.

На рисунку 2 наведено отримані в процесі застосування запропонованого у доповіді варіанту комбінованого методу значення загального доходу від надання сервісів, які підтверджують обґрунтованість формування пакетів сервісів для провайдерів. Дійсно, сервіси 1 – 3, 8, 10 – 13 та 17 забезпечують дохід. При цьому запропонований у доповіді метод дозволяє для кожного провайдера сформувати вигідний, як для провайдера, так і для ІТ-компанії, пакет сервісів, враховуючи обмеження: для провайдера 0 – сервіси 1, 8, 11, 12, 13, 17; для провайдера 1 – сервіси 3, 8, 10, 17; для провайдера 2 – сервіси 2, 3, 8, 13, 17.

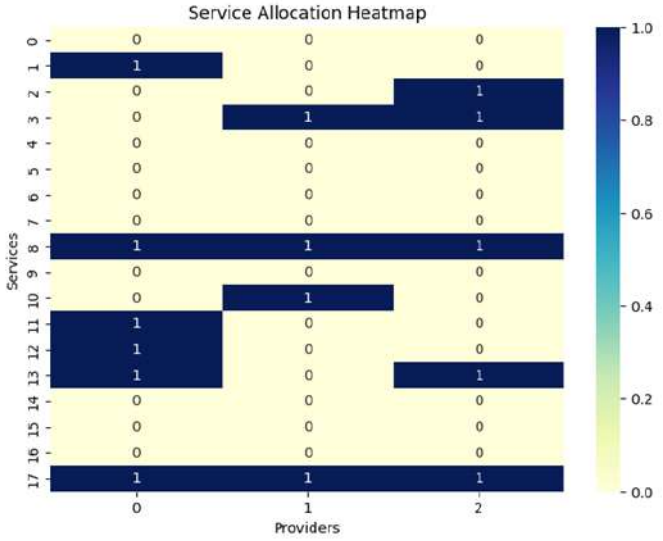

Рис. 1. Приклад розв'язку задачі формування пакетів сервісів

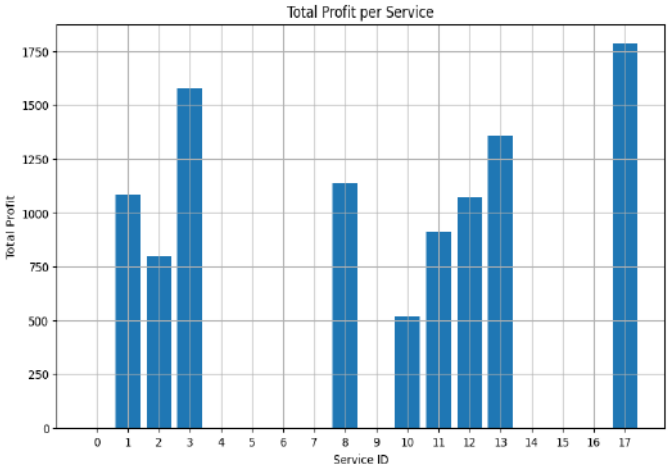

Рис 2. Обчислені значення загального доходу від надання сервісів

Інформаційна система формування пакетів сервісів становить зручний і корисний інструмент ведення бізнесової діяльності з надання інфокомунікаційних сервісів для провайдерів і ІТ-компаній.

Висновки

В рамках дослідження процесів життєвого циклу сервісів в інформаційних системах провайдерів інфокомунікацій сформульована задача формування пакетів сервісів. Поставивши за мету підвищення ефективності діяльності ІТ-компаній і провайдерів, автори запропонували метод формування пакетів сервісів, який враховує особливості цієї галузі діяльності в умовах сьогодення.

Для багатокритеріальної задачі математичного програмування запропонований варіант комбінованого методу, який інтегрує такі прийоми, як: а) виділення однокритеріальних підзадач для ІТ-компанії і провайдерів; б)

розв'язання однокритеріальних підзадач за допомогою спеціальних алгоритмів, які враховують їх специфічні особливості; в) визначення спільної частини у розв'язках підзадач для ІТ-компанії і провайдерів; г) використання спеціальних евристичних процедур для доповнення знайденої спільної частини до повного найкращого рішення початкової багатокритеріальної задачі з врахуванням системи знижок.

Для розв'язання однокритеріальних підзадач і отримання найвигідніших для ІТ-компанії пакетів сервісів та найвигіднішого пакету сервісів для кожного провайдера за умови виконання усіх обмежень (ресурсних, провайдера, міжсервісної залежності) запропоновані варіанти ймовірнісно-жадібного алгоритму.

Також запропоновані алгоритми для визначення спільної частини у розв'язках підзадач для ІТ-компанії і провайдерів та доповнення знайденої спільної частини до повного найкращого рішення початкової багатокритеріальної задачі з врахуванням системи знижок.

Виконане експериментальне дослідження показало працездатність і ефективність розробленого варіанту комбінованого методу розв'язання задачі формування пакетів сервісів. Цей метод покладений в основу створюваної системи формування пакетів сервісів.

Перспективним напрямом досліджень видається створення більш ефективних варіантів комбінованого методу розв'язання задачі формування пакетів сервісів.

Також вимагає додаткового дослідження проблема створення системи формування пакетів сервісів, в основу реалізації якої будуть покладені запропоновані у доповіді алгоритми та інші відомі ефективні алгоритми.

Література

1. Field, J. M. Service research priorities: designing sustainable service ecosystems / J. M. Field, D. Fotheringham, M. Subramony, A. Gustafsson, A. L. Ostrom, K. N. Lemon, J. R. McColl-Kennedy // Journal of Service Research. – 2021. – 24(4). – P. 462-479.

2. Favoretto C. From servitization to digital servitization: How digitalization transforms companies' transition towards services / C. Favoretto, G. H. Mendes, M. G. Oliveira, P. A. Cauchick-Miguel, W. Coreynen // Industrial Marketing Management. – 2022. – 102. – P. 104-121.

3. Serrano, J. An IT service management literature review: challenges, benefits, opportunities and implementation practices / J. Serrano, J. Faustino, D. Adriano, R. Pereira, M. M. da Silva // Information. – 2021. – 12(3). – 111.

4. Richter, H. IT-service value modeling: a systematic literature analysis / H. Richter, B. Lantow // International Conference on Business Information Systems (2021, June). – Cham: Springer International Publishing. P. 267-278.

5. Mora, M. Agile IT Service Management Frameworks and Standards: A Review / M. Mora, J. Marx-Gomez, F. Wang, O. Diaz // Arabnia, H.R., Deligiannidis, L., Tinetti, F.G., Tran, QN. (eds) Advances in Software Engineering, Education, and e-Learning. Transactions on Computational Science and Computational Intelligence. Springer, Cham. – 2021. – P.

6. Agutter C. (2020). ITIL Foundation Essentials ITIL 4 Edition-The ultimate revision guide / C. Agutter. – IT Governance Publishing Ltd., 2020.

7. Maddern, H. End-to-end process management: implications for theory and practice / H. Maddern, P. A. Smart, R. S. Maull, S. Childe // Production Planning & Control. – 2014. – 25(16). – P. 1303-1321.

8. Чимшир В. Платформа підтримки життєвого циклу сервісів в інформаційних системах провайдерів інформаційно-комунікаційних послуг / В. Чимшир, С. Теленик, О. Ролік, Е. Жаріков // Адаптивні системи автоматичного управління. – Том 1.– № 42 DOI: https://doi.org/10.20535/1560-8956.42.2023.279172.

9. MacLean D. Implementation and impacts of IT Service Management in the IT function / D. MacLean, R. Titah // International Journal of Information Management. – 2021. – 70. v102628. – P. -.

10. Гавриленко О. Вирішення задачі впровадження пакетів сервісів за допомогою статистичної інформації / О. Гавриленко, В. Чимшир, Е. Жаріков, С. Теленик, Р. Омельченко // Адаптивні системи автоматичного управління. – 2023. – Том 2. – № 43. С.

DOI: https://doi.org/10.20535/1560-8956.43.2023.292257

11. Гавриленко О. Метод формування пакетів сервісів за допомогою алгоритмів кластеризації / О. Гавриленко, В. Чимшир, Е. Жаріков, С. Теленик, О.Амонс // Адаптивні системи автоматичного управління. – 2024. – Том 1. – № 44. – С.182-191. DOI: https://doi.org/10.20535/1560-8956.44.2024.302437

12. Chymshyr V. Models and methods for forming service packages for solving of the problem of designing services in information systems of providers / V. Chymshyr, O. Zhdanova, O. Havrylenko, G. Nowakowski, S. Telenyk // Inf. Comput. and Intell. syst. – 2024.– no. 5. – pp. 29–54. DOI: https://doi.org/10.20535/2786-8729.5.2024.316432

Technology for Information System Component Interaction Using Reactive Signal Databases

Danylo Vitkovskyi, Serhii Telenyk
Information Systems and Technologies Department
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
vitkovskiy.danylo@lll.kpi.ua, s.telenyk@gmail.com

Abstract. **This paper presents the technology for interaction of information system components based on a database of structured data, changes in which are monitored asynchronously in real time. The purpose of this technology is to simplify the interaction between information system components. This technology is especially relevant for systems whose data is hierarchical. Experimental testing has shown the high potential of this technology in the case of implementing a full-fledged DBMS based on it.**

Keywords: asynchronous communication, database management system, information system, microservice architecture, reactive system, technology.

Технологія взаємодії компонентів інформаційних систем на основі бази даних реактивних сигналів

Вітковський Данило Олександрович, Теленик Сергій Федорович
Кафедра інформаційних систем та технологій
НТУУ «КПІ ім. Ігоря Сікорського»
Київ, Україна
vitkovskiy.danylo@lll.kpi.ua, s.telenyk@gmail.com

Анотація. **У даній роботі розглядається технологія взаємодії компонентів інформаційних систем на основі бази структурованих даних, зміни яких відслідковуються асинхронно, у режимі реального часу. Метою даної технології є спрощення взаємодії між компонентами інформаційної системи. Ця технологія є особливо актуальною для систем, природа даних яких є ієрархічною. Експериментальні дослідження показали високий потенціал даної технології у разі реалізації повноцінної СКБД на її основі.**

Ключові слова: асинхронна комунікація, мікросервісна архітектура, реактивна система, система керування базою даних, технологія, інформаційна система.

Вступ

Важливим аспектом будь-якої інформаційної системи є комунікація між її компонентами та/або клієнтами. Обраний спосіб комунікації може вплинути як на зручність його використання, так і на надійність системи – у разі похибки в реалізації обраного способу комунікації, система може не працювати взагалі або ж працювати ненадійно.

Так, якщо система з мікросервісною архітектурою має підтримувати транзакційні операції, то має існувати легкий спосіб скасування операції коли вона ще не завершена. У такому випадку, не розумно використовувати для комунікації підходи виду RPC (англ. Remote Procedure Call, віддалений виклик процедури), на кшталт REST API, оскільки це вимагатиме аби кожний з мікросервісів, здатних до скасування операції, знав про кожний інший мікросервіс, якому потрібно повідомити про скасування. Це створює квадратичну залежність між мікросервісами, що ускладнює їхню реалізацію та процес оновлення/додавання нових мікросервісів [1].

У разі ж використання асинхронної комунікації, наприклад, через систему повідомлень, мікросервіси можуть не знати один про одного, а лише про загальний механізм обробки подій. Таким чином, залежність кількості зв'язків від кількості мікросервісів стає лінійною, а додавання нових мікросервісів не вимагає зміни коду тих мікросервісів, що вже існують.

У даній роботі розглядається технологія взаємодії компонентів інформаційних систем на основі бази структурованих даних, зміни яких відслідковуються асинхронно, у режимі реального часу. Метою даної технології є спрощення взаємодії між компонентами інформаційної системи.

Принцип роботи

Структури даних, з якими іде робота у коді системи, мають, в основному, ієрархічну структуру. Якщо кілька компонентів системи мають спільні дані, то їх прийнято зберігати у базі даних, як єдиному джерелі правди. Ідея технології, що розглядається, полягає у тому, щоб спростити взаємодію між компонентами системи, шляхом автоматичного повідомлення про зміни у даних, що зберігаються у базі даних, й автоматичного оновлення локальних копій цих даних у компонентах системи.

На зміни вузлів структури даних у базі можливо виконати підписання. Повідомлення про ці зміни розповсюджуються вздовж ієрархії: якщо зміна деструктивна (деякі з елементів ієрархії видаляються), то про неї повідомляються всі як батьківські, так і дочірні вузли, інакше – лише батьківські.

Комунікація між клієнтом та СКБД[1] відбувається за протоколом WebSocket, що дозволяє реалізувати асинхронну комунікацію та отримання повідомлень про зміни у реальному часі.

Дані, що передаються, закодовані у двійковому форматі, що дозволяє зменшити об’єм переданих даних, а також зменшити час на їхнє кодування/декодування. Це дозволяє зменшити навантаження на мережу, а також зменшити затримки при передачі даних.

Результати експериментальних досліджень

Робота запропонованої технології перевірялась шляхом реалізації прототипу СКБД та клієнтської бібліотеки з використанням мови програмування Rust [2], з подальшим заміром швидкості виконання операцій. Проводилося порівняння зі швидкістю виконання еквівалентних операцій СКБД MongoDB [3] з використанням Change Streams [4].

Вимірювання проводились на комп’ютері з наступними характеристиками:

- процесор: «AMD Ryzen 5 5600H with Radeon Graphics × 6»;
- оперативна пам’ять: 32 ГБ;
- ОС: Linux Mint 22.1 Cinnamon;
- Docker версії 28.0.4;
- MongoDB версії 8.0.8;
- Rust версії 1.86.0.

Як можна побачити з графіків, що наведені нижче, швидкість виконання операцій прототипу запропонованої технології є помітно вищою, ніж у MongoDB, що свідчить про її потенціал у разі реалізації повноцінної СКБД.

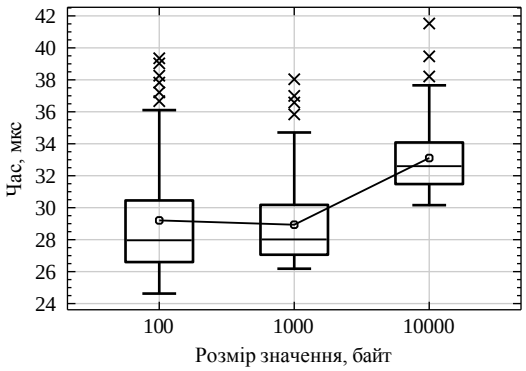

Рис. 1. Час виконання операцій зчитування з бази даних прототипу (`get untracked`)

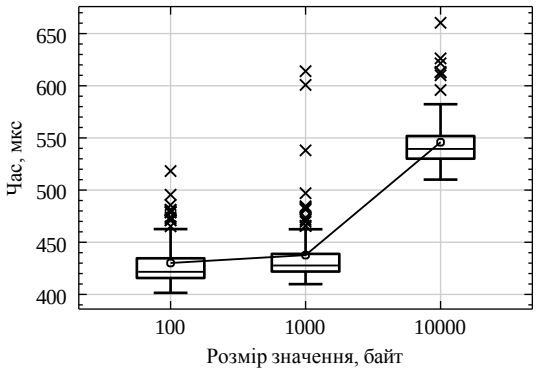

Рис. 2. Час виконання операцій зчитування з бази даних MongoDB (`find one`)

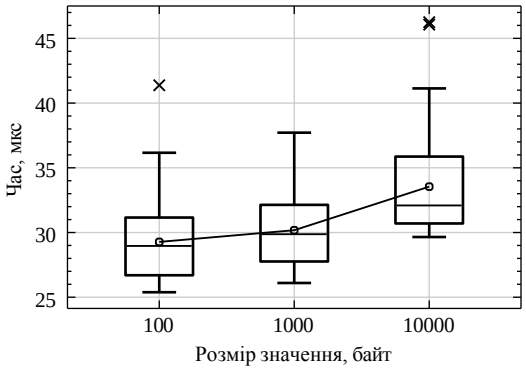

Рис. 3. Час виконання операцій запису до бази даних прототипу (`set`)

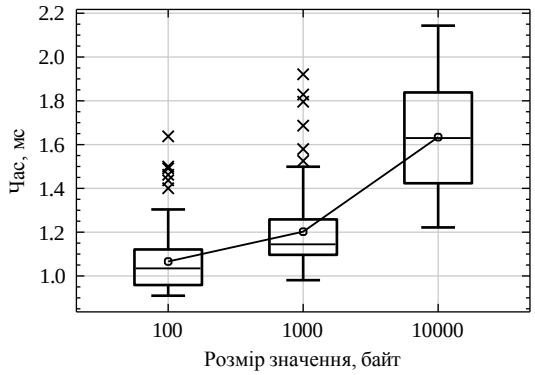

Рис. 4. Час виконання операцій запису до бази даних MongoDB (`insert one`)

[1]Система керування базою даних.

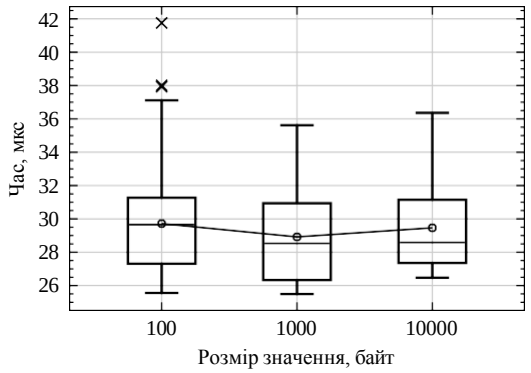

Рис. 5. Час виконання операцій видалення з бази даних прототипу (`delete`)

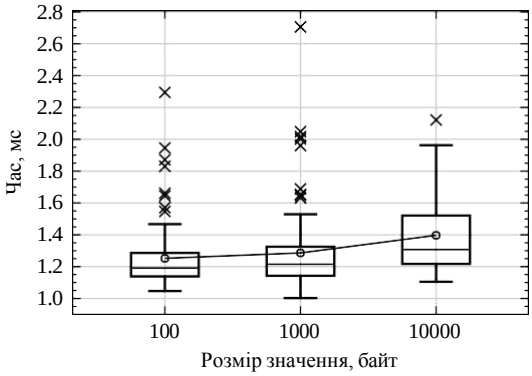

Рис. 6. Час виконання операцій видалення з бази даних MongoDB (`delete one`)

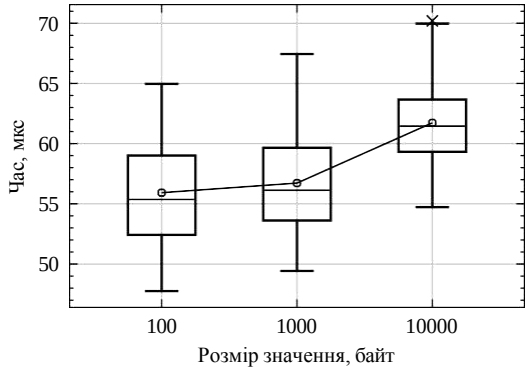

Рис. 7. Час виконання операцій зчитування з підписанням на зміни з бази даних прототипу (`get`),

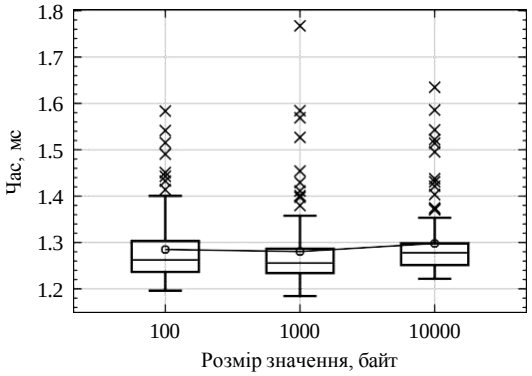

Рис. 8. Час виконання операції підписання на зміни у базі даних MongoDB (`watch`)

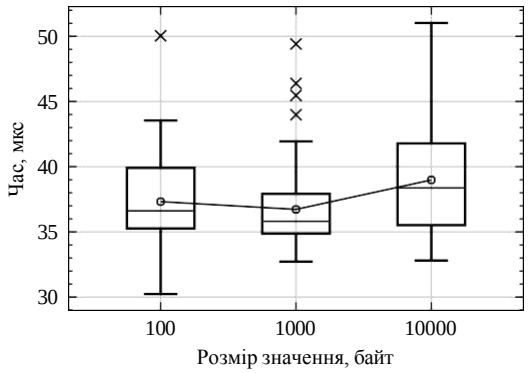

Рис. 9. Час виконання операцій запису (за наявності підписників на зміни) до бази даних прототипу (`set`),

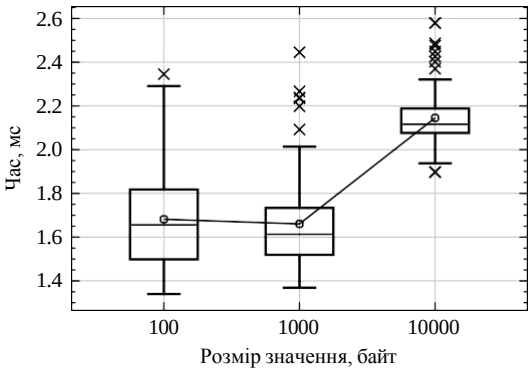

Рис. 10. Час виконання операції оновлення даних у базі даних MongoDB (`update`)

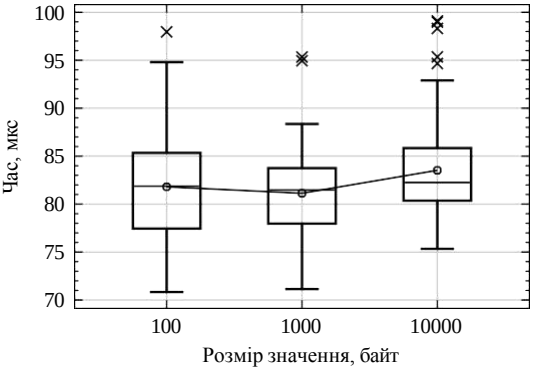

Рис. 11. Час очікування на повідомлення про зміни у базі даних прототипу (`notification`)

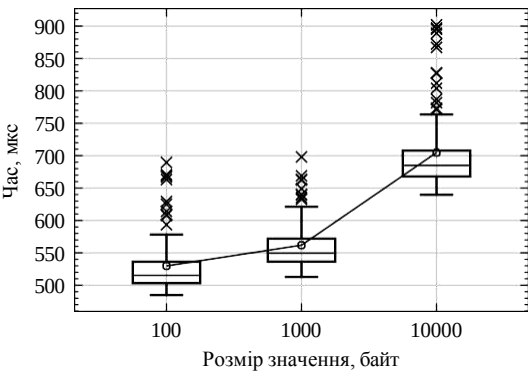

Рис. 12. Час очікування на наступну зміну у базі даних MongoDB (`next change`)

Варто зазначити, що під час замірів, TCP з'єднання прототипу СКБД на основі запропонованої технології працювало зі встановленим прапорцем «`TCP_NODELAY`», що

дозволяє вимкнути алгоритм Нагла (англ. Nagle's algorithm) [5]. Цей алгоритм дозволяє зменшити кількість пакетів, що передаються мережею, однак, у разі частих запитів, як от під час вимірювання, він може призвести до затримок у передачі даних, оскільки певний час очікує на заповнення пакету або ж на підтвердження отримання пакету від приймача.

Опис кодування даних

Під час розробки прототипу було використано двійкове кодування даних що передаються мережею.

Для опису байтового представлення даних на рисунках нижче, використано граматики на основі нотацій РБНФ[2] [6,7].

Форму граматики було додатково розширено, аби дозволити посилання на попередньо визначені терми у поточному правилі: префікс «%» перед ім'ям терму (обов'язково виділяти ім'я за допомогою «<» та «>») вказує на те, що значення, до якого його було розгорнуто під час розбору тексту, використовується у подальшому розборі як літерал або число повторень. Посилатися можливо лише на попередні терми у тій же або батьківській групі термів.

Таким чином, граматика виду

```
A = B %<B>
B = "foo" / "bar"
C = n %<n><OCTET>
n = %d1 / %d2
```
(1)

означає те саме, що граматика

```
A = ("foo" "foo") / ("bar" "bar")
C = (%d1 OCTET) / (%d2 OCTET OCTET)
```
(2)

але дозволяє потенційно більш компактний запис.

```
сигнал = %d0        ; ніщо                                    abnf
       / %d1        ; тригер
       / %d2 BIT    ; булеве значення
       / %d3 8OCTET ; знакове ціле число
       / %d4 8OCTET ; дійсне число
       / %d5 байти
       / %d6 рядок  ; UTF-8 байти
       / %d7 список
       / %d8 об'єкт ; пари ім'я-значення
байти = довжина %<довжина>OCTET
рядок = довжина ; кількість байтів
        *символ ; сумарна кількість байтів відповідає довжині
список = довжина %<довжина><сигнал>
об'єкт = довжина %<довжина><пара>
пара = байти сигнал

довжина = 4OCTET ; беззнакове ціле число
символ = <одно-байтовий символ>
       / <дво-байтовий символ>
       / <три-байтовий символ>
       / <чотири-байтовий символ>
<одно-байтовий символ> = %x00-7F
<дво-байтовий символ> = %xC0-DF %x80-BF
<три-байтовий символ> = ( %xE0 %xA0-BF %x80-BF )
                      / ( %xE1-EC %x80-BF %x80-BF )
                      / ( %xED %x80-9F %x80-BF )
<чотири-байтовий символ> = ( %xF0 %x90-BF %x80-BF %x8-BF )
                           / ( %xF1-F3 %x80-BF %x80-BF %x80-BF )
                           / ( %xF4 %x80-8F %x80-BF %x80-BF )
```

Рис. 13. Байтове представлення даних

```
шлях = корінь решта                                           abnf
корінь = байти
решта = <кількість частин> %<кількість частин><частина>
частина = ( %d0 індекс ) / ( %d1 ключ )
індекс = u32
ключ = байти
<кількість частин> = u32
u32 = 4OCTET ; беззнакове ціле число
```

Рис. 14. Граматика двійкового представлення шляху

```
<пакет від клієнта> =  %d0 <VAL-ID> BIT шлях           ; get        abnf
                     /  %d1 <ACK-ID> шлях сигнал       ; set
                     /  %d2 <VAL-ID> BIT шлях сигнал   ; fetch-set
                     /  %d3 <VAL-ID> BIT шлях сигнал   ; get-or-set
                     /  %d4 <ACK-ID> шлях              ; trigger
                     /  %d5 <ACK-ID> шлях індекс сигнал ; insert
                     /  %d6 <ACK-ID> шлях сигнал       ; push back
                     /  %d7 <ACK-ID> шлях сигнал       ; push front
                     /  %d8 <ACK-ID> шлях              ; delete
                     /  %d9 <VAL-ID> шлях              ; fetch-delete
                     / %d10 <ACK-ID> шлях              ; pop-back
                     / %d11 <VAL-ID> шлях              ; pop-back-fetch
                     / %d12 <ACK-ID> шлях              ; pop-front
                     / %d13 <VAL-ID> шлях              ; pop-front-fetch
                     / %d14 <ACK-ID> <SUB-ID> шлях     ; unsubscribe
<VAL-ID> = u64 ; ID запиту, що цікавиться значенням
<ACK-ID> = u64 ; ID запиту, що очікує підтвердження
u64 = 8OCTET
```

Рис. 15. Граматика байтового представлення пакетів від клієнта

[2]Розширена Бакус-Наур форма; англ. ABNF, augmented Bakus-Naur form.

```
<пакет від СКБД> = %d0 відповідь                                    abnf
                 / %d1 оповіщення

відповідь = %d0 <ACK-ID>                      ; acknowledge
          / %d1 <VAL-ID> (%d0 / %d1 сигнал)   ; maybe value
          / %d2 <ERR-ID> <помилка обходу>     ; error.
                                              ; Конкретне представлення
                                              ; помилки обходу
                                              ; залежить від деталізації
                                              ; реалізації
<ERR-ID> = %d0 <ACK-ID>
         / %d1 <VAL-ID>

оповіщення = %d0 шлях                ; notification
           / %d1 шлях сигнал         ; set
           / %d2 шлях індекс сигнал  ; insert
           / %d3 шлях сигнал         ; push back
           / %d4 шлях сигнал         ; push front
           / %d5 шлях                ; delete
           / %d6 шлях                ; pop back
           / %d7 шлях                ; pop front
```

Рис. 16. Граматика байтового представлення пакетів від СКБД

```
<помилка обходу> = %d0                          ; порожній шлях       abnf
                 / %d1 шлях індекс              ; індекс поза межами
                 / %d2 <пройдений шлях> ключ    ; нема значення за шляхом
                 / %d3 <пройдений шлях> ключ                ; не об'єкт
                 / %d4 <пройдений шлях> (%d0 / %d1 індекс)  ; не список
<пройдений шлях> = %d0      ; порожній
                 / %d1 шлях ; не порожній
```

Рис. 17. Граматика байтового представлення помилок обходу

Висновки

У результаті проведеного дослідження було розроблено технологію, що дозволяє полегшити комунікацію між компонентами інформаційних систем та/або їх клієнтами шляхом автоматичного відстеження змін у структурованих даних, що зберігаються у базі даних.

Це особливо актуально актуально для роботи з даними, що за своєю природою мають ієрархічний характер (наприклад, конфігурації), та для роботи з системами, що мають велику кількість компонентів, які мають спільні дані, що потребують синхронізації.

У результаті проведення експериментальних досліджень було показано високу швидкість роботи та потенціал використання пропонованої технології у разі реалізації повноцінної СКБД.

Література

1. Вітковський Д., Теленик С. База даних публічних реактивних функціональних сигналів // Адаптивні системи автоматичного управління. КПІ ім. Ігоря Сікорського, 2024. вип. 2, № 45. с. 204–221.

2. Klabnik S., Nichols C., Krycho C. Rust Programming Language [online]. URL: https://www.rust-lang.org/ (дата звернення: 09.05.2025).

3. MongoDB Inc. MongoDB [online]. 2025. URL: https://www.mongodb.com/ (дата звернення: 16.04.2025).

4. MongoDB Inc. Change Streams - Database Manual v8.0 - MongoDB Docs [online]. URL: https://www.mongodb.com/docs/manual/changeStreams/ (дата звернення: 27.03.2025).

5. Nagle J. Congestion Control in IP/TCP Internetworks [online]. RFC Editor, 1984. № 896. 9 с. URL: https://doi.org/10.17487/RFC0896.

6. Crocker D., Overell P. Augmented BNF for Syntax Specifications: ABNF [online]. RFC Editor, 2008. № 5234. 16 с. URL: https://doi.org/10.17487/RFC5234.

7. Kyzivat P. Case-Sensitive String Support in ABNF [online]. RFC Editor, 2014. № 7405. 4 с. URL: https://doi.org/10.17487/RFC7405.

A generalized model of the gaze estimation process for web applications

Stepan Korol, Oliinyk Volodymyr
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
korolst43@gmail.com, oliinyk.volodymyr@gmail.com

Abstract. **This paper presents a generalized model for real-time gaze estimation in web applications using an RGB camera. The model is adaptable to various methods of facial landmark detection and gaze direction estimation. The structure of the model and its mathematical formulation are provided, offering a foundation for further research and development in the field of human–computer interaction.**

Keywords**:** ***gaze estimation, web applications, generalized model, gaze direction, image normalization.***

Узагальнена модель процесу відстеження погляду користувача вебзастосунків

Король Степан, Олійник Володимир
КПІ імені Ігоря Сікорського
м. Київ, Україна
korolst43@gmail.com, oliinyk.volodymyr@gmail.com

Анотація. У роботі представлено узагальнену модель відстеження погляду користувача вебзастосунків із використанням RGB-камери в реальному часі. Модель є адаптивною до різних методів детектування ключових точок обличчя та визначення напрямку погляду. Наведено структуру моделі, а також її математичне представлення, що може бути основою для подальших досліджень та розробок у сфері інтерфейсів людина-комп'ютер.

Ключові слова: відстеження погляду, вебзастосунки, узагальнена модель, напрямок погляду, нормалізація зображення.

Вступ

Інтерактивні вебзастосунки все частіше використовуються не лише для презентації контенту, але і для активної взаємодії з користувачем у реальному часі. У цьому контексті особливу актуальність набуває відстеження погляду користувача як засіб безконтактного управління, персоналізації інтерфейсу, моніторингу уваги та збору аналітики поведінки. На відміну від десктопних або апаратно залежних рішень, веборієнтовані рішення мають враховувати специфіку браузера, необхідність роботи з єдиним RGB-відеопотоком, а також вимоги до продуктивності та захисту персональних даних [1]. Існуючі реалізації не мають чіткої структурної уніфікації, що ускладнює їх інтеграцію, оцінку та вдосконалення у межах вебплатформ.

Саме тому, запропоновано узагальнену модель процесу відстеження погляду користувача у вебзастосунках, яка структурує основні етапи роботи таких систем. Модель охоплює основні методи визначення напрямку погляду та детектування ключових точок обличчя, забезпечує модульність та адаптованість до обмежень браузерного середовища. Вона може слугувати основою для побудови ефективних і масштабованих вебрішень у галузях освіти, охорони здоров'я, цифрового маркетингу та психології. Її концептуальна універсальність створює підґрунтя для подальшого розвитку персоналізованих та етично безпечних засобів відстеження погляду у браузерному середовищі.

Структура узагальненої моделі

У межах проведеного дослідження розроблено узагальнену модель процесу відстеження погляду користувача вебзастосунків, орієнтовану на застосування у вебсередовищі. Запропонована структура, представлена на рис.1, відображає основні етапи обробки зображення, які є спільними для більшості сучасних систем, незалежно від конкретного алгоритмічного або архітектурного рішення. Початковим етапом є захоплення відеопотоку з вбудованої у пристрій або зовнішньої RGB-камери, що слугує єдиним джерелом інформації у браузерному середовищі. Це дозволяє уникнути залежності від інфрачервоних камер чи апаратних трекерів погляду та забезпечити

масштабованість рішень на основі запропонованої системи.

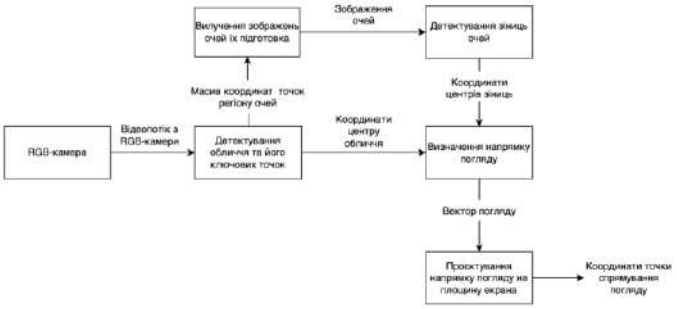

Рис. 1. Структурна схема узагальненої моделі відстеження погляду користувача вебзастосунків

На наступному кроці здійснюється виявлення обличчя користувача та визначення координат ключових точок обличчя, таких як очі, ніс, вуха. Ці анатомічно значущі орієнтири слугують базою для просторової прив'язки моделі визначення напрямку погляду до обличчя та забезпечують стабільну ідентифікацію його регіонів незалежно від повороту голови чи незначних змін виразу обличчя. Виявлені ключові точки дають змогу локалізувати зону очей – як основну область інтересу та оцінити положення голови у просторі шляхом побудови матриці повороту і вектора паралельного перенесення відносно камери. Таким чином, здійснюється первинна нормалізація координат і приведення зображення до стандартизованої системи, узгодженої з внутрішніми параметрами камери.

З метою зменшення варіативності вхідних даних, обрізані зображення очей нормалізуються з урахуванням положення голови, що дозволяє привести усі зразки до єдиної системи координат. Це дає змогу знизити вплив зовнішніх чинників, зокрема кута зйомки, освітлення, розміру або положення обличчя в кадрі. Нормалізовані зображення мають фіксовану просторову орієнтацію й масштаб, що забезпечує узагальнюваність моделі та підвищує її стійкість при застосуванні в реальних умовах.

На основі вирівняного зображення формується вхід до блоку визначення напрямку погляду, в основі якого у більшості випадків застосовується нейронна модель, яка реалізує регресійне перетворення з простору зображень у простір орієнтацій вектора погляду. Такі моделі зазвичай представлені згортковими нейронними мережами або гібридними архітектурами, які поєднують витягування ознак із просторовим контекстним моделюванням (наприклад, трансформерами чи позиційно-залежними регресійними шарами). Отриманий напрямок, зазвичай у вигляді кутів нахилу та повороту, описує орієнтацію погляду користувача у координатній системі нормалізованої камери. Цей напрямок потребує подальшого перетворення у координати площини екрана, де відбувається фактична взаємодія користувача з вебінтерфейсом.

Для адаптації до індивідуальних особливостей користувача впроваджується процедура калібрування, у межах якої модель навчається зіставляти визначений вектор погляду з реальною точкою фіксації на екрані. Такий підхід компенсує анатомічні відмінності між користувачами, помилки нормалізації, а також невраховані параметри оптики камери. Як правило, для цього користувачеві пропонується пройти калібрувальну сесію у вигляді послідовного фіксування погляду на серії контрольних точок, що розташовані на площині екрана. На основі зібраної відповідності між вектором погляду та фактичною позицією на екрані застосовуються методи лінійної або поліноміальної регресії, які дозволяють побудувати індивідуальну модель проєкції для кожного користувача.

Завершальним етапом є відображення обчисленої точки спрямування погляду у координатах вікна вебзастосунку.

Математичне представлення узагальненої моделі

У запропонованій моделі відстеження погляду процес починається із захоплення поточного RGB-кадру. На цьому зображенні детектор обличчя виділяє прямокутну область разом з множиною двовимірних опорних точок. Використовуючи підмножину ключових точок, що відповідає області очей, формується пара регіонів, з якої вирізаються локальні патчі, що нормалізуються до фіксованого розміру та діапазону інтенсивностей, після чого надходять до детектора зіниць. Далі для кожного ока обчислюється координата центру зіниці у локальній системі відповідного патча. Тоді нормалізований напрямок погляду для одного ока задається рівнянням:

$$\hat{g} = \frac{K^{-1}(u, v, 1)^{\top} - o}{\| K^{-1}(u, v, 1)^{\top} - o \|_2}$$

де $\hat{g}$ – одиничний 3-вимірний вектор напрямку погляду для відповідного ока;

K – матриця внутрішніх параметрів RGB-камери (фокусні відстані й координати головної точки);

(u,v) – 2-D координати центру зіниці у пікселях локального патча ока;

o – 3-D координати геометричного центру ока.

Зведений вектор погляду g отримують усередненням двох одержаних напрямків і подальшою нормалізацією:

$$g = \frac{\hat{g}_l + \hat{g}_r}{\| \hat{g}_l + \hat{g}_r \|_2}$$

де g – єдиний нормалізований напрямок погляду користувача;

$\hat{g}_l$, $\hat{g}_r$ – одиничні вектори погляду лівого й правого ока відповідно. Такий підхід дозволяє не лише згладити похибки, пов'язані з неточністю локалізації зіниці, але й врахувати бінокулярний ефект, коли обидва ока доповнюють одне одного в оцінці напрямку.

Після цього розглядається промінь, що виходить із вибраної стартової точки o (зазвичай це середина між центрами очей) у напрямку g. Екран моделюється як площина з одиничною нормаллю n та зсувом d, параметри яких один раз оцінюють під час первинної грубої калібровки. Точка перетину променя з цією площиною отримується наступним чином:

$$p = o + \left(- \frac{n^{\top} o + d}{n^{\top} g} \right) g$$

де o – вибрана стартова точка променя (середина між очима);

g – вже нормалізований зведений напрямок погляду;

n, d – нормаль та зсув площини, якою моделюється екран;

p – просторова точка на поверхні екрана, куди спрямований промінь.

Слід зазначити, що така модель є геометричною апроксимацією реального процесу фіксації погляду, і, попри свою концептуальну ясність, вона не враховує всіх

джерел похибки, таких як невідповідність реальної точки фокусування і оптичної осі ока (ефект паралакса), а також спотворення, пов'язані з індивідуальними особливостями анатомії.

Оскільки внутрішні параметри камери, а також орієнтація екрана не забезпечують ідеальної точності, після знаходження *p* використовується коротке калібрування на 5 чи 9 точок. Під час якого, на основі пар значень просторових точок та екранної мітки, навчається регресійна модель, що виконує перетворення тривимірної координати перетину променя з площиною у двовимірні координати екрана. Найчастіше для цього застосовують методи лінійної регресії (наприклад, з регуляризацією) або поліноміального наближення другого порядку, які демонструють стабільну поведінку при невеликій кількості калібрувальних точок.

Методологічні підходи до визначення напрямку погляду

У межах узагальненої моделі процесу відстеження погляду користувача вебзастосунків виділяються кілька основних підходів до визначення напрямку погляду. Найбільш поширеними серед них є методи на основі візуальних характеристик [2], що передбачають використання фрагментів зображення – зокрема, області очей або обличчя як вхідних даних для машинного алгоритму, що виконує регресію напрямку погляду. Переважно реалізуються за допомогою згорткових нейронних мереж, які здатні автоматично навчатися релевантним ознакам без необхідності ручного проектування ознакового простору. Такі методи є достатньо гнучкими та здатними до адаптації під нові умови, однак потребують великої кількості анотованих даних і можуть демонструвати зниження точності при значному відхиленні голови або нестандартному освітленні.

Методи на основі ключових точок використовують координати попередньо визначених ландмарків обличчя (наприклад, центри зіниць, кути очей, положення носа та брів), які отримуються за допомогою моделей детекції обличчя. Вони дозволяють будувати відносні просторові ознаки, що менш чутливі до варіацій зовнішнього вигляду та фону. Цей підхід забезпечує високу обчислювальну ефективність та є зручним у випадках, коли зображення має низьку роздільність або містить шуми [3].

Методи на основі геометричної моделі ока [4] базуються на відтворенні тривимірної структури очного яблука та його взаємодії з оптичною системою камери. У рамках таких методів вектор погляду обчислюється як промінь, що виходить з центру очного яблука крізь зіницю і перетинає площину екрана. Для цього необхідна наявність або апроксимація параметрів внутрішньої калібровки камери, що в умовах вебзастосунків часто є недоступним. Незважаючи на це, геометричні методи залишаються фундаментальними в теоретичних моделях та використовуються для побудови високоточних систем у контрольованих умовах.

Гібридні підходи [5] комбінують елементи кількох наведених вище класів. Наприклад, координати ключових точок можуть використовуватись для вирівнювання або нормалізації області очей, яка далі подається до моделі на основі зображення. Такий підхід дозволяє поєднати геометричну стабільність з гнучкістю навчальних методів, що є особливо важливим для роботи у браузерному середовищі з його обмеженими ресурсами та великою варіативністю умов використання.

Висновки

У межах даної роботи сформовано узагальнену модель процесу відстеження погляду користувача, спеціально адаптовану до особливостей функціонування вебзастосунків. Структура моделі дозволяє використовувати різні підходи до визначення напрямку погляду, зокрема на основі зображень, ключових точок та геометричної моделі ока, що забезпечує її гнучкість і сумісність із сучасними алгоритмами та технологіями. Особливістю моделі є її модульність, що дозволяє легко поєднувати або замінювати окремі компоненти в залежності від обраного технічного рішення чи обмежень середовища. У ході розробки моделі було приділено увагу її адаптивності до обмежень браузерного середовища, таких як обмежена обчислювальна потужність, відсутність доступу до апаратної калібровки та потреба в захисті персональних даних.

Таким чином, розроблена модель може слугувати уніфікованою основою для побудови веборієнтованих систем відстеження погляду, що працюють у режимі реального часу. Її застосування дозволяє не лише стандартизувати процес реалізації, а й забезпечити порівнянність результатів між різними реалізаціями в дослідницьких та прикладних контекстах. Представлені результати формують основу для подальших досліджень і вдосконалення методів відстеження погляду користувачів вебзастосунків.

Література

[1] Ansari M., Kasprowski P., Obetkal M. Gaze tracking using an unmodified web camera and convolutional neural network // Applied Sciences. 2021. Vol. 11.

[2] Cheng Y. et al. Appearance-based gaze estimation with deep learning: a review and benchmark // State Key Laboratory of Virtual Reality Technology and Systems. 2024.

[3] Oliinyk V. An efficient face mask detection model for real-time applications // Adaptive systems of automatic control. 2022. Vol. 1, №40. - PP 54-64.

[4] Kaur H., Jindal S., Manduchi R. Rethinking model-based gaze estimation // Association for Computing Machinery. 2022. Vol. 5.

[5] Krafka K., Khosla A., Kellnhofer P. Eye tracking for everyone // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016.

Improvement of Pearson Correlation to Overcome the Matrix Sparsity Problem in Recommender SystemS

Sherstuik Ihor, Holubiev Leontii
Igor Sikorsky Kyiv Polytechnic Institute
Kiev, Ukraine
igor.sherstuk8229@gmail.com, golubev.leontiy@lll.kpi.ua

Abstract. **The article examines the issue of reduced recommendation accuracy under conditions of rating matrix sparsity. An improved approach to Pearson correlation is proposed by introducing a regularization factor that accounts for the number of common ratings between users. Experimental results demonstrate the effectiveness of the method in cases with 30–55% missing values. The proposed solution leads to improved accuracy metrics and reduced prediction errors.**

Keywords*: *ecommender system, Pearson correlation, regularization, matrix sparsity, prediction accuracy.

Удосконалення кореляції Пірсона для подолання проблеми розрідженності матриці в рекомендаційних системах

Шерстюк Ігор, Голубєв Леонтій
КПІ імені Ігоря Сікорського
Київ, Україна
igor.sherstuk8229@gmail.com, golubev.leontiy@lll.kpi.ua

***Анотація*. У статті розглянуто проблему зниження точності рекомендацій в умовах розрідженості матриці. Запропоновано вдосконалення методу кореляції Пірсона шляхом введення регуляризаційного множника, що враховує кількість спільних оцінок. Експериментально доведено ефективність підходу при 30–55% відсутніх значень. Метод забезпечує покращення метрик точності та зменшення похибок.**

Ключові слова: рекомендаційна система, кореляція Пірсона, регуляризація, розрідженість матриці, точність рекомендацій.

Вступ

У сучасних рекомендаційних системах колаборативна фільтрація є одним з найефективніших і найпоширеніших методів персоналізації. Проте однією з її основних проблем залишається розрідженість матриці оцінок, яка призводить до погіршення якості рекомендацій [1]. Зокрема, класичний підхід на основі кореляції Пірсона дає нестабільні або хибні результати при недостатній кількості спільно оцінених

об'єктів [2]. Це створює потребу в удосконаленні методу з урахуванням обмеженості даних.Пірсона.

Математичні моделі

У колаборативній фільтрації персоналізовані рекомендації формуються на основі схожості між користувачами. Одним з базових підходів є використання коефіцієнта кореляції Пірсона, який оцінює ступінь подібності між користувачами на основі спільно оцінених об'єктів [1].
Класична формула має вигляд:

$$sim(u,v) = \frac{\sum_{i \in I_{uv}} (r_{u,i} - \overline{r_u})(r_{v,i} - \overline{r_v})}{\sqrt{\sum_{i \in I_{uv}} (r_{u,i} - \overline{r_u})^2} \cdot \sqrt{\sum_{i \in I_{uv}} (r_{v,i} - \overline{r_v})^2}} \quad (1)$$

де $r_{u,i}$ – оцінка користувача u для об'єкта i;

$\overline{r_u}$ – середнє значення оцінок користувача u;

I_{uv} – множина об'єктів, оцінених обома користувачами u та v.

Цей метод працює добре, коли є достатньо даних, однак при високій розрідженості матриці оцінок виникає серйозна проблема: навіть незначна кількість спільних оцінок може призводити до високої, але хибної схожості [3]. У таких умовах класична формула стає ненадійною.

Запропонована математична модель дозволяє сформувати більш стійкі та надійні рекомендації у випадках, коли інформація про користувачів є частково відсутньою, а матриця оцінок має високий ступінь розрідженості.

Результати експериментальних досліджень

У межах дослідження було проведено серію експериментів із моделюванням різних рівнів розрідженості даних. Матриця оцінок складалася з 50 користувачів і 20 об'єктів, де поступово збільшувався відсоток відсутніх значень — від 10% до 55%. Основна увага приділялася оцінці якості рекомендацій у випадках, коли частка пропущених оцінок перевищувала 30%, оскільки саме в цих умовах класичний метод Пірсона демонструє найменшу стабільність [3].

На рис. 1. чітко спостерігається тенденція: зі зростанням рівня розрідженості показники MAE і RMSE для класичного підходу зростають значно швидше, ніж для регуляризованої версії. Найбільш виражена різниця фіксується після позначки у 40% пропущених оцінок. У цьому діапазоні регуляризований метод утримує більш плавне зростання похибки, тоді як класичний втрачає точність різко і нерівномірно.

Щоб зменшити вплив подібних похибок, у роботі запропоновано вдосконалити цей підхід шляхом введення регуляризації, яка враховує кількість спільних оцінок при обчисленні схожості [4]. Регуляризована формула виглядає так:

$$sim_{reg}(u,v) = \frac{n}{n+\lambda} \cdot sim(u,v) \quad (2)$$

де: n – кількість спільно оцінених об'єктів;

$\lambda > 0$ – регуляризаційний параметр, що визначає ступінь згладжування.

Таким чином, чим менше спільних оцінок між користувачами, тим сильніше згладжується значення схожості. Це дозволяє зменшити вплив випадкових або статистично ненадійних збігів.
На основі скоригованої схожості обчислюється прогнозована оцінка для користувача u щодо об'єкта i:

$$\hat{r}_{u,i} = \overline{r_u} + \frac{\sum_{v \in N_u(i)} sim_{reg}(u,v) \cdot (r_{v,i} - \overline{r_v})}{\sum_{v \in N_u(i)} |sim_{reg}(u,v)|} \quad (3)$$

$N_u(i)$ – множина користувачів, які оцінили об'єкт i та мають схожість із користувачем u.

Ще один помітний ефект — це стабільність метрик Precision і F1-score при зміні рівня відсутніх даних. Якщо для класичного алгоритму ці значення знижуються вже при 30–35% пропусків, то у вдосконаленого варіанту спостерігається стійкий результат навіть при 50% порожніх комірок. Це свідчить не лише про поліпшення абсолютних значень похибки, але й про більш надійне функціонування алгоритму за умов нестачі інформації.

Особливу увагу було приділено впливу параметра регуляризації λ. Як показали експерименти, надто малі значення цього параметра не усувають проблему нестабільності, тоді як надто великі можуть призводити до заниження оцінки схожості навіть у релевантних парах. Оптимальним виявився діапазон λ від 10 до 40, де баланс між надійністю і чутливістю алгоритму є найбільш ефективним.

Загалом, результати свідчать про помітну зміну динаміки похибок і точності залежно від розрідженості, причому саме вдосконалений метод демонструє більш передбачувану поведінку, що особливо важливо в практичних сценаріях з неповними даними.

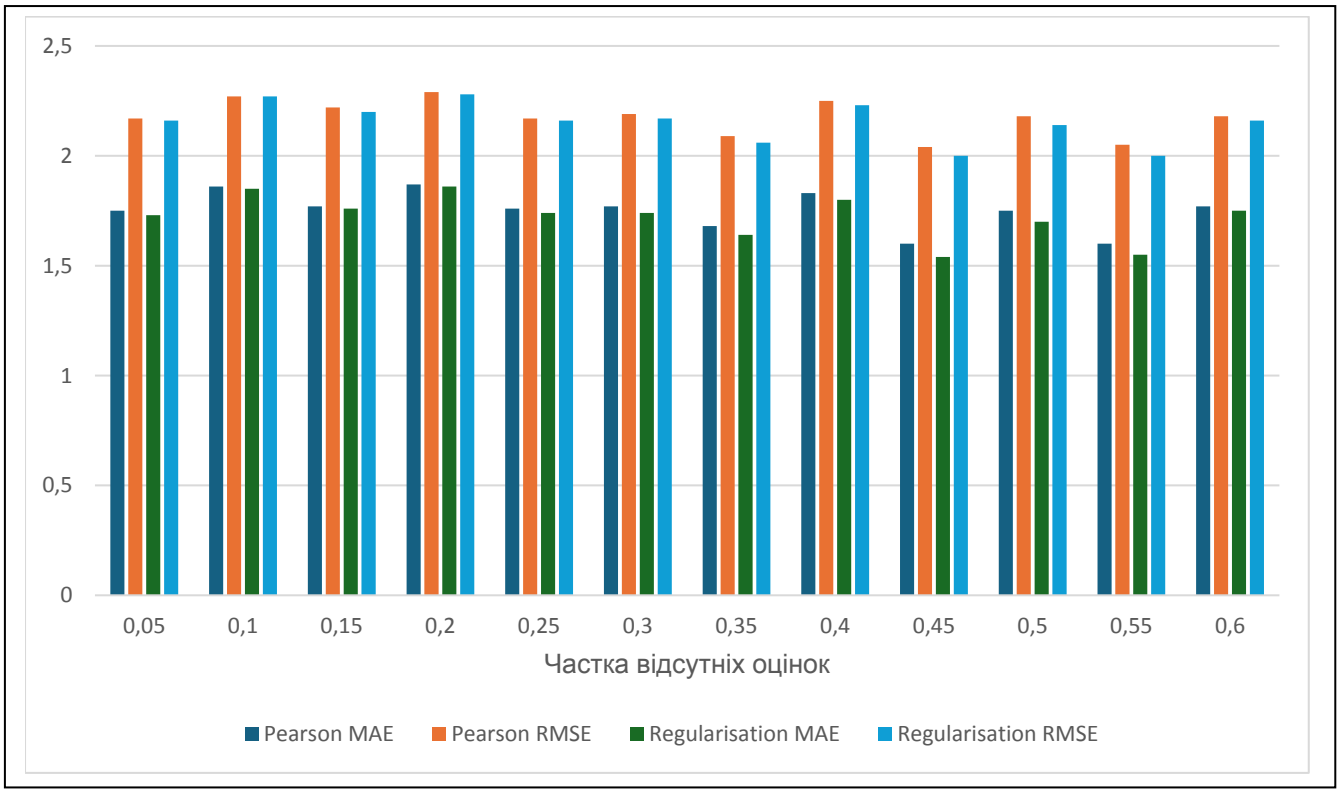

Рис. 1. Порівняння критеріїв оцінки рекомендацій для кореляції Пірсона та удосконаленого методу для різної частки відсутніх оцінок

Висновки

Запропонований у роботі метод удосконалення кореляції Пірсона шляхом регуляризації демонструє підвищену стійкість до проблеми розрідженості матриці. Завдяки врахуванню кількості спільно оцінених об'єктів в обчисленні схожості вдалося зменшити похибки та покращити якість рекомендацій. Отримані результати підтверджують доцільність застосування цього підходу у практичних системах, де дані є неповними або частково відсутні.

Література

1. Anwar T., Uma V. Comparative study of recommender system approaches and movie recommendation using collaborative filtering // Int. J. Syst. Assurance Eng. Manage. – 2021. – Vol. 12, No. 3. – P. 426–436.

2. Haruna K., Akmar I. M., Damiasih D. et al. A collaborative approach for research paper recommender system // PLoS ONE. – 2017. – Vol. 12, No. 10. – e0184516.

3. Lin K., Sonboli N., Mobasher B., Burke R. Calibration in collaborative filtering recommender systems: a user-centered analysis // Proc. 31st ACM Conf. on Hypertext and Social Media. – 2020. – P. 197–206.

4. Kumar P., Gupta M., Rao C. et al. A Comparative Analysis of Collaborative Filtering Similarity Measurements for Recommendation Systems // Int. J. Recent and Innovation Trends in Computing and Communication. – 2023. – Vol. 11. – P. 184–192.

Vehicle Identification Method Based on Combined Video Stream Analysis

Andrii Chykrii, Dmytro Halushko
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

Abstract. **The paper proposes a method for object identification based on the analysis of multiple video streams. An algorithm and data structure for effective clustering and fusion of spatial-probabilistic data are developed. The method's stability to spatial errors is analyzed. Examples are given that clearly illustrate the influence of some parameters of the surveying equipment on the stability of the method. The paper implements and tests a prototype of an automated system.**

Keywords*: *recognition, identification, video stream, aerial photography, data fusion, data aggregation, clustering.

Метод ідентифікації транспортних засобів на базі аналізу комбінованих відеопотоків

Чикрій Андрій, Галушко Дмитро
КПІ імені Ігоря Сікорського
м. Київ, Україна

Анотація.**У роботі запропоновано метод ідентифікації об'єктів на базі аналізу кількох відеопотоків. Розроблено алгоритм та структура даних для ефективної кластеризації та злиття просторово-ймовірностних даних. Проведено аналіз стійкості методу до просторових похибок. Наведено приклади, які наочно ілюструють вплив деяких параметрів засобів зйомки на стійкість методу. У роботі реалізовано та протестовано прототип автоматизованої системи.**

Ключові слова:розпізнавання, ідентифікація, відеопотік, аерозйомка, злиття даних, агрегація даних, кластеризація.

Вступ

Отримання максимально точної і повної інформації щодо розташування рухомих наземних об'єктів певних типів на великій площі потребує великої кількості засобів аерозйомки, оскільки кожен такий засіб має обмежену зону видимості. З іншого боку, якщо в нас достатньо щільне покриття такими засобами, то виникає проблема надлишковості даних, бо в поле зору різних камер можуть потрапляти одні й ті самі об'єкти. Саме тому потрібен специфічний аналіз даних мультикамерної зйомки, який враховує усі ці аспекти. Проведений аналіз предметної області та існуючих рішень [1-10] дозволив сформувати бачення щодо розроблення ефективного інструмента для аналіза даних мультикамерної аерозйомки, який дозволяє ідентифікувати та відстежувати рухомі об'єкти на великій площі. Отже, актуальність дослідження визначається потребою в ефективному інструменті для аналізу даних мультикамерної аерозйомки, який дозволяє ідентифікувати та відстежувати рухомі об'єкти на великій площі.

СУТЬ МЕТОДУ

Отримання максимально точної і повної інформації щодо розташування рухомих об'єктів певних типів на великій площі потребує великої кількості засобів аерозйомки, бо кожен такий засіб має обмежену зону видимості. Метою роботи є розробка ефективного інструмента для аналізу та злиття даних мультикамерної аерозйомки, який дозволяє ідентифікувати та відстежувати рухомі об'єкти на великій площі. Розроблено метод ідентифікації рухомих об’єктів на базі аналізу комбінованих відеопотоків.

Розглянуто ситуацію, коли у тривимірному просторі знаходиться кілька рухомих камер, кожна з яких окрім координат має три кути повороту та ступінь збільшення:

$$C_i(t) = \{x_i(t), y_i(t), z_i(t), \alpha_i(t), \beta_i(t), \gamma_i(t), zoom_i(t) \mid i = 1,\ldots,n\}, \quad (1)$$

де $x_i(t), y_i(t), z_i(t)$ – координати камер, які залежать від часу; $\alpha_i(t), \beta_i(t), \gamma_i(t)$ – кути повороту камер, які залежать від час; $zoom_i(t)$ – коефіцієнти оптичного збільшення камер, які залежать від часу.

На площині XY знаходиться певна кількість об’єктів, за якими спостерігають ці камери (рис. 1). Заздалегідь визначена певна кількість класів об’єктів

$K = \{k \mid k = 1,\dots,m\}$. Кожній камері відповідає екран, на якому транслюється те, що бачить камера: $S_i(t) = \{[0, width_i] \times [0, height_i] \mid i,\dots,n\}$. Порядковий номер камери будемо називати її ідентифікатором. Запропонована система розпізнавання ідентифікує об'єкт на екрані – визначає його координати (x_s, y_s), його клас та ступінь довіри до результата розпізнавання $(confidence)$.

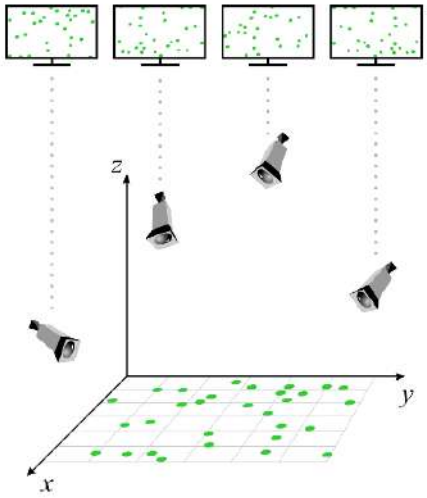

Рис. 1. Схема розташування камер

Введено величину $\varepsilon > 0$. – найменша можлива відстань між будь-якими об'єктами, зафіксованими однією камерою, а координатну площину розбито на клітини розміром δ, $\delta \geq \varepsilon$, де кожна клітина проіндексована: $\{cell_{ij} \mid i, j \in Z\}$.

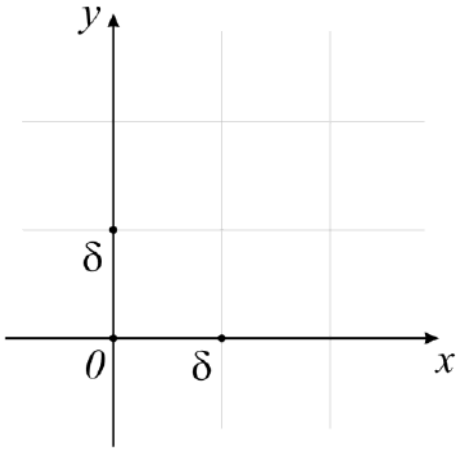

Рис. 2. Розбиття координатної площини

Для визначення класу групи обрано принцип зваженого голосування. Для оптимізації обчислень при додаванні нових об'єктів використано просторове хешування. На рис. 3 сірі клітини – непорожні, білі – порожні; колір кружечків – клас об'єкту; колір контуру кружечка – камера, з якої зафіксований об'єкт; коло – позначення об'єктів, що належать до однієї групи; колір кола і хрестику – клас групи; координати хрестику – координати групи.

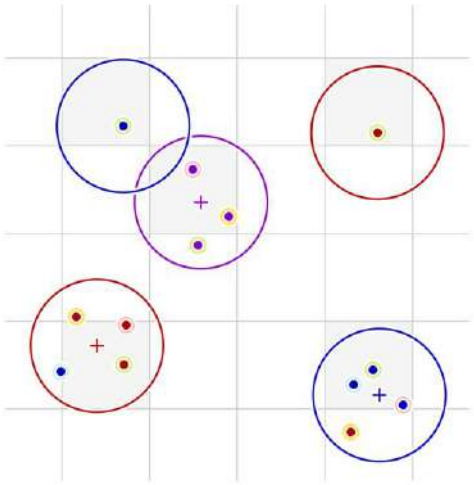

Рис. 3. Просторове хешування

Для додавання нового об'єкту до множини злитих даних виконуються наступні кроки:

– знаходимо клітину, яка відповідає координатам об'єкта,
– по цій клітині та сусіднім 8 клітинам шукаємо групи, у яких нема об'єкту з таким самим ідентифікатором камери, до яких відстань менша за ε.

Якщо такі групи існують, то

– серед знайдених груп обираємо групу, до якої найменша відстань,
– додаємо об'єкт до групи,
– обчислюємо координати групи:

$$x_g = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}, \quad y_g = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i}, \tag{2}$$

де n — кількість дочірніх об'єктів, (x_i, y_i) — координати дочірнього об'єкту, w_i — його вага.

– якщо координати групи після цього перемістилися в іншу клітину, то переносимо групу в іншу клітину,
– визначаємо клас групи:

$$class_g = arg \max_K W_K, \tag{3}$$

$$W_K = \sum_{i \epsilon I_K} w_i, \tag{4}$$

де I_K — це індекси об'єктів класу K, і визначаємо клас групи як клас, на якому досягається максимум таких сум

Якщо таких груп нема, то додаємо об'єкт так само, як ми додавали перший об'єкт. За цим алгоритмом додаємо усі об'єкти набору D. Отримані координати груп та класи цих груп — це і є результат злиття даних і остаточної ідентифікації об'єктів.

Результатом роботи запропонованого алгоритму є координати груп та класи груп, що є результатом злиття даних і остаточної ідентифікації об'єктів.

РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

Аналітично доведено, що часова складнісь алгоритму злиття даних - *O(n).*

Проведено вимірювання, за результатами яких:

– Експеременталь підтверджена часова складність алгоритма злиття даних

- Отримана статистична інформація щодо створених під час роботи алгоритма проміжних структур даних
- Досліджена залежність часу виконання алгоритму від деяких параметрів

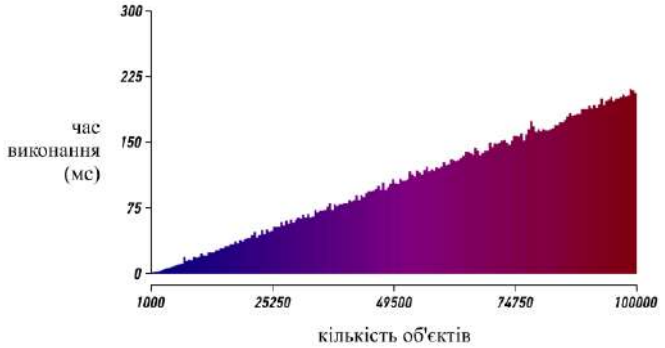

Рис. 4. Залежність часу виконання алгоритму від кількості об’єктів

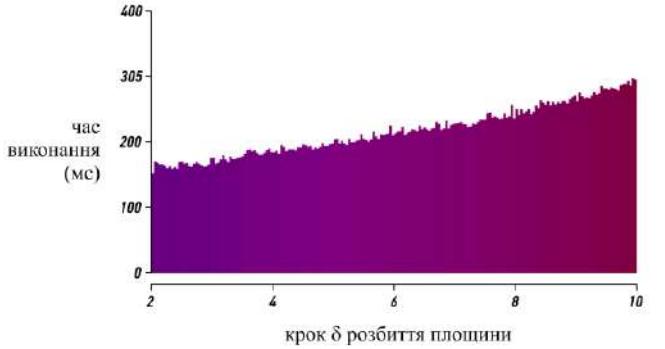

Рис. 5. Залежність часу виконання алгоритму від кроку розбиття площини

Висновки

Розроблено метод ідентифікації транспортних засобів на базі кількох відеопотоків. Проведено аналіз стійкості методу до просторових похибок. Розглянуто приклад, який наочно ілюструє вплив параметрів камер на стійкість методу. Реалізовано та протестовано прототип автоматизованої системи. Експерементально підтверджено часову складність алгоритма злиття даних O(n), досліджено залежність часу виконання алгоритму від кількості об’єктів та кроку розбиття площини.

Література

1. Everitt B. S., Landau S., Leese M., Stahl D. Cluster Analysis. 5th Edition. Chichester: John Wiley & Sons, Ltd, 2011. 352 p.
2. Aggarwal C. Data clustering: Algoriths and Applications. CRC Press, 2014. 617 p.
3. Hall D. L. and Llinas J. An introduction to multisensor data fusion, Proceedings of the IEEE. 1997. Vol. 85, no. 1. P. 6–23 doi: 10.1109/5.554205.
4. Blackman S. S. Association and fusion of multiple sensor data. Multitarget-Multisensor: Tracking Advanced Applications. Artech House. 1990. P. 187–217.
5. Federico Castanedo. A Review of Data Fusion Techniques. The Scientific World Journal. 2013. Volume 2013, Issue 1, 704504. doi: 10.1155/2013/704504
6. Dasarathy B. V. Sensor fusion potential exploitation-innovative architectures and illustrative applications. Proceedings of the IEEE. 1997. Vol. 85, no. 1. P. 24–38. doi: 10.1109/5.554206.
7. Date C. J. An Introduction to Database Systems. Pearson Education, 2004. 1034 p.
8. Jain A., Murty M., Flynn P. Data Clustering: A Review. ACM Computing Surveys. 1999. Vol. 31, no. 3. P. 264–323.
9. Berkhin P. Survey of clustering data mining techniques. Technical report. San Jose: Accrue Software, 2002. 56 p.
10. Quijano N., Passino K. M. Honeybee Social Foraging Algorithms for Resource Allocation: Theory and Application. Columbus: Publishing house of Ohio State University, 2007. 39 p.
11. Чикрій Ан.О. Автоматизована система забезпечення ситуаційної обізнаності із застосуванням безпілотних авіаційних комплексів / Ан.О.Чикрій // Міжнародний науково-технічний журнал «Проблеми керування та інформатики». – 2025. – № 1. – С. 60-71.

The Current State of Automated Analysis of Scientific Publication Activity Using Digital Identifier Data

Dmytro Miach, Yana Romashkevych, Mariia Soldatova
Faculty of Informatic and Computer Engineering
Igor Sikorsky Kyiv Polytechnical Institute
Kyiv, Ukraine
d.miach@kpi.ua, y.romashkevych@gmail.com, benten1093@gmail.com

Abstract. **Monitoring the publication output of academic staff is crucial for internal performance evaluation and for shaping an institution's profile in external rankings. Automated analyses leveraging digital identifier data aim to streamline and accelerate this process but are subject to specific vulnerabilities, which are examined in this article.**

Keywords*: *DOI, publication activity, bibliographic description, information system.

Сучасний стан автоматизованого аналізу публікаційної активності науковців на основі даних цифрового ідентифікатора

М'яч Дмитро, Ромашкевич Яна, Солдатова Марія
КПІ імені Ігоря Сікорського
м. Київ, Україна
d.miach@kpi.ua, y.romashkevych@gmail.com, benten1093@gmail.com

Анотація. **Публікаційна активність викладачів та науковців є важливим критерієм в контексті проведення внутрішнього оцінювання (рейтингування) роботи науково-педагогічних працівників, а також для формування портрету закладу вищої освіти (ЗВО) в зовнішніх рейтингах. Автоматизований аналіз публікаційної активності на основі даних цифрового ідентифікатора покликаний спростити та пришвидшити даний процес, проте має окремі вразливості, які будуть розглянуті в контексті даної публікації.**

Ключові слова: DOI, публікаційна активність, бібліографічний опис, інформаційна система.

Вступ

Публікаційна активність науково-педагогічного працівника (далі - НПП) відіграє важливу роль у виконанні ним Кадрових вимоги щодо освітньої діяльності за рівнем вищої освіти відповідно до Ліцензійних умов провадження освітньої діяльності [1], а також в межах КПІ ім. Ігоря Сікорського – в процесі проведення конкурсного відбору або обрання за конкурсом при заміщенні вакантних посад науково-педагогічних працівників. [2].

Впровадження автоматизованого аналізу публікаційної активності науковців має забезпечити збір інформації щодо публікацій і забезпечення уніфікації в підсистемі рейтингування для подальшого використання.

Аналіз існуючого стану збору даних

З 2017 року Науково-технічна бібліотека ім. Г. І. Денисенка (далі – НТБ) здійснює моніторинг та аналіз публікаційної активності НПП та вчених, які афілійовані з КПІ ім. Ігоря Сікорського. Базою для моніторингу є наукометричні бази даних Scopus та Web of Science Core Collection.

За результатами моніторингу та аналізу:

- укладаються переліки працівників, публікації яких оприлюднені у виданнях, що входять до міжнародних наукометричних баз даних Scopus та/або Web of Science Core Collection та відповідають критеріям преміювання;
- укладають переліки нових публікацій вчених КПІ ім. Ігоря Сікорського [3];
- з 2020 року ведеться облік усіх публікацій (з усіма показниками впливовості);

- готуються додатки на запит НДЧ для подальшого подання до ЄДЕБО (атестація за науковими напрямами, держзамовлення магістрів тощо).

З 2020 року в межах автоматизованої інформаційної системи «Електронний Кампус» реалізовано підсистему «Рейтинг НПП», в якій відповідно до Норм бального оцінювання діяльності науково-педагогічних працівників здійснювався збір інформації щодо виконання НПП зазначених критеріїв із їх подальшим оцінюванням [4].

Відповідно до вказаних Норм, проводиться оцінювання публікацій НПП відповідно до критеріїв щодо категорій видань, а також щодо участі в співавторах здобувачів вищої освіти та/або іноземних учених [5].

В існуючій системі внесення та перевірка внесених даних відбувається в ручному режимі шляхом залучення фахівців НТБ у ролі «відповідального за доповнення інформації» та «адміністратору пункту» [6].

Рис. 1. Кількість записів за напрямками діяльності з науково-інноваційної роботи за 2023-2024 н.р.

Відповідно до даних, наявних в підсистемі «Рейтинг НПП» за 2023-2024 н. р. публікації складають 22,8% записів від всіх записів внесених за напрямком «Науково-інноваційна робота» (рис.1).

Таблиця 1

Відомості щодо внесених DOI статей

Тип публікації (статті)	**Всього записів**	**Наявність DOI**
– стаття у журналах, що індексуються базами Scopus та/або Web of Science*, віднесених до Q1	198	173
– стаття у журналах, що індексуються базами Scopus та/або Web of Science*, віднесених до Q2	237	218
– стаття у журналах, що індексуються базами Scopus та/або Web of Science*, віднесених до Q3	600	525
– стаття у журналах, що індексуються базами Scopus та/або Web of Science*, віднесених до Q4	628	492
– стаття у журналах, що індексуються базами Scopus та/або Web of Science*, та не мають квартилю	103	70
– стаття у зарубіжних періодичних наукових виданнях країн ОЕСР, що не індексуються базами Scopus та/або Web of Science	224	107
– стаття у фахових журналах категорії Б, що не індексуються базами Scopus та/або Web of Science	2356	1501
– стаття у фахових журналах категорії Б, що не індексуються базами Scopus та/або Web of Science (англійською)	933	647

Вимога щодо внесення DOI (digital object identifier) для публікацій була необов'язковою, але як видно з табл. 1, в якій наведено інформацію щодо внесених даних про DOI в описі публікацій, то більше ніж в 67% статей, що індексуються базами Scopus та/або Web of Science, вказано даний ідентифікатор, у видань категорії Б це значення коливається від 63 до 69%

Використання зовнішнього ідентифікатору публікацій для аналізу

DOI як унікальний цифровий ідентифікатор об'єкта (згідно стандарту ISO 26324) з одного боку, а також інструмент по скороченню посилань з іншого боку, виступає багатофункціональним інструментом для уніфікації інформації щодо внесених публікацій [7].

Експериментальне очищення даних від дублікатів щодо публікацій, що містить DOI без їх додаткової верифікації, показало, що із 3765 записів за текстовим описом унікальними виявились 3600 записів, а за DOI – 2760. Даний показник доводить, що ручне внесення інформації різними особами (надавачами) призводить до виникнення більшої кількості дублюючих записів, які не піддаються автоматичному виявленню.

Аналіз публікація на дублікати важливий для того, щоб уникнути спотворення інформації при формуванні подальших звітів в контексті, який буде відділений від сутності працівника в контекст публікацій.

За допомогою API від Crossref із використанням DOI можна отримати інформацію щодо метаданих для кожної із публікацій, які передаються видавцем до кожного із ідентифікаторів [8].

Подальше очищення даних шляхом перевірки коректності DOI та наявності активних метаданих щодо публікації дозволило відсіяти ще 11% записів.

Безпосередньо для наповнення бази публікаційної активності із метою подальшого аналізу можна використовувати методи Crossref REST API, які повертають метадані щодо публікацій у вигляді JSON-файлів.

В структурі об'єкту, що повертається на запит по конкретному DOI, особливо важливими для автоматизування системи можна виділити наступні поля:

- PublishedDate –дозволяє визначити чи відноситься публікація до вказаного періоду рейтингування (оцінювання);
- Author_1_Orcid, Author_2_Orcid тощо – дозволяє автоматизовано визначити особу автора в системі;
- Author_1_Affiliation, Author_2_Affiliation тощо – дозволяє визначити чи вказано місце місце роботи автора, в іншому випадку – чи вказано саме КПІ ім. Ігоря Сікорського, якщо науковець подає її в межах Університету.

Не менш критичним для автоматизування процесу обробки інформації в межах системи та зменшення навантаження на працівників, призначених на ролі «відповідального за доповнення інформації» та «адміністратору пункту» є наповнення довідника Університету даними щодо ORCID профілів науковців.

Важливим є те, щоб ORCID профіль науковця був актуальним і використовувався автором при подачі публікацій.

В такому випадку роль НТБ від постійного збору інформації переходить до верифікація та вдосконалення метаданих.

В контексті бази публікаційної активності бібліотекарі можуть і повинні: верифікувати надіслані авторами DOI, виправляти та уточнювати метадані (наприклад, уніфікувати афіліації), контролювати коректність записів у базі, щоб дані були точними й відповідали стандартам.

У табл. 2 наведено показники проаналізованих метаданих публікацій, що вже наявні в підсистемі «Рейтинг НПП» та були внесені з вказанням DOI із відповідними полями, які необхідні для формування бібліографічного опису публікацій із подальшим збереження в базу публікаційної активності.

Таблиця 2

Статистика щодо полів метаданих публікацій

Поле JSON	**К-сть**	**%**	**Відповідник**
DOI	2451	100	
Status	2451	100	Актуальність ідентифікатора
Title	2398	97,84	Назва публікації
Publisher	2448	99,88	Видавець
PublishedDate	2437	99,43	Дата видання
Type	2448	99,88	Тип публікації
Author_1_Given	2091	85,31	Автор 1, ім'я
Author_1_Family	2101	85,72	Автор 1, прізвище
Author_1_Orcid	1149	46,88	Автор 1, ORCID
Author_1_Affiliation	331	13,50	Автор 1, афіляція
Author_2_Given	2134	87,07	Автор 2, ім'я
Author_2_Family	2143	87,43	Автор 2, прізвище
Author_2_Orcid	1151	46,96	Автор 2, ORCID
Author_2_Affiliation	317	12,93	Автор 2, афіляція
Author_3_Given	1426	58,18	Автор 3, ім'я
Author_3_Family	1433	58,47	Автор 3, прізвище
Author_3_Orcid	722	29,46	Автор 3, ORCID
Author_3_Affiliation	266	10,85	Автор 3, афіляція
Author_4_Given	830	33,86	Автор 4, ім'я
Author_4_Family	835	34,07	Автор 4, прізвище
Author_4_Orcid	411	16,77	Автор 4, ORCID
Author_4_Affiliation	214	8,73	Автор 4, афіляція
Author_5_Given	658	26,85	Автор 5, ім'я
Author_5_Family	661	26,97	Автор 5, прізвище
Author_5_Orcid	326	13,30	Автор 5, ORCID
Author_5_Affiliation	152	6,20	Автор 5, афіляція

Відштовхуючись від даних табл. 2 варто зауважити, що необов'язковість полів при заповненні метаданих публікацій призводить до відсутності в половини авторів вказаного ORCID. Проте, зважаючи на об'єми даних, які підлягають обробці, вказаний відсоток дозволяє частково автоматизувати процес.

Висновки

Використання цифрового ідентифікатора DOI має важливе значення для покращення процесу збору, уніфікації та аналізу публікаційної діяльності науковця, проте потребує залучення додаткових інформаційних та людських ресурсів для повноти отриманої інформації.

Література

1. Про затвердження Ліцензійних умов провадження освітньої діяльності: Постанова КМУ від 30.12.2015 № 1187: станом на 4 січ. 2024 р. URL: https://zakon.rada.gov.ua/laws/show/1187-2015-п#Text (дата звернення: 12.05.2025)
2. Про затвердження Порядку проведення конкурсного відбору або обрання за конкурсом при заміщенні вакантних посад науково-педагогічних працівників та укладання з ними трудових договорів (контрактів): Наказ КПІ ім. Ігоря Сікорського від 24.09.2021 р. № НУ/201/2021. URL: https://osvita.kpi.ua/sites/default/files/downloads/Наказ_№НУ_201_2021_від_24_09_2021.pdf .
3. Оцінка | Науково-технічна бібліотека ім. Г. І. Денисенка. URL: https://www.library.kpi.ua/research/otsinka/#2 (дата звернення: 09.05.2025).
4. М'яч Д. О. Підсистема «Рейтинг НПП» АІС «Електронний кампус»: індивідуальний дослідницький проєкт… бакалавра: 126 Інформаційні системи та технології / М'яч Дмитро Олександрович. — Київ, 2022. — 64 с. URL: https://ela.kpi.ua/handle/123456789/57792 (дата звернення 09.05.2025)
5. Про затвердження Положення про рейтингування науково-педагогічних працівників КПІ ім. Ігоря Сікорського та норм бального оцінювання діяльності науково-педагогічних працівників: Наказ від 25.12.2023 р. № НОН/386/2023
6. Розподіл ролей користувачів системи рейтингування науково-педагогічних працівників / М. О. Солдатова та ін. Телекомунікаційні та інформаційні технології. 2024. Т. 82, № 1. С. 106–113. URL: https://doi.org/10.31673/2412-4338.2024.019901 (дата звернення: 07.05.2025).
7. DOI® Resolution Documentation. URL: https://doi.org/the-identifier/resources/factsheets/doi-resolution-documentation (date of access: 08.05.2025).
8. Crossref REST API. Swagger UI. URL: https://api.crossref.org/swagger-ui/index.html (date of access: 08.05.2025).

USING AN ADAPTIVE APPROACH TO FINANCIAL PORTFOLIO FORMATION METHODS

Terentiev Hlib, Havrylenko Olena
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
terentiev.hlib@lll.kpi.ua, gelena1980@gmail.com

Abstract. **The paper compares the use of a method based on genetic algorithms and a method based on quadratic programming for selecting asset ratios when forming an investment portfolio based on asset diversity, savings diversification, and portfolio profitability. The combined use of these methods is suggested using and adaptive approach.**

Keywords: quadratic programming, genetic algorithms, diversification.

ВИКОРИСТАННЯ АДАПТИВНОГО ПІДХОДУ ДЛЯ МЕТОДІВ ФОРМУВАННЯ ФІНАНСОВИХ ПОРТФЕЛІВ

Терентьєв Гліб, Гавриленко Олена
КПІ імені Ігоря Сікорського”
м. Київ, Україна
terentiev.hlib@lll.kpi.ua, gelena1980@gmail.com

Анотація. **У доповіді порівнюється використання методу базованого на генетичних алгоритмах та методу базованого на квадратичному програмуванні для підбору коефіцієнтів активів при формуванні інвестиційного портфелю на основі різноманітності активів, диверсифікації заощаджень та прибутковості портфелю. Та пропонується їх використання сумісно використовуючи адаптивний підхід.**

Ключові слова: квадратичне програмування, генетичні алгоритми, диверисфікація.

Вступ

Акції як цінні папери вже існують сотні років, при цьому це не зменшує їх популярності та цінності сьогодні. Публічні акціонерні компанії випускають пакет акцій із метою залучити кошти для розвитку. Цінні папери можуть надавати право на керування компанією через співвласництво та отримання відсотку від прибутку товариства у вигляді дивідендів. Утім, ризик хибних рішень від управління компанії, загальна політична ситуація у країні базування компанії, проблеми із залучення нових клієнтів при маркетинговій компанії та нездатність розуміти потреби існуючих клієнтів можуть призвести до зменшення прибутковості, що безпосередньо вплине на вартість цінних паперів, випущених цим товариством. У найгіршому випадку на товариство чекатиме банкрутство. Для зменшення ризику втрати коштів інвестори шукають прибуткові активи, диверсифікують інвестиції, вкладаючи у різні акціонерні товариства та незалежні сфери таким чином створюючи інвестиційний портфель.

При цьому існують різні стратегії інвестування. Використання цих стратегій залежить від мети та цілей інвестора. У цій роботі проводиться огляд методів підбору коефіцієнтів активів портфелю за допомогою методу, що використовує квадратичне програмування, та методу, що використовує генетичні алгоритми, для відображення користувацьких стратегій. Дослідження поєднує обидва методи у загальний метод, який ґрунтується на використання адаптивного підходу для врахування цілей вкладника.

МАТЕМАТИЧНА МОДЕЛЬ

ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

ПОСТАНОВКА ЗАДАЧІ

На основі історичних даних із заданою періодичністю, зазвичай із денним періодом, про ціни акцій на період відкриття і закриття денних торгів та заданої множини активів підібрати множину активів і їх співвідношення у інвестиційному портфелі.

Задачею є знаходження фінансового портфелю, який буде рости з часом у ціні для отримання прибутку та буде достатньо диверсифікованим. Поняття потрібної чи достатньої диверсифікованості кожен інвестор визначає самостійно на основі власних переконань. Загалом це поняття пов'язують зі зниженням загального ризику портфелю.

Метою дослідження є порівняння прибутковості портфелю створеного за допомогою генетичних алгоритмів та портфелю створеного за допомогою квадратичного програмування та створення методу підбору фінансових портфелів на їх основі.

МАТЕМАТИЧНА МОДЕЛЬ

Задачу квадратичного програмування з n змінних та m обмежень визначають наступним чином [1]. Де ціллю є знайти n-просторовий вектор x, що мінімізуватиме

$$\min\left\{\frac{1}{2}x^T Px + q^T x\right\}, x \qquad (1)$$

за умови, що

$$Gx \leq h \qquad (2)$$

$$Ax = b \qquad (3)$$

Тобто кожен елемент вектору результату Gx буде меншим чи рівним відповідному компоненту h. Де,

q— вектор простору n;

P — симетрична матриця $n \times n$;

h — вектор простору n;

G — матриця $m \times n$.

Для вирішення цієї задачі пропонується використовувати модуль CVXOPT [2].

Визначимо задачу максимізації для генетичних алгоритмів для створення інвестиційних портфелів. Формально задача визначається розв'язком наступної задачі:

$$\max(f(x)), x \in \Omega \qquad (4)$$

де Ω є множиною усіх можливих розв'язків, тобто варіантів інвестиційних портфелів для заданих вхідних даних. Як функція відповідності у генетичному алгоритму використовується коефіцієнт Шарпа [3]. Його задачею є одночасна максимізація прибутковості, тобто середнього значення прибутку та мінімізація ризику, тобто середньоквадратичного відхилення прибутів.

МЕТОДОЛОГІЯ

У ході експерименту ітеративно запускаються задачі генетичного алгоритму. Задача квадратичного програмування запускається один раз. Надалі порівнюється прибутковість за рахунок реальних значень. Використовувались фінансові дані індексу Національної фондової біржі (NSE) Індії NIFTY 50 з 01.01.2015 по 01.01.2019 [1]. Як вартість активу використовується ціна на момент закриття торгів.

РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТУ

Експеримент проводився для оцінки найкращих створених портфелів обома методами задля цього було запущено 15 ітерацій експерименту для отримання найкращого портфелю створеного за допомогою генетичних алгоритмів серед них. Для квадратичного програмування використовується 100 створених рівнянь для пошуку кращого результату.

На рисунку можна побачити менший прибуток від портфелю створеного за допомогою квадратичного програмування у перші місяці.

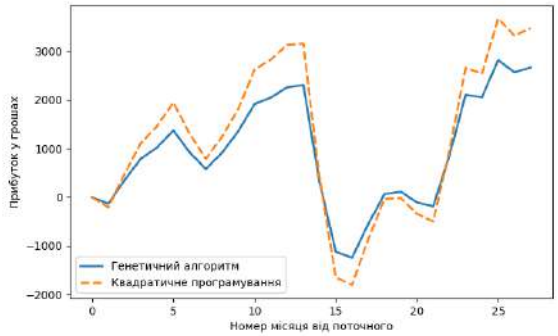

Рисунок 1. Порівняння прибутковості створеного портфелю методами генетичного алгоритму та квадратичного програмування

Надалі спостерігається схожий тренд прибутку між обома портфелями, незалежно від того чи це від'ємний чи додатній прибуток. Проте, портфель, створений методом генетичних алгоритмів хоч і приносить менший прибуток на певних інтервалах, він має і менші втрати. Щоб розібратися з причиною такої поведінки розглянемо активи обох портфелів із відсотковим значенням у портфелі більше 1%.

У наступній таблиці наведемо активи портфелю, створеного за допомогою квадратичного програмування та їх відповідну частку. Часткою буде десятковий дріб, що відповідає відсотковому відношенню активу до всього портфелю. Наприклад, якщо портфель створено на 1000 грн., то на актив із часткою 0.6 рекомендовано алгоритмом витратити 600 грн.

Таблиця 1. Таблиця значущих активів портфелю, створеного за допомогою квадратичного програмування

Тикерна назва активу	Частка активу у портфелі
BAJAJFINSV	0.99999557

Наведемо аналогічну таблицю для портфелю, створеного генетичними алгоритмами за схожим принципом до попередньої таблиці.

Таблиця 2. Таблиця значущих активів портфелю, створеного за допомогою генетичних алгоритмів

Тикерна назва активу	Частка активу у портфелі
BAJAJFINSV	0.670
ASIANPAINT	0.131
BAJFINANCE	0.054
BAJAJ-AUTO	0.017
JSWSTEEL	0.017
TCS	0.011
BRITANNIA	0.010

У таблиці активів портфелю, створеного за допомогою генетичних алгоритмів, наведено 91% портфелю, усі інші активи не пройшли поріг фільтрування для відображення у таблиці. З обох таблиць можна помітити, що метод квадратичного програмування, коли знаходить ефективний актив

він повністю максимізуватиме його частку у портфелі. Це є доречною стратегією, але більш ризиковою. Головною задачею створення портфелю є максимізація прибутку та мінімізація ризику, у даному випадку прибуток перевершив ризик. У той час як портфель з схожим відношенням Шарпа є більш диверсифікованим, хоча теж робить вибір на тому ж активі. Це знижує ризики та слідує правилу інвестування «не вкладати усі яйця до єдиного кошика».

Приведемо наступний приклад для розуміння різниці між портфелями, при інвестуванні до портфелю з таблиці 1 у сценарії при якому компанія, що випускає актив, збанкрутувала, Ви втрачаєте 100% свого портфелю, своїх активів і вкладених коштів. При інвестуванні до портфелю із таблиці 2 при тому ж сценарії втрати активу під кодом BAJAJFINSV Ви втратите лише частину свого портфелю, а саме 67% коштів, проте Ви збережете 33% інвестованих коштів. Тим самим другий портфель є більш диверсифікованим.

Висновки

Провівши порівняння результатів, а саме створених інвестиційних портфелів, на прибутковість та різноманіття між методами квадратичного програмування та генетичних алгоритмів, метод квадратичного програмування показав більш агресивну стратегію інвестування, яка має на меті отримати найбільший прибуток, у той час як метод створення портфелю за допомогою генетичних алгоритмів притримується більш помірної та зваженої стратегії інвестування, яка вбачає диверсифікацію, усереднення чи мінімізацію ризиків. Це гарантує більший захист від зовнішніх політичних та суспільних впливів на ринок, які цей метод не може передбачити.

На основі цих результатів пропонується одночасне використання обох методів підбору портфелю для створення методу на основі адаптивного підходу, який враховуватиме агресивність стратегії користувача. Для агресивних стратегій, які зосереджені на збільшенні прибутку понад усе, пропонується використовувати метод квадратичного програмування, у той час як для помірної та низькоризикової стратегій інвестування пропонується використовувати методи базовані на алгоритмах із класу генетичних, які максимізуватимуть відношення Шарпа, або мінімізуватимуть ризик через дисперсію прибутку.

Література

1. Nocedal, Jorge; Wright, Stephen J. Numerical Optimization (2nd ed.). Berlin, New York: Springer-Verlag. 2006. p. 449

2. L. Vandenberghe, The CVXOPT linear and quadratic cone program solvers, March 20, 2010. URL: https://www.seas.ucla.edu/~vandenbe/publications/coneprog.pdf

3. SHARPE, William F. Mutual fund performance. *The Journal of business*, 1966, 39.1: p. 119-138.

4. NIFTY-50 Stock Market Data (2000 - 2021) / Kaggle. URL: https://www.kaggle.com/datasets/rohanrao/nifty50-stock-market-data.

Method for Calculating the Integral Indicator of the Risk of Changes to a Software Product

Dmytro Halushko, Kyrylo Znova
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
dmytro.halushko@lll.kpi.ua

Abstract. **An approach for quantitatively assessing change risk in a software product is presented. The method compares two versions of a graph model generated from technical documentation. For every node in the updated graph, three measurable quality indicators—performance efficiency, reliability, and maintainability—are calculated. Aggregating these indicators with topological and historical features yields an integral risk indicator that ranks the modified components. The indicator can be used in CI/CD pipelines, test‑planning, and technical‑debt management without relying on static code analysis.**

Keywords*: *software product quality, information services, information systems and technologies.

Метод обчислення інтегрального показника ризику змін до програмного продукту

Галушко Дмитро, Знова Кирило
КПІ імені Ігоря Сікорського
Київ, Україна
dmytro.halushko@lll.kpi.ua

Анотація. Запропоновано підхід до кількісної оцінки ризику змін у програмному продукті на основі графового представлення його структури. Пропонується алгоритм порівняння двох версій графової моделі, сформованої з технічної документації. Для кожної вершини нової версії обчислюються три вимірювані показники якості – ефективність продуктивності, надійність, підтримуваність. Згортання цих показників із топологічними та історичними характеристиками утворює інтегральний показник ризику, який ранжує змінені компоненти. Показник може використовуватись у CI/CD-конвеєрі, плануванні тестування та управлінні технічним боргом без залучення статичного аналізу коду.

Ключові слова: якість програмного продукту, інформаційні послуги, інформаційні системи та технології.

Вступ

Продуктові компанії, що виробляють програмний продукт одночасно підтримують десятки взаємозалежних продуктів, що постійно модифікуються під тиском ринку, регуляторних вимог і динаміки інфраструктури. Це змушує їх враховувати ризики введення нових змін ще під час проектування. У роботі [1] запропоновано гібридну NLP/HeteroGNN‑систему, яка автоматично конвертує технічну документацію до програмного продукту у графи дій з метою виявлення ризикованих місць у імплементації для подальшого, більш детального їх тестування. Результати довели, що такий підхід зменшує ручну працю та дозволяє пришвидшити вихід релізів. Попри це, проблематика кількісної оцінки ризику при внесенні змін у вже побудовані графи дій залишилася відкритою. Компанії здебільшого покладалися на експертні оцінки або високовартісні статичні аналізатори коду, що не завжди доступно для мікросервісних середовищ із різнорідними технологічними стеками. У цій роботі запропоновано граф‑орієнтований метод обчислення ризиків змін у програмному продукті на основі на основі аналізу послідовних версій документації та підмножини об’єктивно вимірюваних характеристик, що описані в стандарті вимог до якості систем і програмних продуктів ISO/IEC 25010:2016 ISO/IEC 25010 [2]. Запропоновано метод, що дозволяє автоматично локалізувати ризиковані зміни, які потребують пріоритетного тестування та детального проектування. Це дозволить мінімізувати ймовірність введення дефектів у продуктове середовище.

Обчислення інтегрального показника ризику змін

В роботі [1] запропоновано алгоритм автоматичного перетворення технічної документації до програмного продукту у граф послідовності дій, що треба виконати програмному продукту при запиті від користувача. В даній роботі запропоновано метод, що розширює такий підхід і дозволяє визначати ризиковані зміни до програмного продукту, що

потребують посиленої уваги. Запропоновано метод, що базується на аналізі зміни графа послідовності дій програмного продукту та метриках моніторингу його роботи.

Початковий варіант графа, сформований із базової редакції технічної документації, зберігається як $G_0 = (V_0, E_0)$, де V_0 – множина вершин початкового графу, а E_0 – множина ребер початкового графу. Після змін у технічній документації до проекту генератор сценаріїв синтезує оновлений граф $G_1 = (V_1, E_1)$.

Для кожного вузла $v_i^0 \in V_0 \mid i = \left[1, \left|V_0\right|\right]$ та $v_j^1 \in V_1 \mid j = \left[1, \left|V_1\right|\right]$ обчислюються три нормовані показники якості [2]: складність операції [3], безвідмовність [4] та структурна складність вершини графу [5-6]. Формули розрахунку наведені у (2-4) відповідно.

$$PE(v) = \frac{1}{2}\left(1 - \frac{lat_{95}(v)}{SLO_{lat}}\right) + \frac{1}{2}\left(1 - CPU(v)\right), \quad (2)$$

де PE – складність операції, що відповідає вершині графу; $lat_{95}(v)$ – 95-й перцентиль затримки відповіді операції; SLO_{lat} – декларована допустима межа латентності, що визначається у мс; $CPU(v)$ – середнє завантаження процесора у в діапазоні від 0 до 100% протягом виконання операції.

$$REL(v) = \frac{2}{3} \cdot uptime(v) + \frac{1}{3}\left(1 - \frac{MTTR(v)}{MTTR_{\max}}\right), \quad (3)$$

де REL – безвідмовність операції, що відповідає вершині графу; $uptime(v)$ – частка успішних виконань операції; $MTTR(v)$ – середній час відновлення після збою в роботі операції; $MTTR_{\max}$ – еталонна верхня межа часу відновлення.

$$MAI(v) = 1 - \min\left(1, \frac{\log_2(1 + In(v)) + \log_2(1 + Out(v)) + BC^*(v)}{C_{\max}}\right), \quad (4)$$

де $MAI(v)$ – структурна складність вершини графу; $In(v)$ та $Out(v)$ – відповідно кількість вхідних та вихідних зв'язків вершини; $BC^*(v)$ – нормована до [0;1] міжцентральність вершини, $C_{\max}$ – поріг прийнятної сумарної складності.

Після розрахунку трьох базових показників *PE*, *REL* та *MAI* для кожної вершини з множини V_1 формується інтегральний показник ризику *R(v)*:

$$\begin{aligned} R(v) &= \alpha \cdot R_{CI}(v) + \beta R_{QS}(v) + \gamma R_{BT}(v) + \delta R_{HIS}(v), \\ R_{CI}(v) &= \frac{\left|\mathrm{Reachable}(v)\right|}{\left|V\right|}, \\ R_{QS}(v) &= 1 - (PE(v) + REL(v) + MAI(v)), \\ R_{BT}(v) &= BC_1(v) - BC_0(v), \\ R_{HIS}(v) &= \frac{Commits(v)}{LOC(v)}, \end{aligned} \quad (5)$$

де α, β, γ, δ – вагові коефіцієнти, визначені емпірично, їх сума має дорівнювати 1; R_{CI} – частка підграфа, транзитивно залежного від вершини графа; *Reachable* — транзитивна множина нащадків вершини; R_{QS} – штраф за погіршення якості; R_{BT} – приріст міжцентральності, що характеризує появу «вузького місця»; R_{HIS} – історична мінливість програмного коду; $Commits$ – кількість комітів у програмний код, що реалізує дану вершину за останні 90 днів; *LOC* – поточна кількість рядків, що реалізує дану вершину.

Розрахований інтегральний показник ризику *R* – це узагальнена, безрозмірна оцінка ймовірності того, що зміни у вузлі призведуть до дефектів або деградації сервісу після релізу нової версії продукту. Він визначає: якість релізу – фокус на вузлах з високим показником під час розробки нової версії продукту може зменшити кількість дефектів у продакшн-середовищі; швидкість виходу у продакшин – чітка пріоретизація задач дає можливість час регресійного тестування; планування майбутніх релізів – накопичувальна статистика даного показника від релізу до релізу показує які частини програмного продукту падають в якості чи взагалі потребують переробки. Показник може використовуватись у плануванні тестування та управлінні технічним боргом без залучення статичного аналізу коду.

Результати експериментів

Для перевірки запропонованого метода було використано існуючий навчальний проєкт, що складається з одного шлюзу Web-API, чотирьох мікросервісів та брокера повідомлень. Упродовж життєвого циклу даного проекту, на момент проведення дослідів, вже було випущено 13 версій (*V1-V13*). Кожна версія мала технічну документацію. Було підготовлено автоматичні наскрізні тести, що покривали весь функціонал проекту та імітували реальне його використання. Моніторинг проводився за допомогою Prometheus, а журнали виконання коду зберігалися для кожного мікросервісу в окремому файлі. Граф послідовності дій для кожної версії будували автоматично за підходом [1], його середній розмір до 50 вершин та до 110 ребер.

Для кожної вершини версій *V1*…*V13* обчислено три нормовані метрики (*PE, REL, MAI*). Початкові ваги (α, β, γ, δ) установлено як (0,4 ; 0,3 ; 0,2 ; 0,1). Після аналізу дефект-логів перших чотирьох версій ваги вручну скориговано до α=0,35, β=0,35, γ=0,15, δ=0,15 критерієм була максимізація значень на топ-10 % вузлів за *R(v)*.

В таблиці 1 відображено розраховані усереднені метрики для релізів *V*5-*V*13.

Таблиця 1

Значення метрик

Версія	PE_{av}	REL_{av}	R_{av}
V5	0,65	0,5	0,55
V6	0,62	0,5	0,51
V7	0,58	0,5	0,49
V8	0,88	0,9	0,35
V9	0,85	0,92	0,31
V10	0,82	0,93	0,26
V11	0,88	0,94	0,25
V12	0,83	0,95	0,21
V13	0,86	0,96	0,19

В таблицю 1 не було додано *V*1-*V*4, адже ваги для розрахунку показника ризику було змінено на базі їх розрахунку.

На рис. 1-3 зображено графіки залежності інтегрального показника ризику від кожної розрахованої запропонованої метрики, розбиті по версіям релізу.

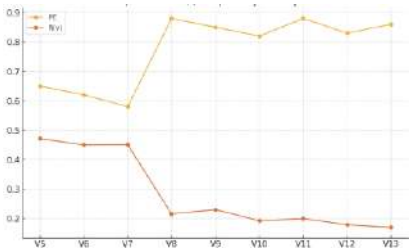

Рис. 1. Кореляція усереднених інтегрального показника ризику та складність операцій

Даний експеримент показав різке падіння ризику після *V*7. Саме в цьому релізі відбулося виправлення помилки у проекті, що створювала високе навантаження на сервер з подальшою відмовою в роботі всього проекту. В той самий час, саме дісно з кожним релізом поступово відбувалися модифікації продукту задля зменшення його навантаження на сервер.

Помилка, що була виправлена у *V*8 покращила показник R_{av} але не стосувалася змін до кодової бази, лише до розгортання самого продукту на сервері, тому жодних стрибків на графіку не відбулося

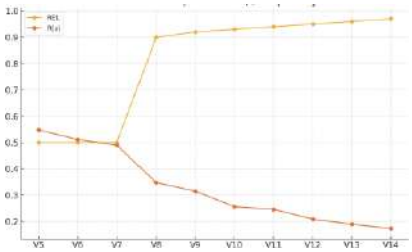

Рис. 2. Кореляція усереднених інтегрального показника ризику та безвідмовності операцій

Залежність R_{av} від REL_{av} явно показала що та сама помилка зникла після її виправлення. Це знизило кількість неуспішних операцій та інтегральний показник ризику знизився у наступному релізі.

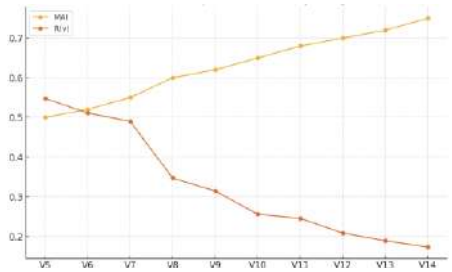

Рис. 3. Кореляція усереднених інтегрального показника ризику та безвідмовності операцій

Так як протягом усіх релізів обраного продукту робилися послідовні і незначні зміни, то такий графік є лінійним.

Висновки

Запропоновано метод обчислення інтегрального показника ризику змін у програмному продукті. Метод поєднує порівняння двох версій графової моделі, що генеруються з технічної документації, та три кількісні показники якості програмного продукту відповідно до стандарту ISO/IEC 25010: ефективність продуктивності, надійність і підтримуваність. Отриманий показник дозволяє локалізувати найбільш ризикові вузли для пріоритетного тестування, код-рев'ю та планування технічного боргу.

Література

1. Галушко Д. О. Автоматизована система налаштування продуктів для провайдерів інформаційних послуг / Галушко Д. О., Знова К. В., Ролік О. І. //. Адаптивні системи автоматичного управління. – 2025. – Т. 1, №.46. С. 178–190.

2. ISO/IEC 25010:2011. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. – Geneva: ISO, 2011.

3. Diego S. Performance Efficiency Evaluation based on ISO/IEC 25010:2011 applied to a Case Study on Load Balance and Resilient / Diego S., Lucas F. V., Jacyana S., Itamir M., Frederico A. S. // Anais do XXIV Workshop de Testes e Tolerância a Falhas. – 2023. – №.24. – P. 108-120

4. Sarwosri S.. Software Quality Measurement for Functional Suitability, Performance Efficiency, and Reliability Characteristics Using Analytical Hierarchy Process / Sarwosri S., Sit R., Umi L. Y., Sultana B. H. // JOIV International Journal on Informatics Visualization. – Dec 2023. – vol. 7, no. 4. – P. 2421-2426

5. Mena D. Maintainability and Portability Evaluationof the React Native Framework Applyingthe ISO/IEC 25010 // Mena D., Santorum M. Advances in Intelligent Systems and Computing. – Jan 2021. – vol. 1273. – P. 429–439

Accelerating the data reconciliation process in high-load distributed information systems by implementing a distributed transactional clock

Mulish Vadym, Krylov Yevhen
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
vadym.mulish@gmail.com

Abstract. **The paper proposes the concept of a distributed transactional clock to improve the data reconciliation process in high-load distributed information systems. The results of the experiments show an average reduction in data reconciliation time of 50 % compared to the systems with a non-distributed transactional clock and 66 % compared to a MongoDB Replica Set, while for critical operations reconciliation is five times faster. The proposed approach also enhances system throughput and fault tolerance.**

Keywords*: *information system, distributed database, data reconciliation, distributed transactional clock, system performance, horizontal scaling, fault tolerance.

Пришвидшення процесу узгодження даних у високонавантажених розподілених інформаційних системах шляхом впровадження розподіленого транзакційного годинника

Муліш Вадим, Крилов Євген
КПІ імені Ігоря Сікорського
Київ, Україна
vadym.mulish@gmail.com

Анотація. **У роботі запропоновано концепцію розподіленого транзакційного годинника для пришвидшення процесу узгодження даних у високонавантажених розподілених інформаційних системах. Результати експериментів демонструють скорочення тривалості процесу узгодження даних у середньому на 50 % порівняно з системами з нерозподіленим транзакційним годинником та на 66 % із MongoDB Replica Set, а для критичних операцій — час на узгодження даних є уп'ятеро менше. Запропонований підхід, водночас, підвищує пропускну здатність системи та відмовостійкість.**

Ключові слова: інформаційна система, розподілена база даних, узгодженість даних, розподілений транзакційний годинник, продуктивність системи, горизонтальне масштабування, стійкість до відмов.

Вступ

У епоху глобальної цифрової трансформації, що супроводжується стрімким зростанням трафіку, все більше і більше інформаційних систем перетворюються на високонавантажені інформаційні систем, які мають обслуговувати велику кількість одночасних запитів і робити це достатньо швидко. Адже затримка обробки клієнтських запитів у інформаційній системі, навіть, на 100 мс може знизити конверсію продукту на 7%, що безпосередньо впливає на доходи компанії та її конкурентоспроможність [1].

Головними викликами у покращенні роботи високонавантажених інформаційних систем є забезпечення необхідного рівня узгодженості даних, високої продуктивності системи, масштабованості та відмовостійкості, а також пошук ефективних методів прискорення процесу узгодження

даних, що може відігравати ключову роль у загальній швидкодії інформаційної системи.

Постановка задачі

Метою даної роботи є опис математичної моделі розподіленого транзакційного годинника та розробка підходу до пришвидшення процесу узгодження даних у високонавантажених розподілених інформаційних системах шляхом впровадження розподіленого транзакційного годинника.

Також, у рамках даної роботи передбачається проведення ряду експериментів, результати яких мають підтвердити тезу про те, що застосування розподіленого транзакційного годинника у системі дійсно пришвидшує процес узгодження даних, порівняно з іншими існуючими механізмами реплікації даних.

Математична модель розподіленого транзакційного годинника

Необхідно формалізувати математичну модель розподіленого транзакційного годинника, що працює на основі періодичної об'єднання запитів у результуючі транзакції з гарантією послідовної узгодженості.

Нехай множина вузлів, які генерують операції для розподіленого транзакційного годинника, позначаються як:

$$I = \{1, 2, \ldots, N\}, \quad (1)$$

де N – загальна кількість вузлів, що генерують операції.

Набір об'єктів даних визначається множиною:

$$D = \{d_1, d_2, \ldots\}. \quad (2)$$

Можливі типи операцій над об'єктами даних узагальнюються множиною:

$$\tau = \{Create, Update, Delete\}. \quad (3)$$

Множина операцій, що обробляє розподілений транзакційний годинник, описується виразом:

$$O = \{r = (i_r, t_r, \mu_r, d_r, \tau_r, p_r)\}, \quad (4)$$

де кожна операція *r* складається з:

- i_r – номер вузла, що її створив, $i_r \in I$, що описано в (1);
- t_r – час створення, $t_r \in R$;
- μ_r – пріоритет операції, $\mu_r \in N$;
- d_r – ідентифікатор об'єкту даних, $d_r \in D$, що описано в (2);
- τ_r – тип операції, $\tau_r \in \tau$, що описано в (3);
- p_r – корисне навантаження, що представляє об'єкт даних.

Для групування операцій у результуючі транзакції вводиться фіксована тривалість агрегаційного вікна $T > 0$. Операції, отримані в інтервалі $[kT, (k+1)T]$ належать до агрегаційного вікна з індексом *k*, тобто:

$$R_k = \{r \in O \mid t_r \in [kT, (k+1)T]\}.$$

Ці множини не перетинаються і у сукупності охоплюють усі операції з множини *O*.

Виділимо, також, підмножину операцій над конкретним об'єктом *d*:

$$R_k(d) = \{r \in R_k \mid d_r = d\}.$$

Функція перетворення

$$f(R_k(d)) = C_k(d).$$

формує результуючу транзакцію $C_k(d)$ за такими правилами:

1. Операції сортуються за пріоритетом μ; операції з найвищим пріоритетом обробляється першим.
2. Якщо серед $R_k(d)$ існує хоча б одна операція з $\tau_r = Delete$, то ця операція визначається результуючою транзакцією $C_k(d)$.
3. У випадку відсутності операції видалення, обирається операція з найбільшою часовою міткою:

$$C_k(d) = \{\arg\max_{r \in R_k(d)} t_r \mid \tau_r \neq Delete\}.$$

Оскільки результуюча транзакція завжди відповідає найновішій та найбільш актуальній операції над об'єктом, загальний хронологічний порядок операцій зберігається, що й гарантує послідовну узгодженість.

Запропонована модель розподіленого транзакційного годинника, таким чином, дозволяє ефективно пришвидшити процес узгодження даних, пріоритезувати критичні зміни та мінімізувати зайві або конфліктні оновлення, зберігаючи при цьому необхідні гарантії узгодженості даних.

Реалізація розподіленого транзакційного годинника

Порівняно з нерозподіленим рішенням, в реалізації розподіленого транзакційного годинника варто виділити такі основні покращення:

- здатність обробляти окрему частину операцій, що відносяться до певного об'єкту без накладних витрат на розподіл операцій між вузлами;
- використання для комунікації платформи багатопотокової обробки подій із гарантованою доставкою Apache Kafka замість HTTP-запитів;
- використання та оперування оптимізованими для передачі та зберігання структурами даних;
- використання реактивного підходу до обробки операцій, що зменшує затримки і ефективніше використовує ресурси.

Інтеграція розподіленого транзакційного годинника в архітектуру високонавантаженої розподіленої інформаційної системи, на відміну від нерозподіленого транзакційного годинника або вбудованого в MongoDB механізму Replica Set, перш за все спрямована на скорочення затримок і зниження обчислювальних витрат, пов'язаних із процедурою узгодження даних у розподіленому середовищі. Крім цього, розподілене рішення відкриває шлях до горизонтального масштабування, що підвищує загальну пропускну здатність системи та її стійкість до відмов.

Експериментальні дослідження

У роботі представлено результати експериментальної перевірки доцільності застосування розподіленого транзакційного годинника в ролі механізму узгодження даних у високонавантажених розподілених інформаційних системах. У межах дослідження було проведено порівняльний аналіз із нерозподіленим транзакційним годинником та класичним механізмом MongoDB Replica Set. У роботі використовуються результати з попередніх досліджень механізму нерозподіленого транзакційного годинника та його порівняння з класичним механізмом MongoDB Replica Set [2].

Для проведення експериментів була спроектована та реалізована розподілена система підтримки фінансових операцій.

Перший блок експериментів був спрямований на вимірювання часу узгодження даних у сценарії з інтенсивною взаємодією користувача з одним об'єктом даних: користувач за дуже короткий проміжок часу виконує серію операцій над власним банківським рахунком — наприклад, поповнює його, знімає кошти чи ініціює перекази.

Усі експерименти було повторено у системах підтримки фінансових операцій, що використовують 3, 5 та 7 екземплярів бази даних, а також використовують різні механізми реплікації даних між ними: Replica Set, нерозподілений транзакційний годинник та розподілений транзакційний годинник.

Результати експериментів у системі з використанням 3 екземплярів бази даних (БД) під керівництвом розподіленого, нерозподіленого транзакційних годинників та механізму Replica Set зображені на рис. 1.

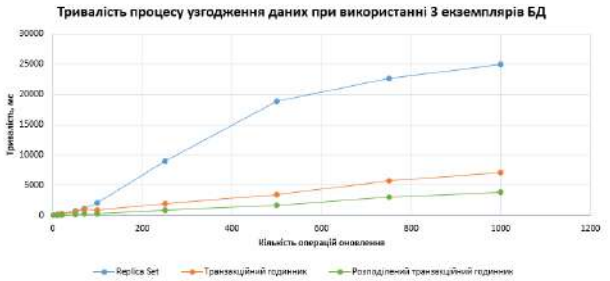

Рис. 1. Тривалість процесу узгодження даних у системах з трьома екземплярами бази даних

Результати ідентичних експериментів з використанням п'яти та семи екземплярів бази даних у різних системах показали схожі результат, тому вони не будуть наведені окремо, але підпадають під загальний аналіз результатів.

Результати показали, що високонавантажена інформаційна система, що використовує розподілений транзакційний годинник забезпечує прискорення процесу узгодження даних в середньому на 50%, порівняно з нерозподіленим транзакційним годинником, та на 66% — у порівнянні з системами, що використовують Replica Set.

Другий набір експериментів мав на меті довести здатність системи, що використовує розподілений транзакційний годинник, швидше обробляти критично важливі запити в умовах великої кількості другорядних операцій. В рамках даних експериментів критично важливою операцією є операція депозиту коштів на рахунок, а операції оновлення особистих даних користувачів є некритичними, тобто мають нижчий пріоритет.

Експерименти проводились з різною кількістю некритичних запитів, але критична операція депозиту на рахунок завжди одна.

Отримані результати у системах, що використовують механізм MongoDB Replica Set, продемонстрували значно гірші результати в обробці критичних операцій, порівняно із системами, що використовують різні варіації транзакційного годинника. Час узгодження даних у таких системах в десятки та, навіть, тисячі разів вище у порівнянні з альтернативними підходами. Це пов'язано з відсутністю механізму пріоритетності, внаслідок чого всі операції обробляються в порядку надходження.

Натомість, транзакційний годинник дозволяє задавати операціям пріоритети, що суттєво пришвидшує реплікацію результатів критичних операцій. Тому, подальший аналіз результатів було зосереджено на порівнянні розподіленого та нерозподіленого транзакційного годинника, без урахування Replica Set, що дозволило наочно виявити переваги розподіленого підходу за різної кількості вузлів у системі.

Результати аналогічних експериментів у системах, що використовують різні типи транзакційних годинників та різну кількість екземплярів розподіленої бази даних зображені на рис. 2.

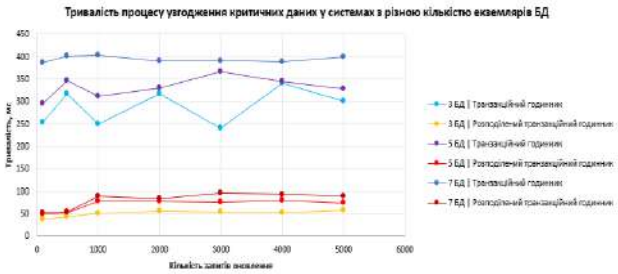

Рис. 2. Порівняння тривалості процесу узгодження критичних даних у системах, що використовують різні типи транзакційного годинника та різну кількість екземплярів бази даних

Відповідно до результатів експериментів, що зображені на рис. 2, можна стверджувати, що розподілений транзакційний годинник дозволяє у 5 разів швидше узгоджувати критичні дані порівняно з нерозподіленим аналогом. Крім того, результати демонструють високу стабільність часу необхідного на узгодження даних у системах, що використовують розподілений транзакційний годинник. При цьому, приріст затримки при збільшенні кількості вузлів розподіленої бази даних становить лише 10–20 мс, що є нижчим за аналогічний показник у системах, що використовують нерозподілений транзакційний годинник, який становить 50 мс, та в сотні разів нижчим за аналогічний показник у системах, що використовують механізм Replica Set.

Висновки

У даній роботі представлено математичну модель та концепцію впровадження розподіленого транзакційного годинника у високонавантажені розподілені системи з метою пришвидшення процесу узгодження даних. Запропонований підхід не лише зменшує час узгодження, а й покращує відмовостійкість, масштабованість і загальну продуктивність системи. Результати експериментальних досліджень продемонстрували середнє скорочення часу узгодження даних на 50%, порівняно з нерозподіленим транзакційним годинником, і на 66% — у порівнянні з Replica Set. У випадку критичних операцій, розподілений годинник забезпечив до п'яти разів вищу швидкодію, а в порівнянні з Replica Set — у сотні й тисячі разів кращу продуктивність.

Література

1. *Building Secure and Reliable Systems: Best Practices for Designing, Implementing, and Maintaining Systems.* O'Reilly Media, 2020. P. 558.
2. Mulish V., Krylov I., Nikitin V., Anikin V. Usage of a transactional clock in a distributed financial operations support system to speed up the data reconciliation process. *Adaptive Systems of Automatic Control*. 2024. Vol. 2 No. 45. P. 26–33. URL: https://doi.org/10.20535/1560-8956.45.2024.313064

Modification of the Reciprocal Rank Fusion Method to Improve Hybrid Search Results in Information Systems with Vector Databases

Mykola Bilyi, Yevhen Krylov
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
mykola.o.b@gmail.com, ekrylov1964@gmail.com

Abstract. **The paper addresses the problem of the influence of non-relevant results on the quality of hybrid search in information systems with vector databases. An improvement to the Reciprocal Rank Fusion method is proposed by grouping results according to their relevance level and applying exponential weighting. This approach reduces the impact of weak results on the overall ranking and improves search accuracy.**

Keywords**: information systems, vector database, lexical search, semantic search, hybrid search, combined search.**

Модифікація методу Reciprocal Rank Fusion для поліпшення результатів гібридного пошуку у інформаційних системах з векторними базами даних

Білий Микола, Крилов Євген
КПІ імені Ігоря Сікорського
Київ, Україна
mykola.o.b@gmail.com, ekrylov1964@gmail.com

Анотація. **У роботі розглянуто проблему впливу нерелевантних результатів на якість гібридного пошуку у інформаційних системах з векторними базами даних. Запропоновано вдосконалення методу Reciprocal Rank Fusion шляхом групування результатів за рівнем релевантності та застосування експоненціального зважування. Це дозволяє зменшити вплив слабких результатів на загальний рейтинг та покращити точність пошуку.**

Ключові слова: інформаційні системи, векторна база даних, лексичний пошук, семантичний пошук, гібридний пошук, комбінований пошук.

Вступ

Векторні бази даних - це сучасні інформаційні системи, які зберігають дані у вигляді векторів, тобто числових списків, що описують об'єкти. Кожне число у векторі - це ознака, яка відображає певну характеристику об’єкта. Наприклад, зображення яблука може бути перетворене на вектор, де одна ознака описує колір, інша - округлість, ще одна - текстуру поверхні тощо. Вектор такого зображення може виглядати так: [0,9 0,2 0,8], де 0,9 - це насиченість червоного кольору, 0,2 - округлість, 0,8 - блиск. У таких базах дані не шукаються за точним збігом, як у звичайних БД, а за схожістю між векторами. Щоб знайти потрібний об’єкт, система порівнює вектори та визначає, які з них найближчі до запиту - тобто найбільш схожі. Найпоширеніші методи порівняння - це евклідова відстань (вимірює «відстань» між точками у просторі) та косинусна схожість (визначає, наскільки вектори дивляться в одному напрямку). Наприклад, якщо вектор фото собаки схожий на вектор фото іншої собаки, то система вважає їх близькими.У реальних комерційних системах векторні бази даних використовуються для зберігання об’єктів, представлених одразу в кількох форматах. Найпоширеніший приклад - це опис товару, де поєднано текстову інформацію (назва, характеристики, опис) та зображення товару. Всі ці дані перетворюються на вектори і зберігаються в базі разом із текстовими даними, що дозволяє здійснювати гібридний (комбінований) пошук.

Зокрема, це дає можливість реалізувати лексичний пошук, семантичний пошук та пошук за зображеннями. Лексичний пошук дозволяє знаходити товари за текстовим описом. Наприклад, пошук за ключовими словами в описі товар. Семантичний пошук дозволяє знаходити товар, який контекстно схожий до запиту користувача, навіть якщо у запиті немає точно таких самих слів. Наприклад, запит "спортивні черевики для бігу" може повернути схожі товари, навіть якщо у їхніх описах використані інші терміни, як «кросівки для тренувань» або «марафонка». Пошук за зображенням дозволяє знайти товар, схожий на зображений на картинці, навіть якщо не зазначено конкретних текстових даних. Наприклад, пошук "червона футболка з емблемою футбольної команди Динамо" дозволить знайти всі схожі футболки, навіть якщо в описах не згадуються конкретні бренди чи клуби.

Опис проблеми

У гібридному пошуку виникає потреба в ефективному поєднанні результатів з різних типів пошуку з метою покращення загальної якості видачі. Reciprocal Rank Fusion (RRF) [1] - простий і ефективний метод об’єднання результатів пошуку з кількох джерел, який визначається за формулою:

$$RRF(d) = \sum_{s \in S} \frac{1}{k + rank_s(d)} \quad (1)$$

де $RRF(d)$- підсумкова оцінка документа d; S- множина пошукових стратегій; $rank_s(d)$ - позиція документа d у списку результатів пошуку s; k - постійна для згладжування (зазвичай k = 60).

Суть методу полягає в тому, що кожному результату присвоюється оцінка, обернено пропорційна його позиції в кожному списку, а потім ці оцінки підсумовуються. Завдяки цьому на перших позиціях об'єднаного списку опиняються результати, які стабільно займають високі місця в кількох пошуках, навіть якщо не є першими в жодному з них. Проблема полягає в тому, що системи ранжування результатів повертають документи в будь-якому разі, навіть якщо між запитом користувача й результатами немає жодної релевантності. Це особливо помітно у пошуку за схожістю, де система повертає найближчі сусідні документи, навіть якщо вони не мають жодного змістовного зв’язку. Наприклад, перше зображення може бути релевантним і отримає ранг 1, але друге й наступні - зовсім не відповідають запиту з точки зору людини. Однак система ранжування зображень все одно надає їм ранги 2, 3 і так далі. Якщо ми розраховуємо RRF тільки на основі цього рангу зазначивши k=60, перший документ отримає оцінку 0,01639, а другий документ отримає оцінку 0,01611, що є досить високим балом для не релевантного документа, оскільки різниця в оцінках досить мала в порівнянні з тим що перший документ релевантний а інший повністю не релевантний. І ця відносно висока оцінка може зменшити значення балів з інших типів пошуку при підсумовуванні результатів.

Розглянемо об’єднаний результат семантичного і лексичного пошуку документів згідно RRF з табл. 1.

Таблиця 1

Номер документу	Бал семантичного пошуку	Позиція згідно семантичного пошуку	Бал лексичного пошуку	Позиція згідно лексичного пошуку	RRF	Позиція згідно RRF
1	0,95	1	0,1	5	0,031778	1
2	0,94	2	0,2	4	0,031754	3
3	0,93	3	0,25	3	0,031746	5
4	0,92	4	0,29	2	0,031754	4
5	0,91	5	0,3	1	0,031778	2

Для спрощення зазначимо що мінімальний і максимальний бал який може надати лексичний і семантичний пошук від 0 до 1 відповідно.

Якщо певний тип пошуку повертає високий бал для документа, це свідчить про сильну відповідність і високу релевантність - навіть якщо інші методи пошуку присвоюють низькі бали. Бали результатів семантичного пошуку свідчать що знайдені документи «дуже сильно» відповідають запиту користувача. В той же час бали результатів лексичного пошуку свідчать що знайдені документи «погано» відповідають запиту користувача відповідно до формул лексичного пошуку, наприклад BM25 [2] або TF-IDF [3].

Відповідно до розрахованого RRF для кожного документа маємо наступний порядок документів в об’єднаному результаті пошуку: документ 1, документ 5, документ 2, документ 4, документ 3. Як бачимо порядок документів не відповідає порядку документів із семантичного пошуку оскільки слабкі оцінки із лексичного пошуку мали вплив на сильні оцінки семантичного пошуку. Документ 3 отримав останню позицію хоча він релевантніший відносно документів 4 і 5. Оцінки лексичного пошуку в даному випадку додали зайві похибки.

Ціль рішення

Рішенням вище наведеної проблеми повинно стати покращення методу RRF для запобігання впливу менш релевантних оцінок на високорелеванті оцінки.

Опис рішення

Щоб визначити які бали з лексичного і семантичного пошуку мають вплив один на одного і які можна порівнювати і враховувати у формулі RRF пропонується поділити набір результатів на різні групи. Для простоти розглянемо три групи: «хороші співпадіння, «непогані співпадіння та «погані співпадіння. Наприклад, використовуючи оцінки з попереднього прикладу, значення 0,91-0,95 належить до групи «хороших співпадінь», тоді як 0,1-0,5 до групи «погані співпадіння». Визначення груп співпадіння пропонується визначати на основі вхідного запиту користувача методом поступового зменшення токенів у запиті (деградації) і порівняння зміненого запиту із початковим запитом користувача.

Наприклад для вхідного запиту користувача «спортивні черевики для бігу» виконаємо розрахунок семантичної і лексичної схожості з рядками які утворені шляхом поступової деградації вхідного запиту. Результати

розрахунків наведені в табл. 2.

Таблиця 2

Запит	Бал семантичної подібності	Бал лексичної подібності
«спортивні черевики для бігу»	0,9547	4,9191
«спортивні черевики для»	0,912	3,9352
«спортивні черевики»	0,877	1,9676
«спортивні»	0,8204	0,9838
«»	0,7655	0,0

Як бачимо з розрахунків ми можемо виділити 4 групи наведені в табл. 3, кожна з яких має власний діапазон оцінок який дозволяє визначити до якої групи належатиме будь який знайдений документ при пошуку по заданому вхідному запиту.

Таблиця 3

Номер групи	Діапазон балів семантичної подібності	Діапазон балів лексичної подібності
1	0,9547-0,912	4,9191-3,9352
2	0,912-0,877	3,9352-1,9676
3	0,877-0,8204	1,9676-0,9838
4	0,8204-0,7655	0,9838-0

Для інтеграції інформації про групу в існуючу формулу RRF пропонується використовувати експоненціальну функцію яка дозволяє зменшувати вплив оцінок які знаходяться у віддалених групах:

$$RRF(d) = \sum_{s \in S} \frac{1}{(k + rank_s(d))^n} \quad (2)$$

де n - номер групи.

Результати впровадження

Перераховані результати з табл. 1 наведені в табл. 4. Досягнуто цілі рішення - «не достатньо хороші оцінки» з лексичного пошуку не мають вплив на порядок «дуже хороших оцінок» семантичного пошуку. Релевантний семантичний порядок результатів збережено.

Таблиця 4

Номер документу	Позиція згідно семантичного пошуку	Група семантичного пошуку	Позиція згідно лексичного пошуку	Група лексичного пошуку	RRF	Позиція згідно RRF
1	1	1	5	2	0,0166301	1
2	2	1	4	2	0,0163731	2
3	3	1	3	2	0,0161249	3
4	4	1	2	2	0,0158851	4
5	5	1	1	2	0,015653	5

Висновки

Запропонована модифікація методу RRF дозволяє покращити злиття результатів гібридного пошуку враховуючи значимість оцінок релевантності документів. Деградація вхідного запиту і визначення груп відповідно до наведених вище кроків займає мало обчислювальних ресурсів та процесорного часу оскільки дана операція виконується на сервері бази даних без використання додаткових запитів в саму базу даних та без використання додаткових запитів по мережі і тільки над вхідним запитом користувача який в середньому має невелику кількість слів. Деградація запиту користувача рекомендовано виконувати з кінця запиту оскільки більшість моделей генерування векторів генерують схожі вектори при схожому початку текстових даних. Також популярні функції для ранжування, які використовуються в лексичному пошуку, дають вищий бал співпадіння коли співпадає початок текстових даних.

Література

[1] G. V. Cormack, C. L. A. Clarke, and S. Buecher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In Proc. SIGIR. 758–759.

[2] Stephen Robertson and Hugo Zaragoza. "The Probabilistic Relevance Framework: BM25 and Beyond." Foundations and Trends® in Information Retrieval, vol. 3, no. 4, 2009, pp. 333–390.

[3] Kim, S.-W., & Gil, J.-M. (2019). Research paper classification systems based on TF-IDF and LDA schemes. *Human-centric Computing and Information Sciences*, 9(1), 30.

Mathematical Model for the Assessment of Expert Qualities

Miahkyi Mykhailo, Gavrilenko Olena
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
mishamyagkiy3@gmail.com, gelena1980@gmail.com

Abstract. **A mathematical model quantifies cryptocurrency expertise using activity in choosen social media, combining weighted sum and statistical analysis. Tested on four experts, it confirms high correlations between expertise, wealth, and social media activity, enhancing the ATAPSN algorithm. The model offers a data-driven approach for ranking experts, with applications in selecting influential figures for cryptocurrency market insights and strategic decisions.**

Keywords**:** ***expertise evaluation, cryptocurrency, social media analysis, statistical analysis, ATAPSN, expert selection, market insights.***

Математична модель оцінки експертних якостей

Мягкий Михайло, Гавриленко Олена
КПІ імені Ігоря Сікорського
м.Київ, Україна
mishamyagkiy3@gmail.com, gelena1980@gmail.com

Анотація. **Математична модель кількісно оцінює експертність у криптовалютах за активністю в обраній соціальній мережі, поєднуючи зважену суму та статистичний аналіз. Тестування на чотирьох експертах підтверджує кореляцію між експертністю, статками та активністю в соцмережах, вдосконалюючи алгоритм АУДСМ. Модель пропонує підхід для ранжування експертів, корисний для вибору впливових осіб у крипторинках.**

Ключові слова: оцінка експертності, криптовалюта, аналіз соцмереж, статистичний аналіз, АУДСМ, відбір експертів, ринкові інсайти.

Вступ

У сучасному світі, де криптовалютні ринки стрімко розвиваються, оцінка експертності фахівців набуває вирішального значення для прийняття обґрунтованих фінансових рішень. Впливові особи, такі як Ілон Маск, Майкл Сейлор чи Роджер Вер, завдяки своїй активності в соціальній мережі X можуть формувати ринкові тенденції, викликаючи значні коливання курсів криптовалют, наприклад, Bitcoin чи Dogecoin. Однак традиційні методи оцінки компетентності, такі як аналіз репутації, досвіду чи публікацій, часто є суб'єктивними та не дозволяють точно визначити рівень експертності. Це створює потребу в розробці об'єктивних, кількісних підходів, які б базувалися на чітких метриках і враховували сучасні джерела даних, зокрема соціальні мережі.

Соціальна мережа X стала унікальним середовищем, де експерти висловлюють думки, діляться прогнозами та впливають на поведінку інвесторів. Наприклад, твіт Ілона Маска про Dogecoin у 2024 році спричинив зростання курсу валюти на десятки відсотків, що підкреслює важливість аналізу активності таких осіб. Водночас відсутність формалізованих методів оцінки експертності ускладнює відбір надійних фахівців для консультацій чи прогнозування ринкових трендів. Суб'єктивні оцінки, засновані на загальній репутації, не враховують таких факторів, як частота дописів, залученість аудиторії чи фінансові досягнення, які можуть бути ключовими показниками впливовості.

Ця робота спрямована на подолання зазначених обмежень шляхом розробки математичної моделі для кількісної оцінки експертності на основі активності в соціальній мережі. Під експертністю мається на увазі числова характеристика відповідності експерта зазначеним критеріям. Модель використовує метрики, такі як кількість підписників, частота дописів про криптовалюти та рівень залученості, для створення об'єктивного рейтингу експертів. Дослідження інтегрує отримані результати в АУДСМ (алгоритм урахування впливу дописів у соціальних мережах на курс криптовалют), що дозволяє не лише оцінити експертність, а й прогнозувати вплив експертів на ринкові процеси. Такий підхід має практичне

значення для інвесторів, компаній та аналітичних платформ, які прагнуть обґрунтовано вибирати експертів для співпраці чи аналізу ринків.

Постановка задачі

Оцінка експертності фахівців у сфері криптовалют є складною задачею через переважання суб'єктивних методів, які не відповідають вимогам сучасних фінансових ринків. Традиційні підходи, що спираються на репутацію, професійний досвід чи публічну впізнаваність, не дозволяють точно кількісно оцінити компетентність експертів, чиї думки впливають на ринкові тенденції. Соціальні мережі, зокрема платформа X, стали ключовим джерелом даних про діяльність фахівців, включаючи частоту дописів, залученість аудиторії та фінансовий успіх. Однак відсутність формалізованих методів для аналізу цих даних ускладнює створення об'єктивних рейтингів експертності, що необхідні для відбору надійних консультантів і аналітиків у криптовалютній сфері.

Проблема суб'єктивності оцінки експертності має кілька вимірів. По-перше, традиційні критерії, такі як кількість публікацій чи тривалість кар'єри, не враховують впливу соціальних медіа, де активність експертів може бути важливішим показником їхньої компетентності. Наприклад, дописи в X часто формують ринкові очікування, але їхній вплив не оцінюється систематично. По-друге, відсутність єдиної системи метрик ускладнює порівняння фахівців за чіткою шкалою, що знижує ефективність їхнього ранжування. По-третє, брак інструментів для аналізу зв'язку між соціальною активністю та ринковими процесами обмежує можливості прогнозування. Ці недоліки вказують на потребу в формалізації показників експертності, які б поєднували соціальний вплив, фінансовий успіх і частоту комунікації, забезпечуючи об'єктивне оцінювання для аналітичних і інвестиційних потреб.

Метою дослідження є вдосконалення математичної моделі для кількісної оцінки експертності фахівців у сфері криптовалют, інтегрованої в АУДСМ. Задача полягає в формалізації показників експертності, шляхом застосування зваженої суми та статистичних методів. Модель має генерувати числову оцінку експертності в діапазоні від 0 до 1, що дозволить створювати об'єктивні рейтинги фахівців. Необхідно забезпечити інтеграцію вищезазначених метрик у АУДСМ для підвищення точності аналізу ринкових тенденцій. Додатково слід перевірити гіпотези про кореляцію експертності з активністю в X і фінансовими показниками, щоб оцінити надійність моделі. Запропонований підхід спрямований на створення універсального інструменту, який буде масштабованим для використання в аналітичних платформах, інвестиційних стратегіях і для відбору експертів, чиї прогнози впливають на криптовалютні ринки.

Методологія

Для оцінки експертності фахівців у сфері криптовалют використано комплексний підхід, що враховує сім критеріїв, які відображають вимоги до експертів. Ці критерії формалізовано через кількісні метрики, зібрані з профілів X та відкритих джерел, таких як Forbes, за тиждень 13–19 жовтня 2024 року, коли експерти публікували дописи про криптовалюти [1].

Критерії оцінки:

1. Публічність – вимірюється кількістю підписників у соціальній мережі, що вказує на соціальний вплив. Наприклад, більша кількість підписників корелює з ширшою аудиторією, здатною реагувати на дописи про криптовалюти;
2. Активність у соціальній мережі – оцінюється середньою кількістю дописів за тиждень. Висока активність свідчить про регулярну комунікацію, що є важливим для впливу на ринкові настрої [2];
3. Різні професійні кола – кількісно оцінено через категорію професійної діяльності (наприклад, технології, інвестиції), закодовану як бінарний показник (1 для унікальності, 0 для повторення). Це забезпечує різноманітність експертів;
4. Зв'язок із криптовалютами – вимірюється кількістю дописів про криптовалюти за тиждень, що відображає спеціалізацію в цій сфері;
5. Незалежність – оцінено як відсутність прямих взаємодій у соціальній мережі (реплаїв, ретвітів, тощо) між експертами, закодовану як бінарний показник (1 для незалежності, 0 для взаємодії);
6. Фінансовий успіх – виражено через статки (у дол. США), що є спрощеним показником кваліфікації у фінансовій сфері. Дані взяті з Forbes станом на 2024 рік;
7. Досвід у фінансових ринках – оцінено через кількість років участі в криптовалютних інвестиціях чи торгівлі, отриманих із біографій експертів.

Оцінка експертності базується на зваженій сумі, яка об'єднує нормалізовані значення критеріїв у єдиний показник E у діапазоні $[0,1]$. Нормалізація проводиться методом min-max, а для критеріїв 1 і 6 застосовується логарифмічне масштабування через нерівномірний розподіл даних (наприклад, статки від \$0.7 млрд до \$330 млрд). Зважена сума порівнюється з методом AHP (Analytic Hierarchy Process) для перевірки стабільності рейтингів [3]. Кореляційний аналіз Пірсона використовується для перевірки гіпотез про зв'язок E зі статками та активністю (за необхідністю).

Математична модель

Нехай на вхід подаються статистичні дані відповідно до відібраних експертів. На виході отримуємо рівень експертності для кожного з експертів.

Основна формула зваженої суми:

$$E_j = \sum_{i=1}^{7} \omega_i \cdot x_{ij} \quad (1)$$

де E_j – рівень експертності для експерта $j = (1..4)$, ω_i – ваги критерію i, сума котрих буде 1, x_{ij} – нормалізовані значення критерію i для експерта j. Ваги відображають пріоритетність критеріїв: публічність і активність мають більшу вагу через їхній вплив на ринки [4].

Нормалізація для критеріїв 2–5 і 7:

$$x_{ij} = \frac{y_{ij} - \min(y_{ij})}{\max(y_{ij}) - \min(y_{ij})} \quad (2)$$

де y_{ij} – значення критерію i для експерта j.

Для критеріїв 1 та 6, використаємо:

$$x_{ij} = \frac{lg(y_{ij}) - \min(lg(y_{ij}))}{\max(lg(y_{ij})) - \min(lg(y_{ij}))} \quad (3)$$

Логарифмічне масштабування зменшує диспропорцію, наприклад, між 200 млн і 0.2 млн підписників.

Для кожного експерта зі значень рівня експертності, складемо вибірку $E = (E_1..E_4)$.

Також, використовуємо кореляцію Пірсона (за необхідності):

$$r = \frac{\sum(x_{ij} - \overline{x_i})(E_j - \bar{E})}{\sqrt{\sum(x_{ij} - \overline{x_i})^2 \sum(E_j - \bar{E})^2}} \quad (4)$$

де $\overline{x_i}, \bar{E}$ – середні значення. Значення $r > 0.7$ підтверджує гіпотези (сильний кореляційний зв'язок за школою Чеддока).

Зважена сума обрана через простоту, швидкість обчислень і легкість інтеграції в АУДСМ для прогнозування ринкових тенденцій [5]. Вона дозволяє адаптувати ваги залежно від контексту, наприклад, підвищити ω_4 для аналізу криптовалютних трендів. AHP, який використовує парне порівняння критеріїв, забезпечує більшу точність при суб'єктивному виборі ваг, але вимагає складних обчислень і матриці порівнянь, що знижує його практичність для реального часу. Тестування AHP показало схожі рейтинги, але з вищими витратами часу, тому зважена сума є кращою для АУДСМ.

Зважена сума з логарифмічним масштабуванням обрана через:

1) Здатність обробляти нерівномірні дані;
2) Простота обчислень, що забезпечить необхідну швидкість для ринкового аналізу;
3) Гнучкість у налаштуванні ваг [6].

Дані з X (підписники, дописи, лайки) та Forbes (статки) забезпечують достовірність. Модель інтегрована в АУДСМ для оцінки впливу експертів на курси криптовалют.

Приклад розрахунку для умовних експертів

Для демонстрації роботи моделі використано умовні дані для умовних експертів (не реальні, лише для ілюстрації).

Відповідні діапазони значень для чотирьох експертів:

- y_{1j}: 0.2 – 200 млн, $lg(y_{1j})$: 5.3 – 8.3;
- y_{2j}: 3 – 15 дописів;
- y_{3j}: 0 – 1;
- y_{4j}: 1 – 7 дописів;
- y_{5j}: 0 – 1;
- y_{6j}: 0.7 – 330 млрд, $lg(y_{6j})$: 8.85 – 11.52;
- y_{7j}: 5 – 15 років.

Після розрахунку нормалізованих значень використовуючи формули (2) та (3) маємо наступні середні значення по кожному з критеріїв: $\overline{x_1} = 0.9, \overline{x_2} \approx 0.583, \overline{x_3} = 1, \overline{x_4} \approx 0.667, \overline{x_5} = 1, \overline{x_6} \approx 0.692, \overline{x_7} = 0.5$

В результаті, враховуючи формулу (1), отримали середнє значення експертності для обраних чотирьох експертів: $\bar{E} \approx 0.771$, що вказує на високий рівень компетентності цих умовних експертів.

Результати експерименту

Експеримент проведено для оцінки експертності чотирьох фахівців у сфері криптовалют на основі їхньої активності в соціальній мережі X за тиждень 13–19 жовтня 2024 року. Модель зваженої суми, описана в попередньому розділі, застосована до семи критеріїв: публічність, активність у X, належність до різних професійних кіл, зв'язок із криптовалютами, незалежність, фінансовий успіх, і досвід у фінансових ринках. Для демонстрації використано умовні дані, які ілюструють роботу моделі, але є лише наближенням до реальних значень. Результати підтверджують гіпотези про кореляцію експертності зі статками та активністю, а також практичну цінність моделі для АУДСМ [1].

Наближені дані для експертів:

1) Ілон Маск: 200 млн підписників, 15 дописів на тиждень, 1 (технології), 3 дописи про криптовалюти, 1 (незалежний), 330 млрд дол. США, 15 років досвіду;
2) Майкл Сейлор: 3 млн. підписників, 10 дописів на тиждень, 1 (інвестиції), 5 дописів про криптовалюти, 1 (незалежний), 5 млрд дол. США, 10 років досвіду;
3) Роджер Вер: 700 тис. підписників, 7 дописів на тиждень, 1 (криптообмін), 4 дописи про криптовалюти, 1 (незалежний), 700 млн дол. США, 12 років досвіду;
4) Тім Дрейпер: 200 тис. підписників, 3 дописи на тиждень, 1 (венчурний капітал), 1 допис про криптовалюти, 2 млрд дол. США, 8 років досвіду.

Відповідно до наведених значень маємо наступні діапазони:

1) Публічність: 0.2 – 200 млн підписників;
2) Активність: 3 – 15 дописів на тиждень;
3) Професійне коло: 1 (бінарний показник);
4) Зв'язок з криптовалютою: 1 – 5 дописів про криптовалюту за тиждень;
5) Незалежність: 1 (бінарний показник);
6) Статки: 0.7 – 330 млрд дол. США;
7) Досвід: 8 – 15 років досвіду у фінансових ринках.

Виконуючи нормалізацію значень за формулами (1) і (2), а також після розрахунку експертності отримуємо відповідні результати в табл. 1

Таблиця 1

Нормалізовані значення та рівень експертності

Експерт	**Критерії**							**Рівень експертності**
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	E
Ілон Маск	1	1	1	0.5	1	1	1	0.925
Майкл Сейлор	0.393	0.583	1	1	1	0.318	0.286	0.574
Роджер Вер	0.183	0.333	1	0.75	1	0	0.571	0.337
Тім Дрейпер	0	0	1	0	1	0.169	0	0.125

Також перевіримо гіпотезу про кореляцію E зі статками та активністю за формулою (4) і отримаємо в результаті $r = 0.92$ для статків, $r = 0.85$ для активності, що підтверджує гіпотези [2].

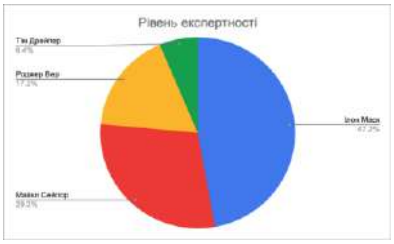

Рис. 1. Рівень експертності.

Ілон Маск має найвищий рівень експертності через лідерство в публічності та статках, тоді як Тім Дрейпер має найнижчий через низьку активність. Результати підтверджують, що модель ефективно ранжує експертів, а високі кореляції зі статками та активністю вказують на її надійність [3]. Модель інтегрована в АУДСМ для прогнозування впливу дописів на крипторинки, що корисно для інвесторів [4].

Висновки

Розроблено математичну модель для оцінки експертності фахівців у сфері криптовалют на основі їхньої активності в соціальній мережі X, що інтегрується в АУДСМ для аналізу ринкових тенденцій. Модель базується на семи критеріях: публічність, активність у X, належність до різних професійних кіл, зв'язок із криптовалютами, незалежність, фінансовий успіх і досвід у фінансових ринках. Експеримент, проведений із демонстраційними даними за тиждень 13–19 жовтня 2024 року, підтвердив ефективність моделі та її практичну цінність.

Експериментальні результати показали, що модель зваженої суми дозволяє кількісно оцінити експертність у діапазоні від 0 до 1, створюючи об'єктивні рейтинги фахівців. Для чотирьох експертів отримано рівні експертності: 0.925, 0.574, 0.337 і 0.125. Найвищий показник (0.925) досягнуто завдяки лідерству в кількості підписників (200 млн) і статках (330 млрд дол.), тоді як найнижчий (0.125) пояснюється низькою активністю (3 дописи/тиждень) і меншими фінансовими показниками (2 млрд дол.). Кореляційний аналіз підтвердив сильний зв'язок між експертністю та статками ($r = 0.92$) і активністю $r = 0.85$), що узгоджується з гіпотезами дослідження. Ці результати свідчать про здатність моделі точно ранжувати експертів, враховуючи їх рівень експертності.

Позитивні аспекти моделі включають кілька ключових моментів. По-перше, вона забезпечує об'єктивність завдяки використанню кількісних метрик, таких як кількість дописів про криптовалюти та залученість аудиторії, що усуває суб'єктивність традиційних методів оцінки, таких як репутація чи досвід. По-друге, логарифмічне масштабування для підписників і статків дозволяє коректно обробляти нерівномірні дані, забезпечуючи справедливе порівняння експертів із різними масштабами впливу. По-третє, модель є гнучкою: ваги критеріїв можна адаптувати залежно від контексту, наприклад, підвищити вагу дописів про криптовалюти для аналізу ринкових трендів. По-четверте, інтеграція в АУДСМ розширює її застосування, дозволяючи прогнозувати вплив дописів на курси криптовалют, що є цінним для інвесторів і аналітичних платформ. Нарешті, простота зваженої суми забезпечує швидкі обчислення, що важливо для реального часу в динамічних ринкових умовах.

Обмеження моделі носять загальний характер. Вони залежать від доступності точних даних із X, що може бути ускладнено через обмеження API чи приватність профілів. Також модель спирається на припущення про незалежність експертів, що важко перевірити в повному обсязі. Крім того, оцінка досвіду спрощена через використання років участі в ринках, що не враховує якісних аспектів.

Результати експерименту демонструють, що модель ефективно ідентифікує впливових експертів, чиї дописи ймовірно впливають на ринки, що підтверджує її цінність для АУДСМ. Високі показники кореляції підкреслюють надійність моделі, а її гнучкість і простота роблять її універсальним інструментом для ранжування фахівців і прогнозування ринкових тенденцій у сфері криптовалют.

Література

1. Gobet F., Campitelli G. Evaluating Expertise: A Multi-Disciplinary Review // Psychological Research. – 2007. – Vol. 71, № 2. – P. 225–241.

2. Ganis M., Kohirkar A. Social Media Analytics: Techniques and Insights for Extracting Business Value Out of Social Media. – New York: Wiley, 2018. – 352 p.

3. Expert Judgment in Risk Analysis / Ed. by T. McCaffrey. – London: Taylor & Francis, 2016. – 256 p.

4. Гавриленко О.В., Мягкий М.Ю. Дослідження впливу публікацій відомих людей на курс криптовалют // Innovations in Scientific Research: World Experience and Realities: тезіси докл. Міжнародної конф. (Рига, 10–12 квітня 2024 р.). – Рига, 2024. – С. 87–90.

5. Мягкий М.Ю., Гавриленко О.В. Інформаційна система для аналізу впливу публікацій експертів на курс криптовалютних обмінів на основі багато-агентного підходу // Інженерія програмного забезпечення і передові інформаційні технології (SoftTech-2023): тезіси докл. Міжнародної конф. (Київ, 9–11 травня 2023 р.). – Київ, 2023. – С. 67–71.

6. Мягкий М.Ю., Гавриленко О.В. Спільноти та групи в соціальних мережах як фактор впливу на курси криптовалют // Матеріали V Міжнародної науково-практичної конференції молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології (SoftTech-2023)» // Зб. наук. пр. – Київ, 2023. – С. 232–237. 2.

Methods of Analysis and Order Formation in Supplier Management

Hontar Mykhailo, Havrylenko Olena
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

Abstract. **The paper presents a mathematical model for order formation that takes into account both the cost and reliability of suppliers. A prototype of an information system has been developed to automate supplier and product assortment management**

Keywords*: *optimization problem, data analysis, decision making, ERP system, mathematical model.

Методи аналізу та формування замовлень при роботі з постачальниками продукції

Гонтар Михайло, Гавриленко Олена
КПІ імені Ігоря Сікорського
м. Київ, Україна
turictscol40@gmail.com

Анотація. **У роботі розглянуто математичну модель формування замовлень з урахуванням вартості та надійності постачальників. Реалізовано прототип інформаційної системи, що автоматизує управління постачальниками та товарним асортиментом.**

Ключові слова: задача оптимізації, аналіз даних, прийняття рішень, ERP система, математична модель.

Вступ

У сучасних умовах цифрової трансформації та глобалізації ланцюгів постачання ефективна взаємодія з постачальниками є критично важливою для стабільної та безперебійної роботи підприємств [1]. Порушення термінів постачання, невідповідність якості товарів та людський фактор під час формування замовлень призводять до фінансових втрат і збоїв у виробництві.

Для вирішення зазначених проблем усе ширше впроваджуються інформаційні системи управління закупівлями, інтегровані з ERP-рішеннями, модулями Business Intelligence, аналітичними панелями [2]. У таких системах важливою складовою стає математичне моделювання, яке дозволяє оптимізувати розподіл замовлень між постачальниками з урахуванням вартості, обмежень і надійності. Метою цієї роботи є побудова математичної моделі оптимального формування замовлень на продукцію та реалізація інформаційної системи, яка підтримує цей процес і автоматизує управління взаємодією з постачальниками.

МАТЕМАТИЧНА МОДЕЛЬ

Для оптимізації процесу формування замовлень між кількома постачальниками запропоновано модель розподілу обсягів продукції, яка дозволяє мінімізувати витрати підприємства та одночасно враховує надійність постачальників. Модель побудована у вигляді задачі лінійного програмування з двома критеріями: вартістю закупівлі та надійністю поставок.

У межах моделі розглядаються множини:

- продукції A_1, A_2,..., A_n
- постачальників P_1, P_2,..., P_m

Метою є знайти такий розподіл замовлень x_{ij}, за якого:

- буде забезпечено повне покриття потреб у продукції
- не буде перевищено постачальницьких можливостей
- буде досягнуто компроміс між мінімізацією вартості й максимізацією надійності.

Позначення:

C_{ij}— вартість одиниці продукції A_j у постачальника P_i

B_{ij} — максимально можливий обсяг постачання

$R_{ij} \in [0;1]$ — коефіцієнт надійності

x_{ij} — обсяг продукції, що замовляється у P_i

S_j — необхідний загальний обсяг продукції A_j

Модель включає дві цільові функції:

Мінімізація витрат:

$$F_{cost} = \sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij} * x_{ij}$$

Максимізація надійності постачання:

$$F_{reliability} = \sum_{i=1}^{m}\sum_{j=1}^{n} R_{ij} * x_{ij}$$

Обмеження:

покриття потреб:

$$\sum_{i=1}^{m} x_{ij} = S_j, \ \forall j$$

обмеження по потужностях:

$$x_{ij} \le B_{ij}, \qquad \forall i,j$$

невід'ємність:

$$x_{ij} \ge 0$$

Для поєднання обох критеріїв у єдину функцію використовується зважений підхід:

$$F_{total} = \alpha * F_{cost} - (1-\alpha) * F_{reliability}$$

де α ∈ [0;1] — ваговий коефіцієнт, що визначає пріоритет між ціною та надійністю [3].

- якщо α=1, оптимізація виконується лише за критерієм вартості;
- якщо α=0, ураховується лише надійність постачальників;
- якщо α=0.5, обидва критерії враховуються рівнозначно.

Програмна реалізація інформаційної системи

Програмна реалізація моделі здійснюється у вигляді веб-застосунку. На поточному етапі реалізовано базову частину інформаційної системи, яка включає:

- Модуль постачальників — створення та редагування даних про постачальників, вибір продукції (рис. 1);
- Модуль продуктів — облік асортименту на складі

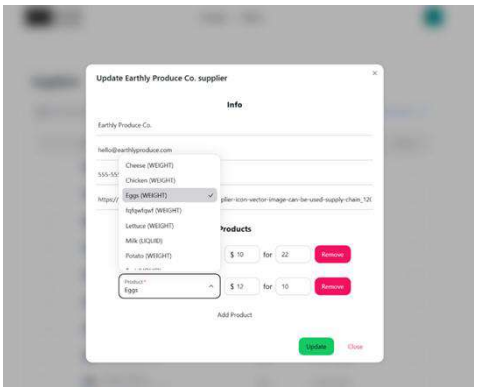

Рис. 1. Форма редагування постачальника з прив'язкою

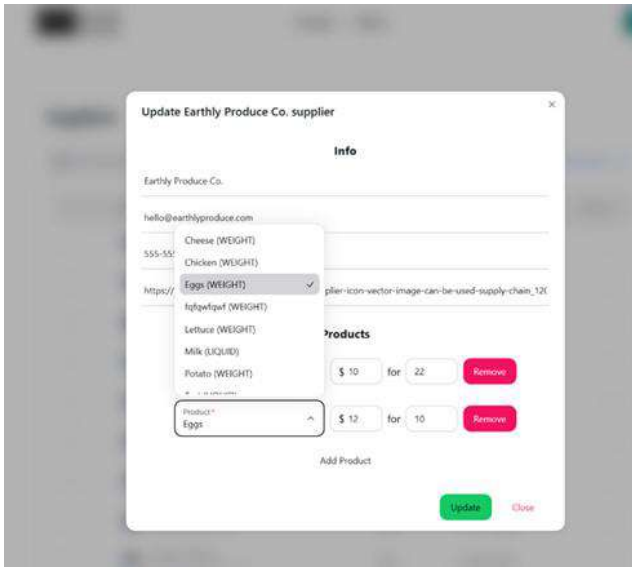

Рис. 2. Таблиця обліку товарів на складі

з вказанням одиниць виміру (рис. 2).

Інші модулі, зокрема формування замовлень, приймання товарів, контроль якості та аналітична звітність, наразі перебувають на стадії проєктування та будуть реалізовані в наступних ітераціях розробки

У планах також розробка модуля оцінки постачальників за ключовими показниками ефективності (KPI), зокрема за термінами виконання замовлень, кількістю рекламацій, стабільністю цін тощо. Окрім цього, передбачена інтеграція з ERP-системами, такими як SAP, для синхронізації даних про запаси, фінанси та логістику [4].

Висновки

Запропонована математична модель дозволяє оптимізувати процес формування замовлень із урахуванням вартості та надійності постачальників. Реалізована інформаційна система автоматизує ключові бізнес-процеси в роботі з постачальниками, підвищує оперативність прийняття рішень та знижує ризики логістичних збоїв.

Подальший розвиток роботи передбачає впровадження алгоритмів оптимізації на основі цієї моделі, використання інтелектуального аналізу даних для оцінки ризиків і повну інтеграцію з інформаційною інфраструктурою підприємства.

Література

1. Chopra S., Meindl P. Supply Chain Management: Strategy, Planning, and Operation, 7th ed. — Pearson, 2019. URL:https://www.pearson.com/en-us/subject-catalog/p/supply-chain-management/P200000000249

2. Turban E. et al. Business Intelligence: A Managerial Approach. — Pearson, 2014

3. Zavadskas E. K., Turskis Z. Multiple criteria decision making (MCDM) methods in economics: An overview. — Technological and Economic Development of Economy, 2011. URL: https://doi.org/10.3846/20294913.2011.593291.

4. SAP Procurement Overview. URL: https://help.sap.com /docs/SAP_PROCUREMENT.

Multifactor Model for Propaganda Detection in Conditions of Information Warfare

Kyryl Feshchenko, Olena Havrylenko
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

Abstract. **The paper presents a multifactor model for the detection of propaganda in the context of information warfare. The model integrates data mining techniques and machine learning algorithms to evaluate the likelihood of propagandistic content in textual data. It addresses the complexity and adaptability of modern disinformation strategies by incorporating multiple linguistic, semantic, and emotional indicators. A prototype system has been developed to automate the analysis of online content, enabling more efficient identification and filtering of potentially harmful information.**

Keywords*: *propaganda detection, information warfare, data mining, machine learning, text analysis, multifactor model.

Багатофакторна модель виявлення пропаганди в умовах інформаційної війни

Фещенко Кирил, Гавриленко Олена
КПІ імені Ігоря Сікорського
м. Київ, Україна
fkirill440@gmail.com, gelena1980@gmail.com

Анотація. **У роботі представлено багатофакторну модель виявлення пропаганди в умовах інформаційної війни. Модель поєднує методи інтелектуального аналізу даних та алгоритми машинного навчання для оцінки ймовірності пропагандистського характеру текстового контенту. Вона враховує складність і адаптивність сучасних стратегій дезінформації шляхом використання низки лінгвістичних, семантичних та емоційних ознак. Розроблено прототип системи, що автоматизує аналіз онлайн-контенту та забезпечує ефективніше виявлення і фільтрацію потенційно шкідливої інформації.**

Ключові слова: виявлення пропаганди, інформаційна війна, інтелектуальний аналіз даних, машинне навчання, аналіз тексту, багатофакторна модель.

Вступ

У світлі повномасштабної агресії Російської Федерації проти України, питання інформаційної безпеки набуло стратегічної ваги. Інформаційна війна, як складова гібридної агресії, спрямована на свідомість громадян, формування маніпулятивних наративів, деморалізацію суспільства та підрив довіри до державних інституцій.

Одним з ключових інструментів є пропаганда — систематичне поширення повідомлень, що містять емоційно забарвлені оцінки, маніпулятивні заклики, викривлені факти та тригерні формулювання[2].

Особливу загрозу становить цифрова пропаганда, що поширюється через соціальні мережі, месенджери та інші інформаційні платформи, з високою швидкістю, анонімно та безконтрольно[3]. Традиційні методи виявлення дезінформації виявилися малоефективними в умовах зростаючої складності контенту.

У зв’язку з цим запропоновано багатофакторну модель автоматизованого виявлення пропаганди, що поєднує методи інтелектуального аналізу даних (data mining) та машинного навчання, з метою оцінки ймовірності пропагандистського характеру текстового повідомлення[4].

Опис Моделі

Вхідні параметри:

I. T — текст повідомлення;

II. H — заголовок;

III. S — джерело публікації.

Вихідний параметр:

I. Total Score (TS) — інтегральна оцінка пропагандистського навантаження, що розраховується за формулою:

$$TS=i=1\sum 10wi\cdot Pi$$

де:

I. $Pi\in[0;1]$ — нормалізоване значення і-го параметра;

II. wi∈[0;1] — ваговий коефіцієнт і-го параметра.

Оцінювані параметри:

№	Позначення	Назва показника
1	P1	Сентимент-аналіз
2	P2	Тригерні слова
3	P3	Простота тексту
4	P4	Надійність джерела
5	P5	Тригерні теми
6	P6	Клікбейтний заголовок
7	P7	Суб'єктивність
8	P8	Заклик до дії
9	P9	Повторювані тези
10	P10	Повторювані тексти у джерелі

Розрахунок вагових коефіцієнтів:

$$w_i = \sum_{j=1}^{10} f_j f_i$$

де:

I. fi — частка випадків, коли Pi>0.3, серед усіх текстів у вибірці.

Це дозволяє адаптивно враховувати релевантність кожного параметру на основі емпіричних даних[1].

Інтерпретація показника Total Score:

Значення TS	Інтерпретація
< 0.1	Відсутні ознаки пропаганди
0.1–0.2	Низький рівень
0.2–0.3	Помітний рівень
0.3–0.5	Помірний рівень
0.5–0.7	Високий рівень
> 0.7	Вельми високий рівень

Значення TS ≥ 0.3 є пороговим і потребує глибшого аналізу.

Результати Експерименту

З метою апробації ефективності багатофакторної моделі виявлення пропаганди було проведено експериментальне дослідження, в рамках якого проаналізовано 10 текстових повідомлень, відібраних за принципом різноманітності пропагандистського навантаження. До вибірки увійшли як тексти, що є зразками відкритої політичної пропаганди (зокрема, публічні виступи тоталітарних лідерів), так і нейтральні за змістом матеріали науково-популярного та освітнього характеру. Оцінка проводилася шляхом обчислення інтегрального показника total score, який відображає рівень ймовірної пропагандистської насиченості тексту.

Для візуальної інтерпретації результатів з табл. 1 класифікувалися за шкалою:

Зелений: total score < 0.3 — тексти без ознак пропаганди або з незначним впливом

Жовтий: 0.3 ≤ total score < 0.5 — помірний рівень пропаганди

Червоний: total score ≥ 0.5 — високий та вельми високий рівень пропаганди

Таблиця 1

Назва повідомлення	**Total score**	**Посилання**
Звернення путіна до Генеральної Асамблеї перед оголошенням СВО	0.64	Офіційний сайт кремля
Звернення Емануеля Макрона про стратегічне бачення Європи	0.41	https://www.rfi.fr/uk
Промова Трампа під час раллі 2020 року у Тулзі	0.63	https://www.rev.com/transcripts/donald-trump-tulsa-oklahoma-rally-speech-transcript
Промова Трампа під час раллі 2020 року перед закінченням підрахунку голосів	0.78	https://www.rev.com/blog/transcripts/donald-trump-speech-save-america-rally-transcript-january-6
Звернення Гітлера до нації (1 вересня 1939 року)	0.61	https://alphahistory.com/nazigermany/hitler-declaration-of-war-against-poland-1939
Промова Гітлера на Спортпалаці	0.68	https://research.calvin.edu/german-propaganda-archive/hitler1.html
Введення до лійної геометрії	0.24	https://www.khanacademy.org/math/geometry/hs-geo-congruence/hs-geo-linear-segments/a/geometry-introduction-to-lines
Магія математики	0.2	https://www.networkpages.nl/the-magic-of-mathematics/
Дослідження океанів	0.23	https://www.noaa.gov/education/resource-collections/ocean-coasts/ocean-exploration
Вирощування яблук	0.25	https://www.nifa.usda.gov/topic/apple-production

Аналіз даних з табл 1. показав чітку кореляцію між тематичним змістом текстів і рівнем виявленого пропагандистського впливу. Так, усі повідомлення, що містять політичну риторику з елементами ідеологічного навантаження (виступи Путіна, Трампа, Гітлера), були класифіковані як високопропагандистські (червона зона), що підтверджується високими показниками total score у межах від 0.61 до 0.78. Це свідчить про здатність моделі

фіксувати маніпулятивні структури, такі як апеляція до страху, нав'язування дихотомій, використання категоричних суджень та інші риторичні прийоми, характерні для пропаганди.

Водночас тексти з науково-популярних ресурсів, присвячені темам освіти, досліджень і побутової інформації (наприклад, «Магія математики» чи «Вирощування яблук»), стабільно потрапляють у зелену зону — їхні показники не перевищують 0.25. Це демонструє здатність моделі ігнорувати інформацію, яка не несе ідеологічного чи маніпулятивного підтексту, що важливо для зменшення кількості хибнопозитивних результатів.

Особливу увагу заслуговують тексти з жовтої зони (наприклад, виступ Макрона), які відзначаються помірним рівнем total score. Вони, як правило, містять елементи стратегічного чи політичного дискурсу, але подаються в більш нейтральному та раціоналізованому тоні. Це свідчить про чутливість моделі до відтінків політичної комунікації, дозволяючи не лише виявляти відверту пропаганду, а й диференціювати проміжні випадки.

Таким чином, результати підтверджують здатність моделі не лише правильно ідентифікувати високопропагандистські тексти, а й ефективно розрізняти ступінь пропагандистської насиченості повідомлень. Це відкриває можливості для її використання у системах попереднього фільтрування контенту в інформаційних просторах, де важливо зберігати баланс між виявленням шкідливого впливу та збереженням свободи слова.

Висновки

Результати проведеного експерименту переконливо демонструють здатність запропонованої багатофакторної моделі з високим ступенем достовірності диференціювати тексти за рівнем пропагандистського впливу. Аналіз показав, що модель ефективно розпізнає збільшену концентрацію пропагандистських елементів у промовах політичних лідерів, які часто містять ознаки маніпулятивного дискурсу та цілеспрямованого впливу на аудиторію. Водночас система демонструє стабільно низькі показники total score при аналізі нейтральних науково-популярних текстів, що свідчить про її здатність уникати хибнопозитивних результатів. Це вказує на високий рівень точності класифікації та релевантності визначених критеріїв оцінювання.

Застосування динамічного алгоритму вагових коефіцієнтів забезпечує виняткову гнучкість системи, дозволяючи їй адаптуватися до широкого спектра стилістичних, жанрових та тематичних особливостей текстів. Такий підхід значною мірою підвищує ефективність роботи моделі в умовах реального інформаційного середовища, де характер повідомлень може варіюватися залежно від джерела, контексту та цільової аудиторії. Отримані емпіричні дані підтверджують не лише наукову обґрунтованість та логічну стрункість запропонованої моделі, але й її практичну придатність до попередньої автоматизованої фільтрації контенту з ознаками пропаганди. Це відкриває перспективи її подальшого розвитку, вдосконалення точності та інтеграції у комплексні системи інформаційної безпеки в умовах гібридної війни та поширення дезінформації.

Література

1. Гавриленко, О., & Фещенко, К. (2025). *Виявлення пропаганди в потоках новин*. Адаптивні Системи Автоматичного Управління 1(46), [164-174]. https://doi.org/10.20535/1560-8956.46.2025.323759

2. Черниш, Н. О., & Мусієнко, І. О. (2021). *Інформаційна війна та пропаганда в умовах гібридної агресії проти України*. Науковий вісник Херсонського державного університету. Серія: Політологія, 2(15), 102–109.

3. Barro, R. J., & Zussman, A. (2022). *Propaganda and manipulation in political discourse: Theory and evidence*. Journal of Political Economy, 130(3), 677–712. https://doi.org/10.1086/717203

4. Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., & Stein, B. (2018). *A stylometric inquiry into hyperpartisan and fake news*. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (pp. 231–240). https://doi.org/10.18653/v1/P18-1022

System for Short-Term Job Search

Rekechynska Liubov, Zhurakovska Oksana
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
bubochka102003@gmail.com

Abstract. The paper considers a system for short-term job search, which aims to improve the interaction between employees seeking temporary employment and employers. It is proposed to use the ranking of vacancies to personalize the search using a weighted convolution of criteria. An interactive map has been added to visualize job offers, which allows users to plan their time more efficiently and keep track of vacancies even within their own neighborhood.

Keywords: short-term job, job search, ranking, interactive map, Multiple Attribute Decision Making Methods.

INTRODUCTION

The demand for short-term employment is steadily increasing in today's socio-economic conditions [1], both among workers and employers. There are several reasons for this trend. Primarily, individuals seek to maintain control over their work schedules. They prefer flexible hours that enable them to balance employment with other activities. This category includes students who need to combine work with studies, individuals who have lost their primary source of income and are seeking new opportunities, and those who require an additional income stream, among others.

On the other hand, employers are often in need of quickly filling positions for temporary tasks. Services such as repair assistance, event coordination, or seasonal agricultural work are among the most common areas of short-term employment [2].

Most existing job search platforms primarily focus on long-term employment. This creates a gap for users interested in quickly and conveniently finding short-term jobs that meet their personal preferences, and for employers who need to hire workers for temporary tasks within a short time frame.

This paper presents the concept of the system for short-term job search, integrated with geolocation functionality, which allows job seekers and employers to interact efficiently. A key feature of the system is a vacancy map that visualizes available jobs based on geographic proximity, which is a crucial aspect of job accessibility [3]. Additionally, job listings are ranked according to specific criteria to personalize and accelerate the job search process.

REVIEW OF EXISTING SOLUTIONS

There are several platforms that allow users to browse job listings and submit their resumes so that employers can review candidates. Among the most popular platforms in Ukraine are Rabota.ua and Work.ua [4]. These platforms have a similar structure and functionality. They allow users to filter job postings or candidates based on various criteria such as employment type, education level, experience, expected salary, and more. While convenient for many users, these platforms do not specialize in short-term employment. Even when search queries such as "part-time job" or "short-term work" are entered, the search results primarily return long-term job offers.

In Ukraine, there are also platforms focused on service-based task execution. Based on a search and analysis of available services, the platform Kabanchik.ua appears to best align with the needs of short-term employment. This platform enables employers to post various types of temporary job categories, and workers can respond to those offers. Each job posting includes essential information to help candidates determine whether the job is suitable for them.

However, Kabanchik.ua also has certain limitations. One significant drawback is the lack of convenient categorization and ranking mechanisms. For job seekers, it is important to be able to filter vacancies based on personal preferences and relevant search criteria. Additionally, in the context of short-term employment, it is crucial to have the option to choose not only the city but also the specific district or neighborhood where the work is located. This functionality is essential for effective time management when completing short-term tasks.

Based on the analysis of current solutions and identified limitations, the goal of the system for short-term job search is to improve the process of finding short-term employment by providing ranking based on multiple criteria and incorporating a vacancy map that displays job locations. This allows users to find jobs within their local area or even within walking distance, depending on their preferences.

SYSTEM STRUCTURE

The system for short-term job search is implemented as a web-based platform. This provides convenient access from various user devices, whether a laptop, smartphone, or any other gadget ensuring that users can connect to the platform regardless of their hardware. Furthermore, this format enables interaction between the two main user categories: workers and employers.

The system architecture follows a client-server model. JavaScript was selected as the primary development language due to its universality in creating both client-side and server-side components. The backend logic is handled by the Node.js platform. For data storage, MongoDB is used as the database management system. As a non-relational database, MongoDB offers flexible and convenient storage for data of various formats and structures.

Fig. 1 presents the general architecture scheme of the system.

The first element users interact with is the registration and login page. If a person wishes to respond to a job posting or create one, they must be registered in the system. Registration requires a phone number and an email address to ensure verification and account security.

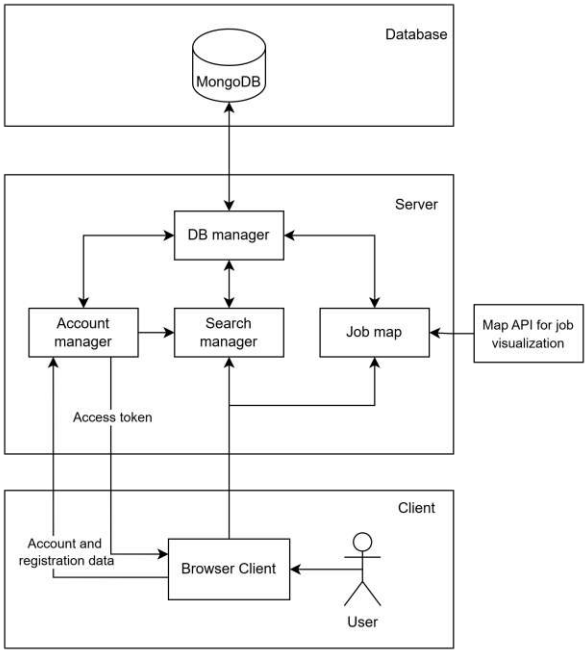

Fig. 1. The general architecture scheme of the system

A key component of the system is the user profile. This is a personalized space for any registered individual. Here, users can edit their personal information, such as phone number or geolocation. They can upload their resume to allow employers to assess how well a candidate matches the job requirements. Additionally, users may add further information about themselves or their company to better describe their experience, skills, or currently available vacancies.

One of the core features of the user profile is the ability to select ranking criteria for job listings. This functionality aims to provide the most relevant job opportunities tailored to each user, improving both the efficiency and personalization of the search process.

Several multi-criteria decision-making methods may be applied to the job ranking process, including VIKOR, TOPSIS, and the weighted sum model [5, 6]. Each method has its own advantages.

After analyzing these methods, the weighted sum model was chosen for vacant ranking due to its ease of integration into systems with a large number of listings.

The Weighted Sum Method is based on calculating a weighted sum of scores across specific criteria that help assess the attractiveness of each job offer. This approach implies that higher scores correspond to more favorable options, making it suitable for multi-criteria decision-making. Below are examples of key evaluation criteria:

- wage, assessed as the total compensation offered for completing a task. The higher the wage, the more attractive the job offer becomes;
- location convenience, which reflects the proximity of the job. It is calculated as an inverse value of the distance that needs to be covered to reach the job site;
- experience match, which is evaluated based on the number of relevant skills that align with the task requirements;
- employer reputation, determined by the average rating received after task completion. A higher score indicates greater reliability and integrity of the employer.

The overall score is calculated using the formula:

$$S = w_1 \times x_1 + w_2 \times x_2 + \ldots + w_n \times x_n \quad (1)$$

where w_i – the weight of the *i-th* criterion, and x_i – the score for that criterion.

This approach makes it possible to generate a personalized list of vacancies that are best suited to the individual.

The system supports two approaches for determining the weight values of criteria:

- automatic weight assignment, based on the information provided in the user's profile. For example, if the user indicates a preference for jobs near their location, the system prioritizes distance when ranking vacancies;
- personalized weight customization, allowing the user to manually assign priority levels to each criterion through an intuitive and user-friendly interface.

Based on the weighted sum model, a search module was developed. While using it, the user can view a personalized selection of job listings. Filters are also available to accelerate the search process.

While browsing, workers can view employer ratings and, if needed, read reviews from previous employees. The rating system is updated after each task is completed. Both parties, employer and worker have the option to leave comments and assign ratings. These ratings influence future ranking and build trust in user profiles.

To simplify and visualize job listings, an interactive map is integrated into the system, displaying all currently active job offers.

When creating a job post, the employer specifies the task location, and the system automatically places a marker on the map. This feature enables users to monitor local opportunities within their district and optimize their time by minimizing travel between tasks.

Once a user selects a job they want to apply for, the employer receives a list of interested applicants. The employer can review their ratings and feedback. After selecting a candidate, a task confirmation form is generated, capturing key parameters such as deadlines, address, and other essential details.

For illustrative purposes, a diagram representing the system components has been developed and is shown in Fig. 2.

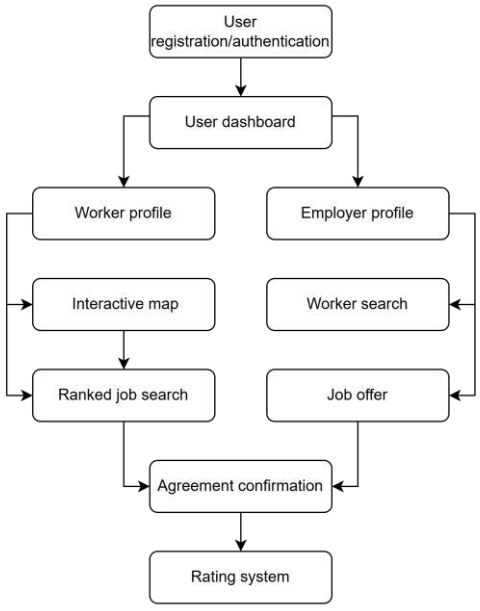

Fig. 2. Diagram of relationships between components

Thus, the structure of the System for Short-Term Job Search is designed to maximize user convenience and provides an intuitive step-by-step experience for both workers and employers.

Conclusions

The developed system for short-term job search facilitates the process of finding temporary employment for workers. At the same time, it enables employers to quickly identify qualified candidates to carry out specific tasks. A high number of potential users is expected, as the platform significantly simplifies the job search process, allowing individuals to find side jobs with minimal effort and time investment.

The architecture and core components of the system have been reviewed, along with a description of their interactions. The use of the Weighted Sum Model for job ranking increases the relevance of search results. Personalized search functionality helps users reduce the time spent finding suitable vacancies, which is crucial in the short-term employment sector. Additionally, the integration of an interactive map allows users to visualize available job opportunities, especially beneficial for those looking for work within their neighborhood or walking distance.

The system has promising potential not only at the city or national level but also for broader application. In the future, functional expansion is possible. One of the key enhancements could include the integration of AI-based modules that recommend job listings based on a user's interaction history and behavior within the system.

References

1. Lorraine Charles, Shuting Xia, Adam P. Coutts. Digitalization and Employment. A Review. ILO. 2022. URL: https://www.ilo.org/employment/Whatwedo/Publications/WCMS_854353/lang--en/index.htm
2. Non-standard employment around the world. International Labour Office - Geneva: ILO, 2016. URL: https://www.ilo.org/sites/default/files/wcmsp5/groups/public/@dgreports/@dcomm/@publ/documents/publication/wcms_534496.pdf
3. The Impact Of Geolocation On Job Search Algorithms. URL: https://medium.com/@alliancejobsin/the-impact-of-geolocation-on-job-search-algorithms-7b36ae7c7818 (Accessed on: 26.04.2025)
4. Unian.ua. economics. URL: https://www.unian.ua/economics/other/reyting-saytiv-z-robotoyu-v-ukrajini-12253731.html (Accessed on: 28.04.2025).
5. Alinezhad, A., Khalili, J. New Methods and Applications in Multiple At-tribute Decision Making (MADM).-Springer, 2019.-233p.
6. Papathanasiou, J., Ploskas, N. Multiple Criteria Decision Aid. Methods, Ex-amples and Python Implementations.-Springer, 2018.-173p.

Information system for developing listening comprehension skills in foreign language learning

Lytovchenko Alina, Zhurakovska Oksana
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
alina.lytovchenko20@gmail.com, o.zhurakovska@kpi.ua

Abstract. **This paper presents an intelligent system for improving listening comprehension in foreign language learning. It converts user-uploaded English texts into synchronized audio and subtitles, with instant translation support. To reduce delays, a dynamic programming algorithm optimizes parallel TTS requests. Experiments show that the method significantly speeds up audio generation, aiding especially Ukrainian users adapting to new language environments.**

Keywords: Text-to-Speech, Interactive Learning, Speech Synthesis, Artificial Intelligence, Automatic Translation, Listening Skills, Audiobook Generation, Dynamic Programing Method.

INTRODUCTION

In the era of rapid development of information technology, the ability to effectively develop foreign language listening skills is becoming important for both personal and professional development. Listening is one of the most complex competencies, the development of which requires regular training, considerable time and high-quality educational resources. The introduction of artificial intelligence methods into the learning process can increase its effectiveness, including by creating adaptive and interactive audio content that meets the needs of each user.

The relevance of this work is determined by the acute need of Ukrainians for effective means of language adaptation, especially in connection with forced relocation due to military operations. The developed information system ensures the organization of the development of listening skills in the learning process by automatically creating high-quality audio accompaniment to textual materials in English, synchronizing audio with the visual display of the text, and an integrated translation tool, which greatly facilitates learning and helps to overcome the language barrier.

Structure of the Information System for Developing Listening Comprehension Skills in Foreign Language Learning

The field of research of this paper is the process of developing listening skills in learning a foreign language and methods of its improvement. The developed system provides automatic generation of audio accompaniment to any English-language texts uploaded by users, provides synchronous text highlighting while listening to it, and allows instant translation of any selected text fragments.

The structural diagram of the information system illustrates the interaction between its main components and the technologies that ensure its full functionality. The structural diagram is shown in Figure 1.

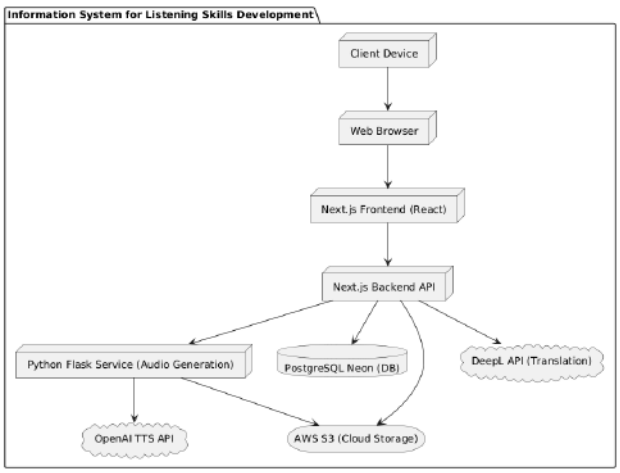

Fig. 1. Structural Diagram of the Information System for Developing Listening Comprehension Skills in Foreign Language Learning

At the first stage, the user interacts with the system via a client device (computer, smartphone, or tablet). The user's device runs a web browser that provides access to the system's web interface.

The system interface is implemented using React technologies and the Next.js framework, which form the front-end application. This component is responsible for presenting information and transmitting user requests to the backend server.

The backend part of the system is also built on the basis of Next.js and processes all received requests by interacting with the following components:

- AWS S3 cloud file storage that stores book files, generated audio files, subtitles (SRT files), and book covers;
- a Flask-based Python service responsible for generating audio files for text materials. This service accesses the OpenAI TTS API, which provides high-quality and natural text-to-speech;

- a cloud-based PostgreSQL Neon database that stores information about users, books, reading progress, and other necessary metadata;
- DeepL API as an external translation service used to instantly translate text fragments from English into Ukrainian in real time.

This architecture ensures stable, scalable and efficient operation of the developed information system.

Problem Statement

The goal of the development is to ensure minimal delay in generating the voiceover of the English text uploaded by the user. An important requirement for the system's operation is the almost instantaneous start of audio playback and the receipt of synchronized subtitles immediately after the text file is uploaded. For this purpose, the Text-to-Speech technology is used with the use of streaming text processing in parts. It is planned to use the OpenAI speech synthesis API, which supports the transmission of audio in fragments, allowing you to start playback before the processing of the entire text is complete.

The task is formulated as follows:

— the user uploads the text of a book consisting of a sequence of paragraphs;

— the text is divided into a set of fragments, each of which corresponds to one paragraph;

— fragments are grouped into packages, each package containing one or more paragraphs;

— the packets are processed sequentially, and the fragments within each packet are processed in parallel through parallel requests to the TTS API. The number of simultaneous parallel requests within a single packet is limited by a set limit;

— the generation time of audio and subtitles for each package is defined as the maximum generation time of one paragraph included in the package;

— the total generation time is equal to the sum of the generation time for all packets.

The optimization task is to find a grouping of paragraphs into batches that minimizes the total time for generating audio and subtitles. The packages should not have any common paragraphs.

The criterion for system efficiency is the minimum time from the moment the text is loaded to the start of audio playback and the total time for generating the entire text.

The proposed solution will allow the user to start listening to the book almost immediately after downloading it, while simultaneously receiving accurate audio in MP3 format and synchronized subtitles in SRT format.

Formally, the input data of the problem is defined as follows:

The input data of the problem are:

T – the text of the book downloaded by the user;

p_i – i-th paragraph of the book, $p_i \in T$, $i = \overline{1, N}$, where $N = |T|$;

P – the maximum number of simultaneous requests;

F – the set of fragments formed from T after splitting the book into paragraphs: $F = \{ p_1, p_2, \ldots, p_N \}$, $|F| \le P$;

G – the set of packages for parallel processing:

$G = \{ g_1, g_2, \ldots, g_K \}$, where each $g_k \subset F$, $k = \overline{1, K}$, K – amount of packages;

S – a set of sequential requests to the TTS API:

$S = \{ s_1, s_2, \ldots, s_P \}$;

$t(g_k)$ - the time of audio and subtitle generation for the packet g_k, $g_k = max\,(t(p_{ik}))$;

$\overline{t}$ – total generation time, $\overline{t} = \sum_{k=1}^{K} t(g_k)$.

Thus, the mathematical formulation of the problem is to find a structure for splitting the set F into K packets (where $K \le P$) with the number of fragments in each packet $|g_k|$, minimizing the value of $\overline{t}$, i.e:

$$\overline{t} = \sum_{k=1}^{K} t(g_k) \to \min,$$

under the conditions of: $\bigcup g_k = F$, $g_k \cap g_1 = \emptyset$ at $k \neq 1$.

The proposed solution will allow the user to start listening to the book almost immediately after downloading it, while receiving accurate audio in MP3 format and synchronized subtitles in SRT format.

Justification of the Solution Method

The easiest way to solve this problem is to make consecutive requests to the OpenAI TTS API, where each request returns audio for text no longer than 4096 characters (the maximum limit of one request). However, this approach can be quite time-consuming, so there is a need to find a more efficient solution.

In modern research, various approaches are widely used to solve optimization problems: evolutionary algorithms [6], multi-agent methods (particle swarm algorithms [5], ant colonies [4]), multi-criteria decision-making methods [3], and dynamic programming algorithms [2].

Evolutionary algorithms imitate the processes of natural evolution and genetics, allowing to obtain approximate solutions to complex problems. Their main advantage is flexibility in customization, but they require complex parameter calibration [6].

Optimization by ant colonies and particle swarms is based on modeling the behavior of natural systems and is effective in large search spaces. The disadvantages of these methods are high computational costs and complexity of setup [4].

Multi-criteria decision-making methods are used when a task has several optimization criteria, which allows finding

compromise solutions, but makes it difficult to find the single best option [3].

Dynamic programming (DP) algorithms solve a problem by breaking it down into smaller subproblems for which optimal solutions are found. They are particularly effective in problems that have an optimal substructure and overlapping subproblems. These characteristics make a DP the best choice for our task, which is to split text into packets while minimizing the time to generate audio [2].

Given this, the dynamic programming algorithm was chosen to solve the problem because of its ability:

- consistently and in detail consider all possible options for splitting paragraphs into packages without missing any potentially optimal solutions;
- to find the exact solution without additional complex settings;
- is easy to adapt to the real limitations of parallel processing.

To solve the problem, we use a dynamic programming algorithm that consists of the following steps:

Step 1. Formation of the input data structure.

At the initial stage, a list of paragraphs is generated. For each paragraph, a pre-calculated synthesis time is determined.

Also, the parallel processing limit k is determined - the maximum number of paragraphs that can be simultaneously synthesized within one package (corresponds to the number of simultaneous requests to the TTS API).

Step 2. Splitting into subtasks using the dynamic programming method.

The algorithm checks different ways of splitting the text into packets and looks for the one with the lowest total time. To do this, we introduce the variable $dp[i]$, which means "the minimum possible time to read the first i paragraphs of the book."

To find $dp[i]$, all possible options for building the latest package are considered (for example, consisting of one paragraph p_i or 2 paragraphs p_{i-1} and p_i, and so on). Then, we determine the voice time of this package, which is added to the voice time of the previous paragraphs. Choose the minimum result among all the options.

Step 3. Calculating the final result.

After sequentially calculating $dp[1]$, $dp[2]$,..., $dp[n]$ the last value obtained $dp[n]$ will be the minimum sounding time for the entire book.

Analysis of the Obtained Results

Let's conduct a series of experiments to compare the speed of the basic algorithm, in which audio is generated sequentially (request after request, each containing no more than 4096 characters), with the proposed method, which uses the optimal distribution of text into packets for parallel audio generation.

A comparison of the results is shown in the graph in Figure 2.

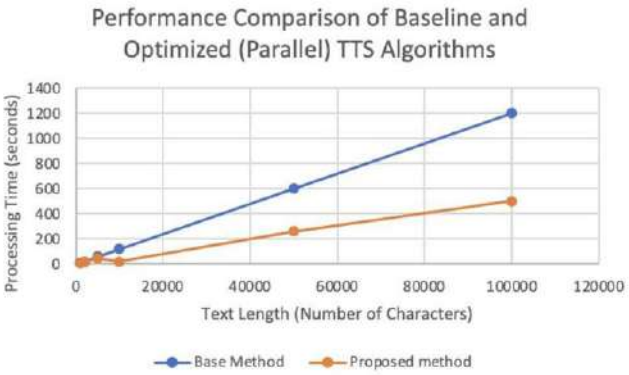

Fig. 2. Performance Comparison of Baseline and Optimized (Parallel) TTS Algorithms

The analysis of the results allows us to conclude that the proposed optimized method is significantly superior to the baseline method, where audio is generated sequentially, request by request every 4096 characters. The baseline method shows a significant increase in generation time with increasing text size, as each new packet waits for the previous one to complete.

The proposed method, which uses a dynamic programming algorithm to efficiently divide the text into packets and process queries in parallel, demonstrates a significant reduction in overall processing time. Even with large amounts of text, this method can reduce the total generation time by almost three times compared to the basic approach, which indicates its high efficiency and practical value.

CONCLUSION

The experiments confirmed the effectiveness of the proposed method of audio generation using the dynamic programming algorithm. Compared to the basic method, where audio is generated by sequential requests of 4096 characters, the optimized approach allowed us to reduce the total processing time by almost three times.

Thus, the use of dynamic programming significantly improves system performance and reduces delays when users access audio materials.

REFERENCES

[1] Pinedo. M. Scheduling Theory, Algorithms, and Systems. New York: Springer, 2012. 69-109 p.

[2] Roughgarden T. Roughgarden Algorithms Illuminated (Part 3): Greedy Algorithms and Dynamic Programming. New York: Soundlikeyourself Publishing, 2019. 1-21 p.

[3] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms. Chichester, UK: Wiley, 2001.

[4] M. Dorigo and T. Stützle, Ant Colony Optimization. Cambridge, MA: MIT Press, 2004.

[5] J. Kennedy and R.C. Eberhart, Swarm Intelligence. San Diego: Morgan Kaufmann, 2001.

[6] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley, 1989.

Information and Analytical System for UAV Routing

Kosobutsky Andrey, Zhurakovska Oksana
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
andrey15112003@gmail.com, o.zhurakovska@kpi.ua

Abstract. **This article discusses the concept of an information and analytical system for UAV routing, focused on ecological monitoring of forest areas. The system automatically generates the optimal flight route based on geospatial, topographic, meteorological data and drone characteristics. A hybrid method combining a territory coverage algorithm and a modified ant colony algorithm is applied. A convenient interface is provided for mission configuration and visualization.**

Keywords: UAV routing, environmental monitoring, territory coverage, ant algorithm, information system, decision-making methods.

INTRODUCTION

In today's environment, when climate change is spreading, forests are degrading, and forest fires are becoming more frequent, ecological monitoring of forests is becoming a matter of paramount importance. Unmanned aerial vehicles (UAVs) are highly effective in monitoring the state of natural ecosystems, as they allow obtaining high-precision images and sensor data from hard-to-reach areas with minimal time and resources. However, the effective use of UAVs directly depends on the quality of the route planning along which the territory is flown.

The process of planning flights for such tasks should take into account a whole range of factors: the spatial structure of the terrain, the presence of flight restriction zones, battery life, weather conditions, and the need for complete coverage of a given area. In the practice of environmental monitoring, the need to systematically fly over large areas without gaps causes particular difficulties, which requires optimization of the route by criteria, mission time, and flight safety. Some of the first works on coverage path planning rely on heuristics, which are simple rules of thumb that can work well, but lack evidence-based guarantees that guarantee the success of the coverage operation [1].

In view of this, it is important to create an information and analytical system that will ensure the automated generation of UAV routes, taking into account all technical, geographical and environmental constraints. Such a system should not only generate efficient routes, but also provide the user with a convenient interface for selecting the overflight area, setting flight parameters, viewing the generated route, and further processing the data obtained.

This paper proposes the concept of such a system, which is based on a combination of the classical Coverage Path Planning approach with metaheuristic optimization methods, in particular, the ant algorithm. Hybridization allows not only to guarantee full coverage of the territory, but also to minimize the time and energy consumption for overflights, which is critical when monitoring large or hard-to-reach forest areas.

SYSTEM ARCHITECTURE

The information-analytical system for routing unmanned aerial vehicles for environmental monitoring of forests has a modular architecture that provides flexibility, scalability, and ease of adaptation to different types of missions. The system is based on the processing of cartographic, terrain and meteorological data, followed by the construction of an optimal

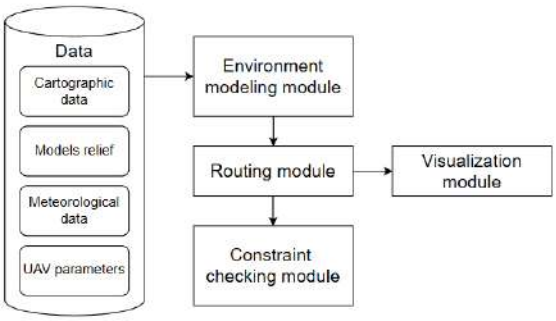

Fig. 1. Block diagram of the system

flight route, taking into account all constraints. The block diagram of the system is shown in Fig. 1.

The system consists of several functional modules. In the first stage, the data collection module receives basic information from open sources such as OpenStreetMap, digital elevation models (SRTM), and meteorological services via API. OpenStreetMap was created by a community of cartographers who add and maintain data on roads, trails, cafes, train stations, and many other objects around the world [2]. The received data is processed by the environment modeling module, which forms an internal graphical model of the terrain, dividing the territory into coverage areas, taking into account terrain features, natural and regulatory restrictions.

The next stage involves the routing module, which implements a hybrid algorithm that combines the classical method of territory coverage with a modified ant optimization algorithm. Ant algorithms have proven to be a very effective approach for routing problems, for example, in telecommunication networks, where system properties, such as channel usage cost or node availability, change over time [3]. First, full coverage of the area is ensured, and then the order of overflight of the areas is optimized to minimize energy consumption and flight time. When building a route, the system takes into account flight time restrictions, maximum safe altitudes, weather conditions, and restricted areas.

After the route is built, the constraint checker automatically validates the route according to the technical characteristics of the selected UAV and applicable regulatory standards. The visualization module then allows the user to view the overflight area and route on an interactive map, edit the route manually if necessary, and evaluate the overall mission performance.

PROBLEMS

The use of unmanned aerial vehicles (UAVs) for environmental monitoring of forests is accompanied by a number of technical and organizational challenges that make it difficult to plan routes effectively.

One of the key issues is the need to ensure complete coverage of a given area without gaps and unnecessary duplication. Standard terrain scanning strategies do not always take into account the complexity of the actual terrain or other areas, as well as the presence of natural and anthropogenic obstacles, which leads to the formation of "dead zones" or excessive resource costs for repeated overflights.

An additional challenge is the limited resources of UAVs, in particular, the autonomous flight time due to battery capacity and payload weight characteristics. This necessitates optimization of routes in such a way as to ensure full coverage of the territory within the available energy reserve of the vehicle.

The task is also complicated by natural conditions, such as significant altitude differences, difficult terrain, dense forests, and the presence of water bodies or gorges. In such conditions, there is a need for dynamic flight altitude planning, avoidance of risk zones, and route adaptation to the changing landscape. In turn, weather factors such as wind loads, precipitation, and temperature changes directly affect energy consumption, flight stability, and data collection quality, which requires the integration of weather data into the routing process.

Regulatory restrictions, which include bans on flights over certain objects, flight altitude restrictions, and the need to obtain special permits, pose a separate problem. When building routes, such restrictions should be taken into account automatically to avoid violating the law. Another important requirement is the scalability and adaptability of the system: it must work effectively both in small areas and large areas, and provide the ability to use several UAVs to monitor the territory simultaneously.

OVERVIEW OF EXISTING SOLUTIONS

Currently, there are a number of systems designed for flight planning and routing of unmanned aerial vehicles (UAVs) in various applications, but most of them have limitations in terms of adaptation to the specific conditions of environmental monitoring of large natural areas.

Mission Planner is an open source flight planning platform compatible with many autopilots, such as ArduPilot and PX4 [4]. It provides route creation using predefined waypoints and supports 3D terrain visualization. The main advantages are its free of charge, active community, and customization, but it has a complex interface for an unprepared user and limited automatic route optimization algorithms.

QGroundControl is a cross-platform, open-source flight planning solution that supports various types of drones [5]. It has functions for automatic route planning with obstacle avoidance. The advantages of the system include free and open source, wide customization options, support for various operating systems, compactness and efficiency. The disadvantages are a less developed user interface compared to commercial systems, limited analytical capabilities, and the need for technical knowledge to make full use of the system.

DJI FlightHub 2 is a corporate platform designed to manage a fleet of drones with the functions of remote route planning, real-time flight monitoring, and data processing [6]. The platform is well suited for centralized management of large groups of UAVs, but has limited flexibility in planning complex routes with full coverage of natural areas.

SkyGrid is a cloud-based solution designed for automated drone mission planning using artificial intelligence [7]. It is suitable for tasks where adaptability to environmental changes is critical, but requires high-quality input data and significant computing resources.

Most existing solutions either focus on basic planning through point routes without full coverage of the territory or are focused on centralized management of groups of drones in corporate applications. Problems for environmental monitoring include the lack of automatic generation of full coverage routes, insufficient consideration of terrain and relief features, limitations in optimizing routes based on time and energy consumption, as well as the high cost of commercial licenses or closed systems.

SOLUTION METHOD

The problem of UAV routing for environmental monitoring of forest areas belongs to the category of Coverage Path Planning (CPP) problems with multiple constraints. In contrast to the tasks of navigation between individual points, in CPP, it is necessary to ensure full coverage of the entire observation area with minimal loss of time, energy, and taking into account the terrain and obstacles. Basic CPP methods, such as zigzag or spiral trajectories, are simple to implement but have significant drawbacks. They do not provide a non-ergo-optimal order of traversing the coverage areas, do not take into account changes in altitude or weather conditions, and are poorly scalable for complex object geometry or a large monitoring area [8].

To solve the problem of routing unmanned aerial vehicles in the process of environmental monitoring of forest areas, a hybrid approach is proposed that combines the method of full coverage of the territory with a modified ant optimization algorithm (Ant Colony Optimization, ACO). This approach makes it possible to ensure the completeness of the overflight of a given area while minimizing energy consumption and mission duration.

Stage 1. Construction of the monitoring area breakdown.

The monitoring area is divided into sub-areas of coverage. Let $A \subset \mathbf{R}^2$ be a given monitoring area, the breakdown is as follows:

$$A = \bigcup_{i=1}^{n} C_i \quad (1)$$

where the monitoring area is divided into n non-overlapping cells C_i. Each cell C_i is considered a separate route segment that must be visited at least once. The division is performed taking into account the resolution of the UAV's sensors, the complexity of the terrain and possible obstacles. Sub-zones are formed to ensure complete coverage of the territory without gaps and minimize the number of required turns and flights.

Stage 2. Building a graph of transitions between subzones.

At this stage, the graph G (2) is constructed, and the weight of the edge between the vertices is determined by the combined cost metric using formula (3):

$$G = (V, E) \quad (2)$$

where V is the set of vertices that correspond to the centers of the cells in partitioning (1); E is the set of edges.

$$w_{ij} = \alpha \cdot d_{ij} + \beta \cdot \Delta h_{ij} + \gamma \cdot r_{ij} \quad (3)$$

where w_{ij} is the weight of the edge between vertices v_i and v_j, $v_i, v_j \in V$;

d_{ij} — is the Euclidean distance between the centers of cells C_i and C_j;

Δh_{ij} — is the height difference between the cells C_i and C_j;

r_{ij} — is an indicator of the risk of transition between C_i and C_j cells;

α, β, γ — weighting coefficients of the importance of the relevant factors;

Each node of the graph corresponds to a separate subzone, and the weights of the edges between the nodes are determined based on heuristic indicators such as the distance between the centers of the subzones, the difference in terrain heights, and the expected energy consumption for the flight.

Stage 3. Building a route.

After the graph is built, a modified ant optimization algorithm is applied. The task is to find a route that minimizes the total cost of transitions and is calculated using the formula (4)

$$P^* = \arg\min_{P \in M} \left(\sum_{(i,j) \in M} w_{ij} \right) \quad (4)$$

where M is the set of all possible routes that cover all cells without gaps. Each ant builds a route by sequentially selecting the next cell, taking into account the pheromone trail and heuristic attractiveness according to formula (5)

$$P_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in J_k} [\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}} \quad (5)$$

where J_k is the set of vertices not yet visited;

P_{ij}^{k} is the probability that ant k, being in vertex i, will choose vertex j as the next one;

τ_{ij} is the pheromone intensity on the edge between vertices i and j;

η_{ij} is the heuristic attractiveness of the transition, which is given as $\eta_{ij} = 1 / w_{ij}$;

α - pheromone exposure parameter;

β is the heuristic influence parameter.

Virtual “agents” imitate the natural process of finding optimal routes to cover the territory. Based on pheromone traces and local heuristics, the order of bypassing subzones is determined, which minimizes the total route length and energy consumption [9]. In the process of iterations, pheromones are amplified on those paths that have shown the best results according to mission quality criteria. After each iteration, the routes update the pheromones on the edges using the formula (6)

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{k=1}^{m} \Delta \tau_{ij}^{k} \quad (6)$$

where is the amount of pheromone left by an ant on an edge after completing its route in one iteration of the algorithm, calculated by formula (7)

$$\Delta \tau_{ij}^{k} = \begin{cases} Q / L_k \\ 0 \end{cases} \quad (7)$$

where Q is the pheromone intensity, L_k is the length of the ant's route k, and p is the evaporation coefficient. Also, if the edge (i, j) is included in the route k, then the ant uses this transition as part of the route, otherwise it does not use it.

After determining the optimal sequence of subzone traversal, a local coverage path is generated for each subzone separately. Within the sub-zones, a classical scanning route planning method is applied to fully survey the entire sub-zone area.

At the final stage, the routes are optimized using local improvement methods, such as the 2-opt algorithm, which allows changing individual trajectory segments to further reduce the path length without losing coverage. At the same time, the system checks whether the built route complies with the UAV's

technical characteristics, such as maximum flight time, power reserve, flight altitude limitations, and weather conditions.

The result of the algorithm is a complete route consisting of an ordered sequence of points with reference to the flight altitude.

System Functionality

The functionality of the system includes selecting a monitoring area through a map, setting technical parameters of the vehicle and mission restrictions, automatic route generation at the touch of a button, the ability to manually edit the route, as well as checking and visualizing the route before execution. Table 1 shows the main elements of the user interface of the developed information-analytical system for routing drones, their functional purpose, and the nature of interaction with the user:

Table 1

The Main Elements Of The User Interface

Component	***Functionality***	***Interaction***
Map	Displaying the territory	Selecting the monitoring area
Parameters panel	UAV settings	Entering technical characteristics
Route controller	Controlling the construction	Starting the algorithm, editing
Information panel	Display of key route metrics	View time, route length, coverage area
Visualization panel	Show the finished route with point and trajectory labels	Switching map layers, zooming in

The user interface of the system should be built taking into account the principles of simplicity, functional completeness and intuitive interaction. The basis of the interface is an interactive map that allows not only viewing geographic layers but also selecting the overflight area by drawing a polygon or uploading a ready-made file.

The UAV's parameter panel allows the user to flexibly customize flight characteristics, including time in the air, altitude, speed, and power consumption. Once the parameters are set, the route controller launches a hybrid route construction algorithm and, if necessary, manually corrects the resulting path. All changes are displayed in real time through the visualization panel, which shows a map with the route, control points, restriction zones, and predicted coverage.

For convenience, there is a dashboard that displays key flight metrics: route length, coverage area, and expected mission time.

Conclusions

The paper proposes the concept of an information-analytical system for routing unmanned aerial vehicles for environmental monitoring of forest areas. The system is based on a combination of the method of full coverage of the territory and the ant optimization algorithm, which allows to ensure full coverage of a given area while minimizing the mission time and energy consumption.

The proposed architecture provides a modular approach to building a system, including components for data acquisition, building a graphical model of the environment, route optimization, and visualization. The use of a hybrid method allows taking into account the technical limitations of UAVs, terrain features, weather conditions, and airspace restrictions.

The obtained results can be used to create practical systems for automated planning of UAV missions in environmental monitoring, forestry, control over the state of natural resources, as well as in other areas where systematic survey of large areas is required.

References

[1] H. Choset, “Coverage for robotics – A survey of recent results,” Annals of Mathematics and Artificial Intelligence, vol. 31, no. 1–4, pp. 113–126, 2001.

[2] “About OpenStreetMap,” OpenStreetMap. [Online]. Available: https://www.openstreetmap.org/about. [Accessed: Apr. 29, 2025].

[3] M. Dorigo, M. Birattari, and T. Stützle, Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique. Brussels: Université Libre de Bruxelles, 2006.

[4] “Mission Planner overview,” ArduPilot. [Online]. Available: https://ardupilot.org/planner/docs/mission-planner-overview.html. [Accessed: May 1, 2025].

[5] “QGroundControl.” [Online]. Available: https://qgroundcontrol.com/. [Accessed: May 1, 2025].

[6] “FlightHub 2,” DJI Enterprise. [Online]. Available: https://enterprise.dji.com/flighthub-2. [Accessed: May 1, 2025].

[7] “SkyGrid – Enabling the future of air mobility.” [Online]. Available: https://www.skygrid.com/. [Accessed: May 1, 2025].

[8] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” Robotics and Autonomous Systems, vol. 61, no. 12, pp. 1258–1276, 2013.

[9] H. Duan, “Ant colony optimization: Principle, convergence and application,” in Handbook of Swarm Intelligence (Adaptation, Learning, and Optimization), B. K. Panigrahi, Y. Shi, and M. H. Lim, Eds. Berlin: Springer, 2011, vol. 8, pp. 373–388.

Task Scheduling System

Kamchatnyi Vladyslav, Zhurakovska Oksana
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
kamchatnyi.vladyslav@lll.kpi.ua, o.zhurakovska@kpi.ua

Abstract. **The scheduling system allows a potential user to determine the date and time interval of a new task using heuristic algorithm that considers already existing tasks. The system consists of a client and a server side. The former one is a mobile application for managing user's daily tasks with the possibility to build a new schedule with heuristic algorithm. The latter is a storage of user's account data and added tasks.**

Key words: schedule, information system, scheduling algorithm, decision-making support.

INTRODUCTION

Nowadays, the combination and balance of work and free time is an important part of every person's daily life. The variety of tasks and their number is growing, but the amount of time available remains constant. It is important to be able to properly allocate this time due to its limited nature and try to increase the efficiency of tasks simultaneously. Another important issue is reducing the burden of regular workflow, which can be achieved with balanced distribution of tasks.

This problem can be solved with a system that recommends a task schedule based on the user's workload, existing and new tasks durations, complexities and priorities.

SYSTEM STRUCTURE

The system consists of several components that make up its client-server architecture [1]. Fig. 1 presents the schematic structure of the system, highlighting three key components: the client, the server and the database.

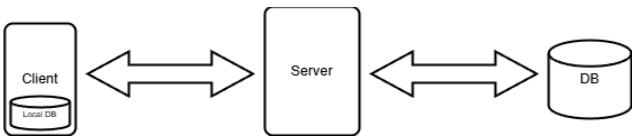

Fig. 1. System architecture diagram

The client-side component of the system is a mobile application developed for the Android operating system. It enables users to perform CRUD operations [2] on their tasks and request a schedule from the system when needed. Furthermore, the client includes a local database that stores user-created tasks.

The server-side component of the system is responsible for processing HTTP requests [3] to authorize users in the application using JWT [4] and for propagating changes in user task data to the database.

The database stores user account information, JWTs, received in requests from the server and tasks created by users.

SCHEDULING ALGORITHM

Input data:

$D = \{ D_1, D_2, ..., D_k, ..., D_m \}$ – set of dates defined by range [current date, approximate deadline specified by the user];

$J = \{ J_1, J_2, ..., J_i, ..., J_n \}$ – set of tasks, each characterized by the following parameters:

p_i – task duration;

pr_i – task priority, equals to 1 if task has low priority; equals to 2 if task has high priority;

d_i – task difficulty, equals to 1 if task is normal; equals to 2 if task is difficult;

f_i – task difficulty, equals to 0 if task is unfixed; equals to 1 if task is fixed;

J_N – new task to add to schedule;

$[T_{min}, T_{max}]$ – user's activity period;

$t_N = [t_{min}, t_{max}]$– desired execution period for the new task within user's activity period $[T_{min}, T_{max}]$ (optional parameter).

Variables:

S_i – start time of task J_i(if $f_i = 0$);

C_i – finish time of task J_i(if $f_i = 0$).

Constraints:

Each task must start and end within activity period:

$$\forall i: S_i \in [T_{min}, T_{max}], C_i \in [T_{min}, T_{max}] \quad (1)$$

No two tasks can overlap:

$$\forall i \neq j: S_j \notin [S_i, C_i] \text{ або } S_i \notin [S_j, C_j] \quad (2)$$

If the optional parameter t_N is specified, the start and end times of the new task must fall within this period:

$$t_{min} \leq S_i \leq t_{max}, t_{min} \leq C_i \leq t_{max} \quad (3)$$

High-priority tasks must start earlier than low-priority tasks:

$$\forall i,j: pr_i > pr_j \Rightarrow S_i < S_j \quad (4)$$

Objective function:

To identify the least loaded date, the workload for each date must first be calculated based on already scheduled tasks. A preference relation is then established based on the calculated workload levels, allowing dates to be ordered by increasing workload:

$$\forall k,r: \sum_{i \in J(D_k)} p_i \cdot pr_i \cdot d_i \geq \sum_{i \in J(D_r)} p_i \cdot pr_i \cdot d_i \Rightarrow L_k \leq L_r, \quad (5)$$

where L_k – date at index k, which is the element of set L ordered in ascending order and L_r – date at index r, which is also the element of set L ordered in ascending order.

To identify an interval in the schedule for a new task, the number of idle periods within activity period must be minimized:

$$N_I = \sum_{i=1}^{n-1} I_i\left(S_{i+1} - C_i > 0\right) \rightarrow min, \quad (6)$$

where I_i – function, which value equals to:

0, if $S_{i+1} - C_i \leq 0$;

1, if $S_{i+1} - C_i > 0$,

N_I – number of idle periods.

To place the new task J_N among the tasks within the activity period $[T_{min}, T_{max}]$, considering the fixed status f_i of each task and minimizing the number of idle periods within activity period.

The scheduling function on the client side can be used when adding a new task or changing the date and execution time of an existing task. The problem can be classified as single machine scheduling problem [5]. Each task executed on the machine has its duration in minutes, difficulty, which can be normal or high, and priority, which can be low or high. Schedule is being built with heuristic algorithm. It has two stages: selecting the date and selecting the time interval for task execution.

At the first stage, the algorithm determines the date for task execution based on the parameters of the new task and the workload of each date within the range specified by user before the algorithm starts. The workload is calculated based on durations, difficulties and priorities of tasks for each date in the specified range (5). Depending on the parameters of the new task, the algorithm recommends either the date with the lowest calculated workload or the nearest date within the specified range. The latter option is only considered if the new task has high priority and normal difficulty.

At the second stage, the algorithm determines a time slot for task execution within the activity period specified by user for the date, which was selected at the first stage. During the schedule building process, the algorithm tries to comply with constraints (1), (2), (3), (4).

The algorithm assigns to each task the earliest possible time slot. The schedule building process consists of four steps:

1. Distribution of the tasks with fixed execution intervals within the activity period;
2. Distribution of the high-priority tasks with unfixed execution intervals within the activity period;
3. Distribution of the low-priority tasks with unfixed execution intervals within the activity period;
4. Rearrangement of the tasks within the new schedule to minimize number of idle periods (6).

An acceptable schedule within the activity period on the selected date is the result of the algorithm, which was built in relatively short computational time.

Fig. 2 presents an example of the schedule, which was built with the algorithm. All tasks in the new schedule were distributed according to constraints that were specified before the start of algorithm.

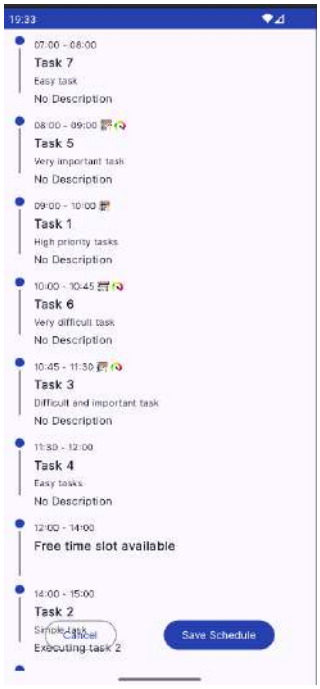

Fig. 2. New schedule built with the algorithm

CONCLUSION

The task scheduling system gives tools to describe upcoming tasks, build new schedule based on descriptions and save all the data. Additionally, the system can assist with determining the execution time of a new task by applying heuristic algorithm to build a new schedule. All user data can be stored either locally on the client side or on the remote server to provide continuous access.

The system helps users to increase their productivity while at the same time reduce the workload. Furthermore, system tries to balance the time spent on work or study and free time monitoring the progress of the tasks.

The system can be used across various situations and use cases. It can be used for planning household chores, creating a task execution plan for students and employees or combining the tasks from different categories such as work, studying and leisure. The most effective way to use the system is for getting recommended time to schedule tasks with deadlines but without fixed execution time.

REFERENCES

1. S. Mbugua, V. Mony, prof. G. M. Nyabuto. Architectural review of Client-Server Models. International Journal of Scientific Research & Engineering Trends. – 2024. – vol. 10, is. 1. – P. 139-140.

2. Prof. K. M. Jadhav, prof. T. S. Patil, prof. P. S. Sutar, prof. H. S. Bhore. Review Paper on Database Basics for Developers : Understanding CRUD Operations. International Journal of Research Publications and Reviews. – 2025. – vol. 6, is. 2. – P. 2696-2697.

3.https://developer.mozilla.org/enUS/docs/Web/HTTP/Guides/Session (referenced 23.04.2025).

4. https://jwt.io/introduction (referenced 23.04.2025).

5. Hongyun Xu, Quan Ouyang. The Review of the Single Machine Scheduling Problem and its Solving Methods. Applied Mechanics and Materials. – 2013. – vols. 411–414 – P. 2081-2084.

Multicriteria Choice of Pre-established Businesses on the Marketplace Using the TOPSIS Method

Vorobiov Oleksii, Zhurakovska Oksana
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
vorobyov.oleksiy1@lll.kpi.ua, o.zhurakovska@kpi.ua

Abstract. **The paper proposes an approach to the formation of recommendations for the selection of pre-established businesses in the marketplace using multicriteria choice methods. The TOPSIS method is used to make informed decisions among a set of alternatives. The system structure, stages and principles of business evaluation, data processing and modeling results are presented.**

Keywords: Multiple decision making, multicriteria analysis, TOPSIS.

INTRODUCTION

In the current economic environment, digital platforms have a significant impact on the process of buying and selling businesses. One of the most promising solutions is to create a marketplace that allows you to evaluate and compare pre-established businesses by a number of criteria. To do this, it is necessary to consider many factors (such as profitability, cost, risks, field of activity, etc.), which are associated with solving the problem of multicriteria choice. For this purpose, it is proposed to apply the TOPSIS multicriteria choice method, which provides a simple and clear interpretation of the results.

BASIC MATERIAL

The developed system functions as an online platform that collects information about ready-made businesses and provides the user with a tool for comparing them. The structure includes a database of objects, a module for evaluating by criteria, normalizing data, calculating proximity to the ideal solution, and global weighting coefficients determined by the administrator.

Thus, it is necessary to select the best pre-established business from a set of alternatives based on the specified criteria. To solve this problem, it is proposed to use the TOPSIS method (Technique for Order Preference by Similarity to Ideal Solution) [1].

The scheme of the method involves the following steps:

- normalization of the evaluation matrix;
- weighing the criteria;
- determination of utopian and dystopian solutions;
- calculating the distances of each alternative to these solutions.
- ranking the alternatives by proximity to the ideal.

PROBLEM STATEMENT

Suppose there is a set of pre-established businesses that are evaluated by entrepreneurs:

$$B = \{B_1, B_2, ..., B_n\},$$

where B_i - is a separate business that is evaluated by several criteria.

Then, for each business, evaluation criteria are determined:

$$C = \{C_1, C_2, ..., C_m\},$$

where each criterion C_j characterizes a certain aspect of the business (financial performance, location, category, adaptation to conditions in Ukraine, etc.).

The next step is to build a matrix of business evaluation criteria for each business:

$$X = \begin{matrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{matrix},$$

where x_{ij} is the business B_i evaluation by criterion C_j.

Since different criteria are of different importance, we will introduce weighting factors:

$$W = \{w_1, w_2, \dots, w_m\},\ w_j \geq 0,\ \sum_{j=1}^{m} w_j = 1,$$

where each w_j corresponds to the relative importance of the criterion.

Based on the information about business valuations and the importance of the evaluation criteria, it is necessary to build a ranking on the set of businesses B, which will allow the user to make the right decision about choosing the most profitable businesses for investment.

DESCRIPTION OF THE SOLUTION METHOD TOPSIS

Using the TOPSIS multicriteria choice method [1, 2] and such known data as set of pre-established businesses B, business evaluation criteria C, the matrix of business evaluations for each criterion X and weighting coefficients W, the first step is to normalize the evaluations of alternatives.

To do this, it is necessary to take into account that in the set of business evaluation criteria C there are criteria that are subject to maximization – profit criteria C^+, and there are criteria that are subject to minimization C^-.

At the same time, regardless of the normalization procedure, the estimates of alternatives for all criteria will be in the interval [0, 100].

Then, for the profit criteria C^+, the evaluation will be determined by the formula:

$$r_{kj}(x) = \frac{x_{kj} - x_j^-}{x_j^* - x_j^-} \quad (1)$$

where

$$x_j^* = \max_k (x_{kj})$$

$$x_j^- = \min_k (x_{kj})$$

And for the cost criteria C^-, the evaluation will be determined by the formula:

$$r_{kj}(x) = \frac{x_j^- - x_{kj}}{x_j^- - x_j^*} \quad (2)$$

The next step is to calculate the weighted normalized scores of the alternatives:

$$\upsilon_{ij} = w_j * r_{ij} \quad (3)$$

where w_{ij} is the weighed coefficient of the criterion C_j, r_{ij} is the normalized scores determined by relations (1) – (2).

Now we need to construct a positive ideal point PIS (utopian point) and a negative ideal point NIS (dystopian point):

$$PIS = A^+ = \{\upsilon_1^+(x), \upsilon_2^+(x), \ldots, \upsilon_j^+(x), \ldots, \upsilon_m^+(x) =$$

$$= \{(\max(\upsilon_{kj}(x) \mid j \in C^+), (\min(\upsilon_{kj}(x) \mid j \in C^-)) \mid$$

$$k = 1, \ldots, n\}$$

$$NIS = A^- = \{\upsilon_1^-(x), \upsilon_2^-(x), \ldots, \upsilon_j^-(x), \ldots, \upsilon_m^-(x) =$$

$$= \{(\min(\upsilon_{kj}(x) \mid j \in C^+), (\max(\upsilon_{kj}(x) \mid j \in C^-))$$

$$\mid k = 1, \ldots, n\}$$

The next step is to calculate the distances of each alternative to the positive ideal point PIS and the negative ideal point NIS:

$$D_k^+ = \sqrt{\sum_{j=1}^m [\upsilon_{kj}(x) - \upsilon_j^+(x)]^2}, k = 1, \ldots, n$$

$$D_k^- = \sqrt{\sum_{j=1}^m [\upsilon_{kj}(x) - \upsilon_j^-(x)]^2}, k = 1, \ldots, n$$

Now to determine the closeness of each alternative to the positive ideal point of PIS (i.e., to determine the similarity to PIS):

$$C_k^* = \frac{D_k^-}{D_k^+ + D_k^-}, k = 1, \ldots, n,$$

where

$$C_k^* \in [0,1], \forall k = 1, \ldots, n$$

The last step is to organize the alternatives of B in descending order of the value C_k^* of similarity to the positive ideal point of PIS. Then the best alternative is determined as follows:

$$B^* = B_p \mid (C_p^* = \max C_k^*)$$

The ranking constructed in this way can be used in the recommendation component of an information system – a marketplace for buying and selling pre-established businesses – to support the decision-making process of users in choosing the best options.

CONCLUSION

The paper proposes an approach to the multicriteria choice of a pre-established business on the marketplace using the TOPSIS method. The implementation of this approach into the system will allow users to rank business alternatives based on objective criteria and personalized weights, which will make the choice process more reasonable and personalized. The modeling results have shown the effectiveness of the approach to practical decision-making tasks, in the presence of a large number of options and conflicting criteria.

It is recommended to implement the proposed approach in real marketplaces for automated business valuation, as well as to apply it in consulting platforms to support investors in choosing the best investment object.

REFERENCES

[1] Alinezhad, A., Khalili, J. New Methods and Applications in Multiple Attribute Decision Making (MADM).-Springer, 2019.-233p.

[2] Papathanasiou, J., Ploskas, N. Multiple Criteria Decision Aid. Methods, Ex-amples and Python Implementations.-Springer, 2018.-173p.

Information System to Support the Activities of Fast-food Establishments

Makovii Viktor, Zhurakovska Oksana
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
makoviy.viktor@lll.kpi.ua, o.zhurakovska@kpi.ua

The article considers the issue of developing an information system to support the activities of fast food establishments. Attention is paid to the recommendatory component of the system, which involves the formation of personal offers to users. Methods for its implementation are proposed, namely the use of association rule and algorithms FP-Growth, Apriori.

Keywords: information system, decision support methods, recommendation generation, association rule

INTRODUCTION

In the modern world, automation of processes in business is the main indicator of efficiency. This is especially true for those parts where speed plays a key role. For example, in fast food establishments, inventory control, cooking accuracy, customer service, and providing them with personal offers must be coordinated and quickly processed. This work examines the main features of supporting the processes of a fast food establishment and proposes methods for forming a system of personal offers.

The main components of the system are shown in Fig. 1.

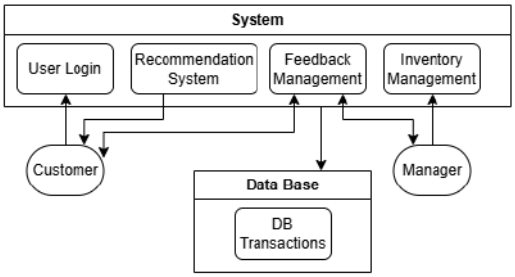

Fig. 1. Main components of the system

PROJECT DESCRIPTION

The main user actions in catering establishments are ordering and viewing the menu, so each user has the opportunity to view the menu and order a dish without registering in the application. However, the system should offer registration to users to notify and update menu items. After registration and authorization, the client has access to leave feedback, pay for the order online, and receive personal offers based on the order history and association rules.

Given the specifics of a fast food restaurant, we can say that the flow of customers will not be constant. Therefore, it will be effective to use computing resources with variable parameters. For speed and flexibility, it is proposed to use cloud services. They allow you to manage computing power as needed, for example, more resources are needed to process orders, and in some periods they are not needed, and it will be optimal to turn them off to prevent costs. For these purposes, systems such as infrastructure as a service (IaaS) or platform as a service (PaaS) are suitable. A bright example is AWS. There are virtual services that can be started and stopped to manage resources using Auto Scaling Group. Such a tool works together with ELB (Elastic Load Balancer), which allows you to balance traffic between instances and can support HTTP or WebSockets requests. But in some cases, you need to keep the service waiting for an order. Then you can keep one instance open and use AWS Lambda, which can perform only one function, such as order processing. This is also suitable for status updates and sending push notifications. You only pay for the execution time of the function.

USER LOGIN

The end-user has the opportunity to register and log in. Then he has access to such functionality as viewing the menu, placing an order, viewing promotions, leaving feed-back, and a personal bonus program. The purpose is to provide access to personal information, order history, loyalty programs and personalized offers.

INVENTORY MANAGEMENT

The manager has the ability to monitor the current state of inventory. The functionality includes checking balances and generating reports, creating and editing orders for sup-pliers, and updating inventory after receipt.

The system should also take into account orders for products from several suppliers and calculate such parameters as reliability of deliveries, conformity of the actual product to the order, and delivery time. Taking into account all these parameters, a supplier characteristic is built and its reliability coefficient is determined. This will allow comparing different suppliers and choosing exactly those products that they deliver better.

CUSTOMER FEEDBACK MANAGEMENT

The main goal of feedback management is to improve customer service and satisfaction. The basic feedback processing scenario includes the following steps: the customer leaves a review, the system classifies the reviews and generates a report on a scale from negative to positive, the system takes into account the impact of reviews on the number of orders, the manager reviews the report to solve problem areas, the system sends a thank you for the re-view.

RECOMMENDATION SUBSYSTEM

The main task of the recommendation subsystem is to form personal suggestions for users. For this purpose, a method of forming recommendations is proposed, which is based on the

application of association rules and decision-making methods based on the history of past orders. The input information used in the proposed method is stored in the Database in the form of a set of transactions. Based on its analysis, patterns in user behavior are identified and recommendations in the form of personal suggestions are formed on this basis. Patterns can be represented in the form of association rules [1], for example, if a buyer orders product A, then often together with it he buys product B. A more detailed implementation should take into account the strength of the relationship (the probability of choosing B).

Fig. 2 presents a diagram of the method for generating recommendations.

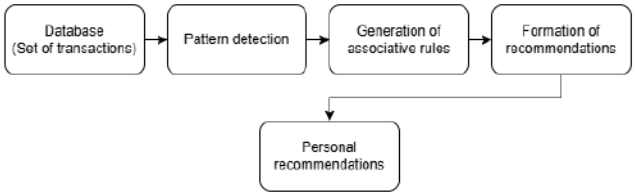

Fig. 2. Schema of recommendation creation

As an illustrative example of such patterns, one can cite the purchase of a cake with coffee.

To generate association rules, it is proposed to use the Apriori and FP-Growth algorithms [1, 2]. It is believed that the FP-Growth method is faster and more efficient than Apriri, because it allows you not to scan the entire database again. The scheme of its operation:

Step 1. Compressing data into a smaller structure, where sets of elements and their frequencies are stored.

Step 2. Dividing the tree into smaller elements and detecting patterns.

Step 3. Generating patterns that describe the relationships between elements.

The formation of recommendations based on the generated association rules will allow the user to offer new, most common menu combinations, taking into account his preferences. Which will allow to increase sales and profit, simplify the analysis of the assortment in order to identify strong and weak positions.

In practice, association rules can be used as discounts on holidays or when a certain amount of orders is reached. To implement full functionality, such elements as a large transaction history, a user identification mechanism, an algorithm for highlighting frequent sets, rule filtering, and an interface where the manager has the ability to edit rules are also required.

Editing should be done according to three parameters.

1) Support - the level of support, and how often a certain set is found in the database. For example, if 10 out of 100 checks have a set A and B, then the support is 0.1.

2) Confidence - The level of confidence, or the probability that when buying A, the client buys B.

3) Lift - shows how much the rule is better than random probability. This parameter shows the independence of A and B. For example, if Lift is greater than, then A increases the probability of B. If it is 1, then A and B are independent. If Lift is less than 1, then A reduces the probability of B. All these parameters need to be filtered to exclude rare combinations or unreliable rules.

DB Transactions

To store transaction data, the optimal option would be to use a relational database. With this type of database, it is possible to qualitatively implement a data storage structure. To store transactions, the best option would be to create two tables, one for storing data on all orders, and the other for storing specific order elements. For example, order 1 has two coffees and two cakes. The order record is stored in the Orders table. The «OrderItems» table has a foreign key to Orders, and in this case, will have two records for individual order elements.

Architecture

The optimal option for implementing the entire project would be to use some parts. Namely, separating the backend and frontend parts. Now many frameworks allow you to create both structural parts in one block, without using other tools. This simplifies development and speeds up the first working version of the project. But for better optimization, those systems were chosen that concentrate and better optimize individual components. Also, taking into account the specifics of the project, it can be assumed that not only the site but also the phone will be used for ordering. Modern tools allow you to build web pages in such a way that all elements are correctly displayed on the screen. Even taking into account the very large difference between the sizes of the monitor and the smartphone and between the smartphones themselves. But when using a monitor, the user interacts with information differently, having a mouse, and the specifics of the UX structure are different between the site and the application. Thus, it would be advisable to develop a version for mobile devices. Having a distributed structure between the front-end and backend simplifies mobile development, where the UI sends requests to an existing controller.

On the backend, architectural separation into structural components is a good practice. For example, separating the authorization/authentication parts, orders, users, and dishes. In turn, using a modular structure together with dependency injections allows you to flexibly configure the project, making it more scalable and easily maintainable.

For example, NestJs distributes the following components:

controller - accepts HTTP requests.

services - are called through controllers and implement business logic.

modules - separate a certain part of the logic, and can be used in other parts of the application using dependency injections.

Mobile Frontend Architecture

An important aspect of any business is to be relevant. But nowadays, with the constant increase in information and a changing world, this becomes a difficult task. Therefore, to create a mobile version of the product, Flutter technology was used, which has such advantages as high development speed and compilation for different platforms. It is impossible to completely separate the writing of UI and taking data from the server, ‘the need to write business logic, so a good approach is to divide by Features. Which is a division not by screens, but by main components. This resembles a modular structure on

the server. It is planned to separate some services for public use in different parts of the code.

In such an architecture, components can be distinguished as

1) Event Handler. This component responds to events from the UI and interacts with the state. In the methods of this structure, you can call a repository that interacts with the database, process data, and publish a new state. Usually, this is where the business logic of individual components is described.

2) The UI part is needed to display data from the state, this is what the user sees. This part specifies which event handler methods need to be applied in parts of the screen.

3) The Repository component interacts with the database. In this case, these will be calls either directly to the database or HTTP calls to the server part.

4) State - The state of some logical part. It can be stored as the state of a separate screen or the state of a separate Feature that is used on several screens.

5) Services. Separate components, they do not have their own data display, but only work with data. A vivid example would be the description of tasks that are performed in the background.

There are several possible variations of using these components, this is due to the features of the application and its specifics. To solve the problem of a fast food application, the optimal option would be to create a modular structure. With this approach, there is a possibility of team development, because the structure is distributed not by screens, but by logical components that can be used on several screens. Thus, an individual developer works on one logical component. Another positive point is the use of services, repositories, and states of logical structures on several screens. This is done using dependency injection. Such a structure speeds up work and has a clear appearance. There is also no need to duplicate the state on separate screens. This can cause a problem where there are several sources of information. In addition, when updating the state used in several parts in parallel, there is a risk of overwriting existing data. An example of this problem is shown in Figure 3.

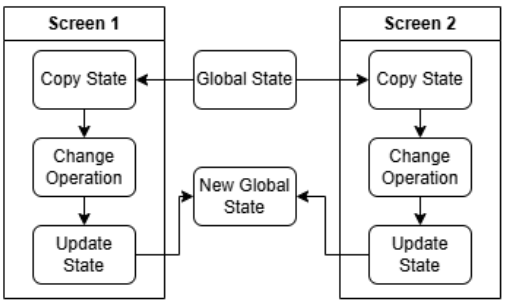

Fig. 3 The problem of parallel data updates

Another problem is the constant retrieval of current data. This is necessary so that the user always sees the current information without sending requests. Since this approach is resource-intensive, it should not be used elsewhere. For implementation in Flutter, Stream is used. Usually, a function is also passed that will be executed when new data is received.

But some tools allow you to save resources and store the last value. One of them is BehaviorSubject, which stores the last value and issues it first. This also solves the problem with new subscribers, they will still be able to get the last value without waiting for a new data update.

Conclusions

The paper presents the structure of an information system to support the activities of fast food establishments. A stack of technologies is proposed and architectural approaches to implementation are considered. A method for generating recommendations is proposed, which allows for generating personal offers to users and is the basis of the recommendation subsystem. The implementation of the system will simplify the organization of work of fast food establishments and increase its efficiency.

References

1. Zaki M.J., Meira Jr. W. Data Mining and Machine Learning: Fundamental Concepts and Algorithms. - Cambridge University Press, 2020. - 776p.

2. Shah K., Shah N., Sawant, V., Parolia N. Practical Data Mining Techniques and Applications. - Auerbach Publications, 2023. - 214p.

A Hybrid Algorithm for Item Selection Considering Constraining Parameters and Its Application in Selecting Hiking Gear

Mykhailo Kutsarskyi, Maryna Khmeliuk
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
kutsarskiyy@gmail.com, khmeliuk.maryna@lll.kpi.ua

Abstract. **This paper addresses the problem of item selection considering a set of constraining parameters. An analysis of classical solution search methods, particularly exhaustive search and simulated annealing, is presented. Based on the identified shortcomings of these approaches, the authors propose a hybrid algorithm that combines a greedy strategy, round-robin category processing, and adaptive scaling of weighting coefficients. The paper formalizes the algorithm and demonstrates its practical application in an automated system for selecting objects based on given conditions.**

***Keywords*: optimization algorithms, hybrid algorithm, client-side development, development technologies and tools, Linux-based systems, mobile application.**

Гібридний алгоритм підбору елементів з урахуванням обмежуючих параметрів та його застосування в підборі туристичного спорядження

Куцарський Михайло, Хмелюк Марина
КПІ імені Ігоря Сікорського
м.Київ, Україна
kutsarskiyy@gmail.com, khmeliuk.maryna@lll.kpi.ua

Анотація. **У тезах доповіді розглянуто задачу підбору елементів з урахуванням множини обмежуючих параметрів. Було проведено аналіз класичних методів пошуку рішень, зокрема повного перебору та імітації відпалу. На основі виявлених недоліків цих підходів авторами запропоновано гібридний алгоритм, що поєднує жадібну стратегію, категоріальну обробку типу round-robin та адаптивне масштабування вагових коефіцієнтів. У роботі наведено формалізацію алгоритму та продемонстровано його практичне застосування у системі автоматизованого підбору об'єктів за заданими умовами.**

Ключові слова: оптимізації, гібридний алгоритм, клієнтська розробка, технології та засоби розробки, Linux-подібні системи, мобільний застосунок.

Вступ

Задачі вибору оптимальної підмножини об'єктів із наявного набору, що задовольняє низці обмежуючих умов, є поширеними в різних прикладних галузях: логістиці, плануванні ресурсів, комплектуванні обладнання, організації подорожей тощо. Вони належать до класу задач комбінаторної оптимізації та характеризуються високою складністю в умовах багатофакторного впливу — вагових, кількісних, функціональних, контекстних або часових обмежень. При цьому класичні методи, зокрема динамічне програмування або повний перебір, стають малоефективними при розширенні простору рішень або змінності параметрів.

Актуальність задач даного типу зумовлює потребу в розробці нових евристичних і гібридних підходів, які дозволяють досягати балансу між обчислювальною складністю та якістю результату. У даній роботі запропоновано реалізацію гібридного алгоритму, що поєднує жадібну стратегію із категоріальним розподілом і динамічним масштабуванням коефіцієнтів релевантності на основі вхідних параметрів. Такий підхід дозволяє ефективно адаптуватися до

змінних умов і формувати оптимізовану вибірку елементів в умовах численних обмежень.

Метою дослідження є побудова алгоритмічного рішення, яке може бути застосоване в різноманітних прикладних контекстах, де потрібно автоматизовано обрати релевантні об'єкти з урахуванням пріоритетів, обмежень і багатовимірного простору рішень.

Пошук рішення

Підбір елементів із множини кандидатів з урахуванням заданих обмежень є NP-складною задачею. У випадку обмеження на загальну вагу, цінність або інші метрики, задача редукується до модифікованої форми задачі про рюкзак (Knapsack Problem), яка не має відомого поліноміального алгоритму для точного розв'язання. Тому у практичних застосуваннях активно використовуються евристичні або гібридні підходи.

Повний перебір (Brute Force)

Метод повного перебору полягає в генерації усіх можливих підмножин елементів і перевірці кожної з них на відповідність обмеженням. При цьому значення цільової функції (наприклад, загальна релевантність або сумарна користь) розраховується для кожного варіанту, після чого обирається найкращий [1].

Нехай маємо множину з n елементів, кожен з яких характеризується цінністю V_i та вагою W_i. Обмеженням є максимальна допустима вага W_{max}. Тоді простір рішень становить 2^n, що робить метод непридатним при великих значеннях n через експоненційне зростання обчислювальної складності.

Головною перевагою цього методу є гарантія знаходження глобально оптимального розв'язку — жоден варіант не буде втрачений або проігнорований. Проте така точність досягається надзвичайно високою ціною в обчислювальному сенсі. Метод повного перебору швидко стає непридатним при зростанні розмірності задачі, навіть якщо кількість можливих елементів не є великою. Це обмежує сферу застосування підходу виняткова демонстраційними або аналітичними сценаріями на малих даних.

Імітація відпалу (Simulated Annealing)

Алгоритм імітації відпалу імітує процес охолодження металу, поступово переходячи до мінімуму функції енергії (або, в нашому випадку, функції невідповідності)[2]. Він починається з довільного рішення та на кожному кроці з певною ймовірністю приймає гірші розв'язки, що дозволяє уникати локальних мінімумів.

Пошук керується температурною функцією $T(t)$, яка зменшується з часом, та ймовірністю переходу до нового стану:

$$P = e^{-\Delta E/T}$$

де ΔE — різниця між поточним і новим рішенням, а T — температура.

Перевагою такого методу є здатність досліджувати простір рішень глобальніше, дозволяючи тимчасово приймати гірші варіанти з метою подолання локальних екстремумів. У разі правильної настройки температурного графіка та функції прийняття рішень, алгоритм демонструє високий потенціал в знаходженні наближених до оптимальних рішень. Водночас основним недоліком є нестабільність результатів — одні й ті самі вхідні дані можуть дати різні результати в залежності від початкового стану та псевдовипадкових переходів. Крім того, вибір параметрів охолодження є нетривіальним і вимагає глибокої емпіричної роботи.

Гібридний евристичний алгоритм з категоріальним розподілом

Для досягнення високої ефективності у задачах з великим числом елементів і гнучкими обмеженнями запропоновано гібридний алгоритм, що поєднує жадібну евристику [3], адаптивне масштабування коефіцієнтів релевантності та категоріальну кругову обробку (round-robin)[4].

Кожен елемент має базову цінність B_i, яка масштабуються згідно з множниками умов M_{ji}, що відповідають за специфічні обмеження, такі як контекст використання чи пріоритет. Актуальна вагомість обраховується за формулою:

$$V_i = B_i \times M_{1i} \times M_{2i} \times \ldots \times M_{ni}$$

Для порівняння релевантності елементів із врахуванням їх ваги використовується коефіцієнт ефективності:

$$S_i = \frac{V_i}{W_i}$$

Після обчислення коефіцієнтів ефективності для кожного елемента система переходить до побудови фінальної вибірки. Спочатку всі елементи групуються за категоріями, після чого виконується кругова ітерація, у межах якої з кожної категорії послідовно обирається найрелевантніший елемент відповідно до значення S_i. Додавання до вибірки триває доти, доки не буде вичерпано доступний ресурс (наприклад, ваговий ліміт), або ж усі категорії не будуть забезпечені необхідними елементами. У разі, якщо певна категорія залишилася порожньою, алгоритм виконує компенсаційний добір для забезпечення повноти функціонального покриття, навіть якщо це потребуватиме перевищення обмеження у межах допустимого допуску.

Перевагою запропонованого гібридного підходу є поєднання швидкодії жадібного алгоритму з гнучкістю адаптивного масштабування вагомості, що дозволяє формувати релевантні рішення у динамічно змінних умовах. Завдяки категоріальному розподілу забезпечується збалансованість результату та уникнення домінування однієї категорії над іншими. Крім того, механізм компенсаційного добору мінімізує ризик втрати критично важливих елементів.

Натомість, алгоритм не гарантує глобальної оптимальності розв'язку, що є типовим обмеженням евристичних методів. Його ефективність значною мірою залежить від коректності початкових множників актуальності, а також від того, наскільки адекватно сформульовано категорії та обмеження. У випадках з великою кількістю тісно пов'язаних умов або ж високою вартістю помилки у виборі, може знадобитися інтеграція додаткових оптимізаційних механізмів.

Практичне застосування

Розроблений гібридний алгоритм було імплементовано в рамках мобільного застосунку TrekMate, функціональність якого полягає в автоматизованому підборі туристичного спорядження відповідно до умов конкретного походу. Система реалізована у форматі клієнтської розробки, що дозволяє здійснювати усі обчислення та збереження даних локально, без необхідності підключення до серверної інфраструктури. Для реалізації алгоритму було використано сучасні технології і засоби розробки, зокрема мову програмування Kotlin [5], бібліотеки Jetpack Compose [6] для UI та Android SDK [7] для взаємодії з файловою системою і локальною базою даних. Зовнішній вигляд застосунку, опитування, рекомендований список спорядження наведено на рис. 1. В основі системи лежить модель багатофакторної оптимізації з обмеженнями, де гібридний підхід дозволив досягти адаптивності, релевантності та структурованої рівномірності вибору елементів.

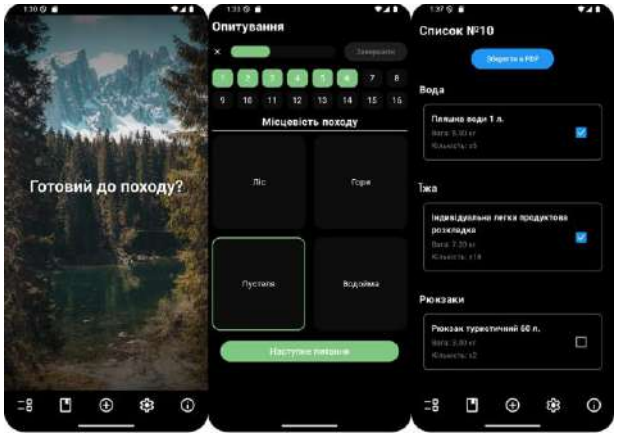

Рис. 1. Зовнішній вигляд TrekMate

Ключовою складовою системи є механізм опитування, що побудований на 16 питаннях загального характеру, які охоплюють найважливіші аспекти подорожі: тривалість, кількість учасників, погодні умови, тип маршруту, сезонність тощо. Така кількість запитань була визначена як достатня для отримання повного уявлення про умови подорожі, необхідного для формування персоналізованих рекомендацій. Зменшення обсягу опитування обмежило б якість зібраних даних, тоді як його розширення могло б негативно вплинути на зручність і швидкість взаємодії з системою. Структура бази питань та відповідей наведена на рис. 2. Відповіді, отримані від користувача, не обробляються безпосередньо алгоритмом, а трансформуються за допомогою спеціального компонента — DataManager. Саме цей менеджер відповідає за переклад відповідей у числові коефіцієнти релевантності, які надалі враховуються алгоритмом під час розрахунку цінності кожного елемента спорядження.

```
{
  "id": 11,
  "question": "Метод приготування їжі",
  "answers": [
    "Багаття",
    "Газовий пальник",
    "Дров'яний пальник",
    "Інтегрована система"
  ]
},
```

Рис. 2. Структура файлу questions.json

База даних, що використовується у TrekMate, представлена у вигляді JSON-файлу equipment.json і містить інформацію про всі можливі елементи спорядження. Структура бази даних equipment.json наведена на рис. 3. Кожен елемент характеризується низкою параметрів: базова вага, базова цінність, категорія, коефіцієнт умовності використання (conditions_multiplier), тривалість споживання (consumption_duration) та поріг колективного користування (participant_threshold). Останні два показники становлять найбільш значущі обмежувальні характеристики, що визначають адаптацію спорядження до конкретного сценарію.

```
{
  "id": 93,
  "name": "Куртка мембранна демісезонна",
  "base_weight": 0.5,
  "base_value": 900,
  "participant_threshold": 1,
  "category": "Верхній одяг",
  "consumption_duration": 100,
  "conditions_multiplier": 1.0
},
```

Рис. 3. Структура файлу equipment.json

Після обробки вхідних даних користувача, система викликає гібридний алгоритм оптимізації, який формує вибірку спорядження, релевантну заданим умовам. Алгоритм працює в кілька послідовних етапів, дотримуючись принципу пріоритетного наповнення з урахуванням значущості категорій.

На початку рюкзак користувача заповнюється базовими елементами життєзабезпечення — їжею та водою. Кількість таких ресурсів масштабується пропорційно до кількості учасників подорожі та тривалості маршруту. Далі відбувається цільовий добір некругових категорій, до яких належать намети, спальні мішки, каремати та інше критичне групове спорядження. Для таких елементів обчислюється необхідна кількість, яка залежить від колективності використання (наприклад, одна двомісна палатка розрахована на двох осіб) та загальних умов походу.

Після цього активується категоріальна кругова обробка (round-robin), у межах якої інші категорії (одяг, медицина, інструменти, засоби особистої гігієни тощо) обробляються по черзі. На кожному циклі обирається найрелевантніший предмет з кожної категорії на основі адаптованої цінності, яка враховує відповіді користувача в опитувальнику, а також обмеження щодо ваги, умовності та спільного використання. Таким чином реалізується жадібна стратегія у межах багатофакторного простору, що дозволяє ефективно балансувати між функціональністю та вагою спорядження.

У підсумку формується персоналізований список спорядження, максимально адаптований до умов подорожі. Фактично, користувач, не підозрюючи про це, сам задає вагомість окремих предметів через відповіді в опитувальнику, і саме ці пріоритети формують логіку добору. Результат записується до JSON-файлу optimizedEquipment.json і передається системі для подальшої візуалізації та взаємодії. Структура файлу optimizedEquipment.json наведена на рис. 4.

```
{
  "id":162,
  "name":"Рюкзак туристичний 40 л.",
  "quantity":3,
  "category":"Рюкзаки",
  "weight":3
},
```

Рис. 4. Структура файлу optimizedEquipment.json

Оскільки застосунок TrekMate розроблено для операційної системи Android [8], в основі якої лежить ядро Linux, було забезпечено сумісність з широким спектром пристроїв і гарантовано ефективне управління ресурсами під час виконання обчислень алгоритму.

Висновки

У межах проведеного дослідження було сформульовано задачу підбору елементів з урахуванням множини обмежуючих параметрів та проведено аналіз класичних підходів до її розв'язання. Метод повного перебору виявився неефективним для масштабних випадків через обчислювальну складність, а алгоритм імітації відпалу — недостатньо передбачуваним у контексті строго структурованих обмежень.

На цій основі було запропоновано гібридний евристичний алгоритм, що поєднує жадібну стратегію з категоріальним підходом round-robin та адаптивною системою вагових коефіцієнтів. Такий підхід забезпечує баланс між точністю відбору і швидкістю обчислення при врахуванні множинних факторів, включно з пріоритетністю, колективністю використання елементів і релевантністю до заданих умов.

Розроблений алгоритм було інтегровано у прикладне програмне забезпечення TrekMate, що демонструє можливість його практичного застосування у задачах автоматизованої генерації персоналізованих списків туристичного спорядження. Алгоритм було адаптовано до специфіки предметної області шляхом розширення структури бази даних і впровадження механізмів попереднього опитування користувача. Після обробки введених параметрів формується структурований файл, який згодом використовується системою для відображення результату у зручному для кінцевого користувача форматі. Перспективи розвитку мобільного застосунку полягають у масштабуванні розробленого функціоналу, зокрема розширення бази даних туристичного спорядження, інтеграції із іншими туристичними сервісами та підвищення гнучкості системи підбору спорядження.

Отримані результати підтверджують доцільність запропонованого підходу та створюють основу для подальших досліджень у напрямі оптимізації вибірок з динамічними параметрами та багаторівневими обмеженнями.

Література

1. Brute Force Approach and its Pros and Cons // GeeksforGeeks. – Режим доступу: https://www.geeksforgeeks.org/brute-force-approach-and-its-pros-and-cons/ (дата звернення 20.03.2025).

2. Simulated Annealing Explained // Baeldung on Computer Science. – Режим доступу: https://www.baeldung.com/cs/simulated-annealing (дата звернення 20.03.2025).

3. Greedy Algorithms // GeeksforGeeks. – Режим доступу: https://www.geeksforgeeks.org/greedy-algorithms/ (дата звернення 20.03.2025).

4. Round-robin // WhatIs by TechTarget. – Режим доступу: https://www.techtarget.com/whatis/definition/round-robin (дата звернення 20.03.2025).

5. Kotlin Docs // Kotlinlang.org. – Режим доступу: https://kotlinlang.org/docs/home.html (дата звернення 27.03.2025).

6. Jetpack Compose // Android Developers. – Режим доступу: https://developer.android.com/jetpack/compose (дата звернення 27.03.2025).

7. Platform Architecture // Android Developers. – Режим доступу: https://developer.android.com/guide/platform (дата звернення 28.03.2025).

8. Kernel Overview // Android Open Source Project. – Режим доступу: https://source.android.com/docs/core/architecture/kernel (дата звернення 29.03.2025)

Approach to Improving Web Application Performance by Transforming JavaScript into WebAssembly

Nieliepin Dmitrii, Amons Oleksandr
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
nelepin.dmitriy@gmail.com, o.amons@kpi.ua

Abstract. **This paper addresses JavaScript performance limitations in web applications and proposes an approach to pre-convert JavaScript code to WebAssembly. Several transformation approaches are described, including compilation, interpretation, and neural network translation. Performance comparisons demonstrate the advantages of using Rust as the target language for compilation.**

Keywords*: *JavaScript, WebAssembly, performance optimization, web application.

Підхід до підвищення продуктивності веб-застосунків за рахунок перетворення JavaScript у WebAssembly

Нєлєпін Дмітрій, Амонс Олександр
КПІ імені Ігоря Сікорського
м. Київ, Україна
nelepin.dmitriy@gmail.com, o.amons@kpi.ua

Анотація. **У статті розглянуто обмеження продуктивності JavaScript у веб-застосунках та запропоновано підхід попереднього перетворення JavaScript-коду у WebAssembly. Описано кілька підходів трансформації, включаючи компіляцію, інтерпретацію та нейромережеве перетворення. Порівняння продуктивності демонструє переваги використання Rust як цільової мови для компіляції.**

Ключові слова: JavaScript, WebAssembly, оптимізація продуктивності, веб-застосунок.

Вступ

В наш час розробка веб-застосунків є одною із найбільших галузей розробки, і як результат JavaScript є одною з основних мов, яка використовується у веб розробці. Однак її продуктивність залишається низькою. Основні причини повільної роботи JavaScript-коду пов'язані з інтерпретованою природою мови, динамічною типізацією та обмеженим доступом до низькорівневих ресурсів у браузерах. Попри наявність сучасних JIT-компіляторів [1], таких як V8 (Chrome) [2] і SpiderMonkey (Firefox) [3], вони все ще працюють у межах динамічної моделі виконання, що обмежує можливості для глибокої оптимізації. Крім того, реалізація ефективного машинного коду ускладнюється постійними змінами типів під час виконання, а браузерна модель безпеки обмежує доступ до потоків, пам'яті та SIMD-інструкцій.

Дослідження [4] показують, що значна частина продуктивності JavaScript-програм втрачається через неефективне використання API, зайве копіювання даних та неоптимальні конструкції циклів. У цих дослідженнях [4] проведено аналіз практичних випадків у 16 популярних проєктах який показав, що більшість оптимізацій пов'язані із заміною повільних API на швидші аналоги, але навіть такі зміни не завжди гарантують стабільне покращення: лише 42,68% оптимізацій були ефективними для всіх версій рушіїв, а 15% — навіть знижували продуктивність у певних випадках.

З іншого боку, очікування користувачів щодо швидкодії веб-застосунків також зростають. Згідно з дослідженнями [5], більшість користувачів вважають веб-сайти «швидкими», якщо First Contentful Paint (FCP) не перевищує 1,8 секунди, а затримка понад 3 секунди призводить до 89% відмов від завантаження сторінки на мобільних пристроях. У таких умовах розробники змушені постійно шукати нові шляхи оптимізації JavaScript-коду.

Зважаючи на це, JavaScript часто стає «вузьким місцем» у складних веб-застосунках, особливо тих, що потребують інтенсивної обробки даних, 3D-графіки, машинного навчання чи криптографії. Ситуація ще більше ускладнюється на мобільних пристроях, де є обмеження апаратних ресурсів і потреба в енергоефективності, а значить і висуваються додаткові вимоги до продуктивності застосунків.

Існуючі методи оптимізації

На даний момент можна виділити три основні підходи до підвищення продуктивності web-застосунків: компіляція в машинний код «на льоту» перед виконанням програмного коду (Just-In-Time або JIT); використання Web Workers та використання WebAssembly (WASM). Нижче коротко розглянемо ці підходи.

Сучасні рушії JavaScript — такі як V8 (Google), SpiderMonkey (Mozilla) та JavaScriptCore (Apple) — використовують Just-In-Time (JIT) компіляцію [1] як основну технологію підвищення продуктивності [2]. Її суть полягає в динамічному перетворенні байт-коду або інтерпретованого коду JavaScript у високоефективний машинний код під час виконання, що дозволяє адаптуватися до конкретного середовища виконання, включно з апаратними особливостями та сценаріями навантаження.

Однак, ефективність JIT-компіляції не завжди гарантується. У емпіричному дослідженні [4] було виявлено, що лише 42.68% оптимізацій JIT-компіляторів стабільно дають приріст продуктивності на всіх версіях рушіїв V8 та SpiderMonkey. У решті випадків — або відсутній ефект, або навіть спостерігається регресія продуктивності [4]. Це свідчить про чутливість JIT до конкретного коду, сценарію виконання та версії рушія, що ускладнює надійне прогнозування швидкодії.

Розширення механізмів захисту JIT-компільованого коду, наприклад через техніки write-protection, призводить до втрат продуктивності — в деяких конфігураціях до 12% зниження швидкодії при активному захисті областей пам’яті [6].

Web Workers – це API, що дозволяє виконувати обчислення в окремому потоці незалежно від основного потоку користувацького інтерфейсу (UI thread) [7]. У стандартному середовищі JavaScript — особливо в браузері — код виконується в однопотоковому режимі, що обмежує можливості масштабування при обробці обчислювально інтенсивних завдань, таких як парсинг великих JSON-файлів, кодування/декодування відео або шифрування.

Використовуючи Web Workers, можна винести подібні задачі за межі основного потоку, запобігаючи блокуванню інтерфейсу користувача, що особливо критично для інтерактивних SPA-застосунків.

Web Workers підтримують ізольоване середовище виконання, де відсутній прямий доступ до DOM, але можливий обмін даними через механізм передачі повідомлень (postMessage) [7]. Це вимагає додаткових зусиль з архітектури застосунку, але дозволяє реалізовувати архітектуру поділеної відповідальності: UI обробляється в основному потоці, а обчислення — у Web Worker'ах.

Проте Web Workers мають і свої обмеження: вони не мають доступу до DOM, Window API, LocalStorage тощо [7], що змушує розробників ретельно планувати передачу даних і взаємодію з основною логікою застосунку. Також створення великої кількості worker'ів може викликати накладні витрати на пам’ять, і деякі браузери мають обмеження на кількість одночасно активних потоків.

WebAssembly (скорочено WASM) є низькорівневою бінарною мовою, що виконується з майже нативною продуктивністю у браузерах [8]. Вона стала важливою технологією для прискорення вебзастосунків, які раніше обмежувалися повільнішим JavaScript.

WebAssembly значно підвищує продуктивність вебзастосунків [8, 9]. Експериментальні вимірювання [9] показали, що WebAssembly-реалізація одного і того самого алгоритму може працювати у 2–4 рази швидше за JavaScript у браузерах Firefox та Chromium. Більш того, мультипотокова WebAssembly-реалізація (з використанням Web Workers і SharedArrayBuffer) перевищила швидкодію однопотокової WASM-версії в 9–13 разів, а вручну оптимізованої JavaScript-реалізації — до 24 разів [9].

До основних переваг WebAssembly можна віднести високу продуктивність, підвищену захищенність, так як код працює в ізольованому середовищі пісочниці. Це мінімізує ризики для системи та дозволяє безпечно запускати сторонній код. Мультиплатформеність і кросбраузерність також є перевагами, які надає WebAssembly, так як WASM підтримується усіма основними браузерами та стандартизується W3C WebAssembly Working Group [9]. Додатково слід зазначити що WebAssembly, на відміну від JavaScript коду, надає можливості для реалізації в коді паралельних обчислень. А саме, WebAssembly підтримує багатопоточне виконання з використанням спільної пам’яті, що дозволяє ефективно задіювати всі ядра процесора [9]. Цей підхід не обмежує розробників у виборі мови програмування, WebAssembly може бути ціллю компіляції для багатьох мов (Rust, C, C++, Go, TypeScript та ін.)

Незважаючи на гарні результати, WebAssembly має деякі недоліки, а саме відсутність прямої взаємодії з DOM, складність налагодження програми, необхідність попередньої компіляції, а також, WebAssembly й досі знаходиться у стадії розробки, тому підтримка garbage collection, SIMD і потоків все ще знаходиться у фазі доопрацювання у деяких браузерах [9].

Запропонований підхід

Як було показано, WebAssembly здатен значно перевершити JavaScript у швидкодії, особливо в задачах, де важливе обчислювальне навантаження. Враховуючи це, пропонується підхід з перетворенням JavaScript-коду у іншу мову програмування з подальшею компіляцією у WebAssembly. Це дозволить підвищити продуктивність без необхідності повного ручного переписування логіки вже реалізованого застосунку.

Запропонований підхід допускає декілька стратегій перетвонення JavaScript в проміжну мову програмування.

Інтерпретація або трансляція з використанням традиційних компіляторів є одним зі способів трансформування JavaScript у проміжну мову, наприклад, TypeScript або іншу, що має статичну типізацію, після чого здійснити компіляцію у WebAssembly за допомогою доступних інструментів (наприклад, AssemblyScript, Emscripten) [8].

Семантична інтерпретація інтерпретатором, цей підхід забов’язує реалізувати інтерпретатор, який виконує JavaScript-код шляхом симуляції виконання і паралельно будує еквівалентну WASM-репрезентацію. Такий підхід дозволяє зберегти поведінку JavaScript при цьому оптимізуючи критичні шляхи виконання.

Також можливе використання великих мовних моделей або спеціально навчених нейронних мереж для трансляції JavaScript-коду в еквівалентний код на мові з підтримкою WebAssembly (наприклад, Rust або C). Такий транслятор зможе враховувати контекст виконання, шаблони коду та навіть семантичні особливості JS-коду.

Для визначення проміжної мови, яка буде компілюватися в кінцевий WASM, було проведено дослідження, де було порівняно продуктивність AssemblyScript (TypeScript), Rust та JavaScript без перетворення у WASM при виконанні однакових задач. Іншими популярними мовами є C# та Java, але ці мови не мають офіційної підтримки компіляції у WebAssembly, і як результат не будь-який програмний код компілюється у програмний код WebAssembly. Тому на даний момент вони не розглядалися як варіанти для порівняння.

Враховуючи що WebAssembly зараз не мають доступу до DOM та користувацького інтерфейсу, для порівняння було обрано приклади чисто алгоритмічних задач, а також задач з використанням системних бібліотек, а саме: обчислення факторіалу, обчислення чисел Фібоначі та хешування тексту за допомогою алгоритму SHA-256. Результати дослідження наведені у табл. 1.

Для порівняння продуктивності кожного з варіантів, використовувався Node Benchmark v2.1.4 [10]. За допомогою цієї бібліотеки проводилися вімири продуктивності скомпільованих WASM модулів та JavaScript коду. В табл. 1 наведено середню кількість прогонів виконання кожної задачі за одну секунду, а також середня похибка.

Данні з табл. 1. демонструють, що в усіх задачах більшу продуктивність має WASM, скомпільований з низькорівневої мови Rust. У випадку з задачею обрахування факторіалу від 3 результати залишаються приблизно рівними, й це показує, що для деяких задач з обчисленням простим функцій, перетворення JavaScript коду у інші мови та їх подальша компіляція у WASM не є доцільною, так як вимагатиме більших зусиль від розробників й не дасть значного покращення у продуктивності застосунку.

Таблиця 1

Порівняння швидкодії скомпільованих WASM та JavaScript

Задача	**Мова програмування**		
	JavaScript	***WASM (AssemblyScript)***	***WASM (Rust)***
Число Фібоначі від 3	57,802,488 ops/sec ±3,32%	58,462,477 ops/sec ±3,50%	108,700,497 ops/sec ±8,71%
Число Фібоначі від 20	12,303 ops/sec ±3,92%	16,363 ops/sec ±4,59%	33,399,969 ops/sec ±6,38%
Факторіал від 3	72,412,393 ops/sec ±3,11%	82,167,078 ops/sec ±5,02%	88,176,421 ops/sec ±3,11%
Факторіал від 20	4,863,279 ops/sec ±3.79%	9,948,652 ops/sec ±4.69%	65,596,132 ops/sec ±4,28%
Хешування за допомогою SHA-256	24,28 ops/sec ±6,23%	14,31 ops/sec ±7,85%	2,635,075 ops/sec ±24,26%

Задача з хешуванням за допомогою алгоритму SHA-256 показує, що Rust WASM впорався з нею набагато швидше, але в цьому випадку також варто врахувати, що AssemblyScript не має вбудованої реалізації хешування, тому це також вплинуло на швидкодію кінцевого WASM. Наявність великої кількості вбудованих криптографічних функцій та хешування є одним із переваг Rust, як мови для оптимізації JavaScript застосунків пов'язаних з криптографією.

Висновки

У цій роботі було розглянуто проблеми продуктивності JavaScript у сучасних веб-застосунках, зокрема ті, що пов'язані з динамічною природою мови, обмеженнями JIT-компіляції та складнощами багатопотокового виконання. Було проаналізовано існуючі підходи до оптимізації — від використання Web Workers до застосування WebAssembly, як засобу підвищення продуктивності застосунків у браузері.

Пропонується новий підхід — попереднє перетворення JavaScript-коду у проміжну мову прогамування з послідуючою компіляцією у WebAssembly. Це може реалізовуватись через традиційну компіляцію в TypeScript, спеціалізовані інтерпретатори або нейронні мережі. Такий підхід дозволяє зберегти зручність розробки JavaScript та водночас суттєво покращити швидкодію критичних ділянок коду. Також це дозволить покращувати продуктивність вже написаних на JavaScript застосунків.

Проведене порівняльне дослідження показало, що Rust демонструє найвищу ефективність, як проміжної мови, перед компіляцією у WASM.

Натомість для простих задач з незначним обчислювальним навантаженням перетворення JavaScript у WASM може бути невиправданим з огляду на додаткові витрати часу та зусиль.

Література

1. JIT-компіляція. URL: https://developer.mozilla.org/en-US/docs/Glossary/Just_In_Time_Compilation (дата звернення 01.05.2025).

2. Chrome V8. URL: https://v8.dev/docs (дата звернення 01.05.2025).

3. SpiderMonkey. URL: https://firefox-source-docs.mozilla.org/js/index.html (дата звернення 01.05.2025)

4. Selakovic M. Performance Issues and Optimizations in JavaScript: An Empirical Study / M. Selakovic, M. Pradel // International Conference on Software Engineering (Austin, 14-22 травня 2016 р.). – Austin, 2016 – С. 61-72.

5. Karka, N. R. Front-End Performance Optimization: A Comprehensive Guide. / Karka, N. R. // International Journal of Scientific Research in Computer Science, Engineering and Information Technology. – 2025. – vol. 11, is 2. P. 83-100.

6. Song, S. WasDom: An Efficient Write Protection for Wasm JITed Code with ARM Domain. / Song, S., Shin, C., Kwon, D. // IEEE Access. – 2025. – vol. 13, – P. 26260-26272.

7. Web-Workers API. URL: https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API (дата звернення 02.05.2025)

8. WebAssembly. URL: https://developer.mozilla.org/en-US/docs/WebAssembly/Guides/Concepts (дата звернення 02.05.2025)

9. Kyriakos-Ioannis D. K. Complementing JavaScript in High-Performance Node.js and Web Applications with Rust and WebAssembly / D. K. Kyriakos-Ioannis, N. D. Tselikas // Electronics. – 2025. – vol. 11, is. 11(19). – P. 1-17.

10. Node Benchmark. URL: https://benchmarkjs.com/docs/ (дата звернення 02.05.2025).

Approaches to implementing the technology of efficient resource management in the cloud

Vladyslav Kovalenko
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
vlad.kov@ukr.net

Abstract. **In this work, the main results of mathematical modeling of the resource management and scheduling problem for cloud data centers' clusters are reviewed, and the proposals regarding implementing the software for resource management are provided. Such proposals include the use of plugins for extra filtration and scoring logic, the use of operators for cluster autoscaling, and the categorization of the servers in the cluster.**

Keywords*: *Kubernetes, optimization, resource management, scheduling.

Підходи до реалізації технології ефективного управління ресурсами у хмарі

Коваленко Владислав
КПІ імені Ігоря Сікорського
м. Київ, Україна
vlad.kov@ukr.net

Анотація. **У даній роботі розглядаються основні результати математичного моделювання задачі управління ресурсами та складання розкладів для кластерів хмарних центрів обробки даних, та надаються пропозиції щодо реалізації програмного забезпечення для управління ресурсами. До таких пропозицій можна віднести використання плагінів додаткової логіки фільтрації та скорингу, використання операторів для автоматичного масштабування кластера, та категоризацію серверів у кластері.**

Ключові слова: Kubernetes, оптимізація, управління ресурсами, складання розкладів.

Вступ

Із урахуванням того, що все більше організацій мігрують свою IT-інфраструктуру до хмари, постає питання збільшення ефективності використання обчислювальних ресурсів у хмарних середовищах, як-от центрах обробки даних [1, 2]. Центральне місце у відповідній дискусії займає технологія Kubernetes, що фактично є стандартом індустрії для розгортання програмного забезпечення у хмарних середовищах; ця технологія містить вбудований компонент для управління ресурсами та складання розкладів (*kube-scheduler*); водночас відомо про випадки, коли розклади, складені *kube-scheduler* призводили до неможливості розгорнути програмне забезпечення. [1, 3–7].

Математичні моделі

У статті [1] було визначені методи, цільові функції та обмеження, що притаманні досліджуваним задачам у галузі управління ресурсами та складання розкладів для кластерів Kubernetes; була визначена для подальшого дослідження задача мінімізації витрат на енергоресурси від роботи серверів та максимізації коефіцієнту середнього використання серверів.

У статті [3] та тезах доповіді [4] були сформульовані дві математичні моделі відповідної задачі для кластерів центрів обробки даних (ЦОД) у термінах дискретної (комбінаторної) оптимізації: динамічна та статична. Динамічна модель має на меті відповідати за глобальні рішення щодо управління ресурсами, як-от вмиканням та вимиканням серверів. Статична модель має на меті відповідати за точкове прийняття рішень щодо призначення конкретних новостворених Pod'ів (що містять контейнеризовані застосунки) на вузли (сервери). Моделі враховують можливості клієнтів як орендувати виділений сервер, так і орендувати потужності спільних хостингових серверів за принципом pay-as-you-go або pay-per-use.

Некеровані параметри (дані, що подаються на вхід) у моделях [3–4] є наступними:

– кількості ядер на серверах;

– обсяги пам'яті на серверах;

– інформація про те, які сервери орендуються як виділені якими клієнтами, а які сервери є спільними хостинговими;

– вимоги щодо кількості ядер для Pod'ів;

– вимоги щодо обсягу пам'яті для Pod'ів;

– інформація про те, якими саме клієнтами створені Pod'и;

– інформація про те, чи дозволяє клієнт запускати Pod на сервері спільного хостингу.

Змінні у моделях [3–4] відповідають наступним рішенням щодо роботи кластера:

– чи має бути конкретний сервер увімкнутий;

– на який сервер має бути призначений конкретний Pod.

Цільові функції (критерії оптимізації), що визначені у моделях [3–4], є наступними:

– мінімізація увімкнених спільних хостингових серверів у кластері (із урахуванням того, що увімкнення та вимкнення виділених серверів регулюється клієнтами, що орендують ці сервери, а не власником кластера);

– максимізація середнього коефіцієнта використання ресурсів спільних хостингових серверів;

– мінімізація загальної кількості увімкнень та вимкнень серверів;

– мінімізація загального коефіцієнта використання ресурсів Pod'ами, які були запущені на серверах спільного хостингу, але створені клієнтами, що орендують виділені сервери;

– мінімізація вільного обсягу обчислювальних ресурсів на сервері, на який призначається Pod.

Обмеження, що визначені у моделях [3–4], є наступними:

– обмеження на використання ресурсів окремих серверів;

– обмеження на використання ресурсів окремими клієнтами;

– заборона на запуск Pod'ів на серверах спільного хостингу без відповідного дозволу на це клієнта;

– заборона на запуск Pod'ів на виділених серверах інших клієнтів;

– заборона на більш ніж однократний запуск та зупинку Pod'а;

– заборона на запуск Pod'а на більш ніж одному сервері;

– заборона на запуск Pod'а на вимкненому сервері.

У відповідних публікаціях [3–4] зазначається необхідність розробки інформаційної технології, що покращить продуктивність системи з точки зору описаних у попередньому розділі критеріїв, на основі відповідних математичних моделей. Зазначається також потреба у проведенні експериментів із визначення ефективності інформаційної технології. У наступних розділах надаються пропозиції щодо проєктування та розробки відповідної інформаційної технології із обґрунтуванням доцільності відповідних пропозицій.

Пропозиції щодо реалізації технології

Концептуально, побудова розкладу у Kubernetes полягає у прийнятті рішень щодо призначень Pod'ів на вузли (сервери), причому кожне таке рішення складається з двох етапів: фільтрації (визначення множини допустимих вузлів) та скорингу (вибір найкращого вузла) [3–4, 8–9].

Kubernetes дозволяє підключати власні плагіни до вбудованого модуля *kube-scheduler* або використовувати власний компонент складання розкладів замість *kube-scheduler* [1]. Плагіни *kube-scheduler* можуть містити додаткову логіку як для етапу фільтрації, так і для етапу скорингу [10–11]. Відповідно, це надає можливість власникам кластерів покращувати продуктивність кластерів із урахуванням їхніх потреб.

Щоразу, коли до системи надходить новий Pod, пропонується знаходити оптимальний сервер для нього за допомогою статичної математичної моделі [4]. Щоразу, коли навантаження у кластері зростає чи спадає до певного рівня, пропонується визначати сервери, які необхідно вмикати та вимикати, за допомогою динамічної математичної моделі [3]. Пропонується, щоб за певними ознаками інформаційна технологія визначала недостатність чи надмірність поточного рівня ресурсів на увімкнених серверах, і відповідно до цього запускала модель. Після запуску моделі та знаходження нею серверів до увімкнення чи вимкнення, технологія має змінювати статус відповідних серверів (вмикати їх, або вивільняти ресурси та вимикати їх, відповідно). До ознак недостатності або надмірності рівня ресурсів пропонується віднести:

– частку зайнятих обчислювальних ресурсів (ядер та пам'яті) на увімкнених серверах, що виходить за межі певного діапазону нормального навантаження;

– високу середню швидкість («різкість») зміни навантаження протягом короткого періоду часу.

Конкретні значення параметрів (інтервалу нормальної частки зайнятих ресурсів та швидкості зміни навантаження) пропонується визначити експериментально. Окрім цього, експериментально пропонується визначити, чи є потреба запускати динамічну математичну модель раз на константний період часу, без виконання умови значної зміни рівня навантаження на кластер. Відповідні експерименти мають на меті знайти такі значення параметрів і таку частоту запуску моделі, за яких визначені цільові функції набували б найкращих значень (очікується, що занадто частий запуск моделі може погіршити ефективність кластера).

Пропонована модифікація процедури фільтрації пов'язана з неможливістю окремих Pod'ів бути розгорнутим на виділених серверах інших клієнтів або на серверах спільного хостингу, якщо клієнт не надав дозвіл запускати Pod на них. Пропонована модифікація процедури скорингу полягає у необхідності враховувати багато критеріїв та запобігати призначенню на сервери спільного хостингу Pod'ів, створених клієнтами, що орендують виділені сервери. Для відповідних модифікацій пропонується реалізація додаткових плагінів *kube-scheduler*.

Компонент *kube-scheduler* не має вбудованої можливості вмикати та вимикати вузли у кластері; натомість рішення про вимкнення та увімкнення вузлів кластера можуть прийматися автоматичним масштабувальником [12]. Kubernetes має вбудований функціонал, призначений для позбавлення та повернення можливості призначення нових Pod'ів на вузол [13–15]. Логіка автоматичного масштабування може бути реалізована в рамках оператора [16–17]. Відповідно, для регулювання вимиканням та увімкненням серверів (що може бути реалізовано на рівні абстракції як забезпечення того, що на вузол не буде призначений жодний Pod) пропонується використовувати оператори або наявні рішення з автоматичного масштабування.

Пропозиції щодо структури кластера

Відомо, що Kubernetes підтримує створення Pod'ів як для виконання одноразових (короткотермінових) задач (за

допомогою об’єктів типу Job), так і для постійного розгортання застосунків (за допомогою об’єктів типу Deployment) [3, 18]. Із урахуванням цієї інформації та припущення, що власник кластера не може самостійно вмикати та вимикати виділені сервери – пропонується додатково категоризувати всі увімкнені спільні хостингові сервери на дві категорії. На серверах однієї з цих категорій можуть бути розміщені всі Pod’и для виконання одноразових задач, на серверах іншої категорії – Pod’и для постійного розгортання застосунків.

Завдяки цій категоризації, пропонується запобігти ситуації, коли знижується навантаження за рахунок завершення виконання одноразових задач, але сервер все ще використовується іншими Pod’ами і його не можна вимкнути. Категоризуючи сервери та групуючи одноразові задачі на одному сервері (в ідеалі – розподіляючи їх за серверами відповідно до очікуваного моменту завершення або очікуваної тривалості виконання), пропонується повністю звільняти окремі сервери від навантаження та своєчасно вимикати їх, що покращить енергоефективність ЦОДу.

Висновки

У тезах доповіді наводяться основні результати попередньої побудови оптимізаційних математичних моделей для задачі управління ресурсами та складання розкладів для кластерів хмарних центрів обробки даних, що містять як виділені сервери, так і сервери спільного хостингу. Наводяться пропозиції щодо програмної реалізації технології, що має на меті покращити продуктивність кластерів, та щодо структури кластера, серед яких можна відзначити використання плагінів для модифікації логіки фільтрації та скорингу, використання операторів для підтримки вимкнення й увімкнення вузлів кластера, та категоризацію вузлів у кластері. Відзначена необхідність подальшого експериментального визначення частоти запуску динамічної математичної моделі.

Література

1. Коваленко В. В. Методи та моделі складання розкладів для оркестратора Kubernetes / В. В. Коваленко, М. М. Букасов // Вісник Вінницького політехнічного інституту. – 2024. – Т. 175, № 4. – С. 86–94.

2. Adoption of cloud computing as innovation in the organization / L. Golightly [et al.] // International Journal of Engineering Business Management. – 2022. – Vol. 14.

3. Kovalenko V. Dynamic mathematical model for resource management and scheduling in cloud computing environments / V. Kovalenko, O. Zhdanova // Information, Computing and Intelligent systems. – 2024. – No. 5. – P. 90–100.

4. Коваленко В. Статична математична модель складання розкладів у середовищах хмарних обчислень / В. Коваленко // Інформаційні системи та технології: результати і перспективи : Матеріали 2-ої Міжнар. науково-практ. конф. (Київ, 5 берез. 2025 р.) – Київ, 2025. – С. 233–236.

5. Carrión C. Kubernetes as a Standard Container Orchestrator – A Bibliometric Analysis / C. Carrión // Journal of Grid Computing. – 2022. – Vol. 20, no. 4.

6. Guruge P. B. Time series forecasting-based Kubernetes autoscaling using Facebook Prophet and Long Short-Term Memory / P. B. Guruge, Y. H. P. P. Priyadarshana // Frontiers in Computer Science. – 2025. – Vol. 7.

7. Boreas – A Service Scheduler for Optimal Kubernetes Deployment / T. Lebesbye [et al.] // Service-Oriented Computing, 22–25 November 2021. – 2021. – P. 221–237.

8. An Improved Kubernetes Scheduling Algorithm for Deep Learning Platform / Shi Huaxin [et al.] // 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (Chengdu, 18–20 December 2020). – Chengdu, 2020.

9. https://kubernetes.io/docs/concepts/scheduling-eviction/scheduling-framework/ (дата звернення 09.05.2025).

10. Taming latency at the edge: A user-aware service placement approach / C. Centofanti [et al.] // Computer Networks. – 2024. – P. 110444.

11. Wang X. Optimization of Task-Scheduling Strategy in Edge Kubernetes Clusters Based on Deep Reinforcement Learning / X. Wang, K. Zhao, B. Qin // Mathematics. – 2023. – Vol. 11, no. 20. – P. 4269.

12. https://kubernetes.io/docs/concepts/cluster-administration/node-autoscaling/ (дата звернення 10.05.2025).

13. https://kubernetes.io/docs/reference/kubectl/generated/kubectl_cordon/ (дата звернення 10.05.2025).

14. https://kubernetes.io/docs/tasks/administer-cluster/safely-drain-node/ (дата звернення 10.05.2025).

15. https://kubernetes.io/docs/reference/kubectl/generated/kubectl_uncordon/ (дата звернення 10.05.2025).

16. Yuan H. A Time Series-Based Approach to Elastic Kubernetes Scaling / H. Yuan, S. Liao // Electronics. – 2024. – Vol. 13, no. 2. – P. 285.

17. Toward Optimal Load Prediction and Customizable Autoscaling Scheme for Kubernetes / S. K. Mondal [et al.] // Mathematics. – 2023. – Vol. 11, no. 12. – P. 2675.

18. Burns B. Kubernetes: Up and Running. Dive into the Future of Infrastructure – 2nd ed. / B. Burns, J. Beda, K. Hightower. – Sebastopol, CA, USA: O’Reilly Media, Inc., 2019. – 255 p.

Telemetry-enhanced adaptive Bloom filter for distributed systems with dynamic hash functions adjustment

Shutenko Viktor, Bohdan Zhurakovskiy
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
victor.shutenko@gmail.com, bogdan68@ukr.net

Abstract. The article **propose the Telemetry-Enhanced Adaptive Bloom Filter (TEABF), a novel probabilistic membership structure that dynamically adjusts the number of hash functions (k) and bitmap size (m) based on real-time host telemetry—including access frequency, key distribution, CPU load, and network latency. In experiments on a three-node Redis cache cluster, TEABF achieves a 30–50 % reduction in false positive rate (FPR) and an 11 % decrease in average query latency compared to state-of-the-art adaptive filters.**

Keywords**: *Bloom filter; adaptive caching; distributed systems.***

Телеметрично-покращений адаптивний Bloom-фільтр для розподілених систем з динамічною зміною кількості хеш-функцій

Шутенко Віктор, Жураковський Богдан
КПІ імені Ігоря Сікорського
Київ, Україна
victor.shutenko@gmail.com, bogdan68@ukr.net

Анотація. **В роботі запропоновано Telemetry‑Enhanced Adaptive Bloom Filter (TEABF) — ймовірнісну структуру даних, яка в режимі реального часу коригує кількість хеш‑функцій (k) та розмір бітового масиву (m) на основі телеметрії хост‑системи (частота звернень, розподіл ключів, завантаження CPU, мережеві затримки). Експериментально доведено зниження імовірності хибних позитивних відповідей на 30–50 % та 11 % скорочення середнього часу відповіді у порівнянні з існуючими адаптивними фільтрами.**

Ключові слова: Bloom‑фільтр; адаптивне кешування; телеметрія; розподілені системи.

Вступ

В сучасних розподілених системах з високим навантаженням (соціальні мережі, інтернет-магазини, фінансові платформи) обсяг одночасних запитів до кешу може сягати сотень тисяч на секунду. Ефективне фільтрування запитів перед зверненням до бекенду дозволяє суттєво знизити затримки та навантаження на базу даних.

Bloom‑фільтр — широко застосовувана ймовірнісна структура даних, яка забезпечує O(1) час вставки та перевірки належності елемента до множини. Однак класичні реалізації з фіксованими параметрами (кількість хеш‑функцій k, розмір бітового масиву m) не враховують змінні патерни навантаження та системні ресурси, що призводить до підвищеної ймовірності хибних позитивів або неефективного використання пам’яті.

Існуючі підходи (Adaptive Bloom Filters, Scalable BF) коригують параметри на основі внутрішніх характеристик фільтра (числа елементів, FPR), але ігнорують зовнішні фактори: навантаження CPU, мережеві затримки, розподіл «гарячих» ключів. Впровадження телеметричних метрик дає змогу врахувати реальний стан системи й підвищити точність прогнозу оптимальних параметрів.

Метою дослідження є розробка Telemetry‑Enhanced Adaptive Bloom Filter — Bloom‑фільтру з адаптивною корекцією параметрів на підставі системної телеметрії.

Огляд bloom фільтрів

Bloom-фільтр — це ймовірнісна структура даних для перевірки належності елементів множині. Bloom-фільтр вперше запропонований Бертоном Блумом у 1970 році як ефективний метод перевірки належності елемента до множини з мінімальними витратами пам'яті та гарантованим нульовим рівнем хибних негативів. Ідея полягає в тому, щоб замінити повноцінне зберігання всіх ключів компактним бітовим масивом і набором хеш-функцій: при вставці елемента відповідні біти встановлюються в 1, а при перевірці — якщо хоча б один із цих бітів дорівнює 0, елемент гарантовано відсутній. Складається з бітового масиву розміру m та k незалежних хеш-функцій.

Операції:

- вставка: для кожного з k хешів елементу обчислюють позицію в бітовому масиві й встановлюють відповідні біти в 1.
- перевірка: аналогічно обчислюють k позицій; якщо хоча б одне з обраних бітів рівне 0 — елемент точно відсутній, якщо всі біти 1 — можливо присутній (можливий хибнопозитивний результат).

Перевагами є компактне зберігання, постійний час вставки та перевірки незалежно від кількості елементів. Обмеження - хибні позитивні відповіді, відсутність механізму видалення (без використання лічильників), потреба налаштування параметрів (k, m) для контролю ймовірності помилок.

Теоретично імовірність хибного позитиву (False Positive Rate) для Bloom-фільтра з розмірами бітового масиву m, числам елементів n і кількістю хеш-функцій k може бути наближеною формулою 1

$$FPR \approx \left(1 - e^{-kn/m}\right)^k \quad (1)$$

а оптимальне k для мінімізації FPR за даних m і n обчислюється як $k = \frac{m}{n} ln2$. Таким чином, змінюючи співвідношення m/n і k, можна гнучко контролювати компроміс між пам'яттю та точністю.

Bloom-фільтри широко застосовуються в дистрибутивних системах (наприклад, для попередньої фільтрації запитів у базах даних або кешах), у мережевих протоколах (IDS/IPS), у пошукових індексах та інших завданнях, де важлива швидкість перевірки належності за незмінно низьких накладних витрат.

Види Bloom Filters:

1) Scalable Bloom Filters: модель динамічного розширення через ланцюжок фільтрів [1]. Забезпечує контроль FPR, але без урахування системних метрик.
2) Dynamic Bloom Filters: автоматичне збільшення ємності при рості числа елементів[2]. Не враховують розподіл запитів у часі.
3) Adaptive Bloom Filters (ABF): змінюють k залежно від частоти появи ключів у мережевому потоці [3]. Немає інтеграції з хост-показниками.
4) Learned Bloom Filters: використовують ML-моделі для передбачення ключів і додатковий фільтр [4]. Потребують попереднього тренування та не адаптуються до змін у реальному часі.

Запропонований підхід

Алгоритм роботи TEABF починається зі збору та нормалізації телеметрії щодо частоти звернень до кешу, завантаження CPU, використання пам'яті та мережевих затримок. На основі цієї інформації система формує згладжену оцінку «гарячості» ключів і визначає цільову імовірність хибних позитивів згідно з поточним навантаженням. Далі обираються оптимальні значення кількості хеш-функцій і розміру бітового масиву, які забезпечують баланс між швидкістю обробки запитів та економією пам'яті. У разі виявлення значущої різниці між новими та поточними параметрами запускається фонове створення нового фільтра: елементи переноситимуться до резервної структури без переривання обслуговування. Після завершення побудови відбувається атомарне перемикання вказівника активного фільтра на оновлену версію, а стара структура поступово очищується та готується до наступного циклу адаптації.

Telemetry-Enhanced Adaptive Bloom Filter (TEABF) побудовано за модульним принципом. Спершу модуль збору телеметрії централізує інформацію про частоту звернень до кешу, завантаження центрального процесора, обсяг використаної оперативної пам'яті та затримки мережевих з'єднань. Далі блок аналізу нормалізує ці дані, формуючи єдину метрику «гарячості» ключів, і вираховує цільову ймовірність хибних позитивів із урахуванням поточного стану системи. Наступним кроком механізм адаптації розраховує оптимальні параметри фільтра — кількість хеш-функцій та розмір бітового масиву — використовуючи класичні формули Bloom-фільтра, скориговані залежно від рівня навантаження й частки «гарячих» запитів.

У разі якщо розбіжність між новими параметрами та поточними перевищує встановлені порогові значення, TEABF ініціює фонове створення відтворюваної копії фільтра. Елементи переноситимуться до оновленої структури без перешкоджання активному обслуговуванню запитів; за завершення перенесення відбувається миттєве перемикання на нову версію за допомогою атомарної операції. Стара інстанція фільтра очищується та переходить у режим очікування наступного циклу адаптації.

Для забезпечення надійності та прозорості роботи передбачено комплекс заходів з моніторингу та журналювання. Всі ключові метрики — від FPR до часу відповіді та пропускної здатності — відображаються в інструментах візуалізації (наприклад, Grafana). Поряд із цим система фіксує зміни параметрів і результати їх валідації в лог-файлах, а у випадку аномальних відхилень автоматично відкотиться до останніх стабільних налаштувань.

Унікальною рисою TEABF є його розширюваність та гнучкість налаштувань: адміністратор може змінювати інтервали збору телеметрії та порогові значення адаптації без перезапуску сервісу. Це дозволяє балансувати між швидкістю реакції фільтра і накладними витратами на обробку даних, адаптуючи систему під специфічні робочі навантаження та бізнес-вимоги.

Експериментальна частина

Для перевірки продуктивності TEABF використано Redis-контур із трьох вузлів із апаратною конфігурацією: процесори Intel Xeon 2.4ГГц, 32ГБ ОЗП, мережевий

інтерфейс 10Гбіт/с. Вузли об'єднані комутаторами з мінімальними затримками (<0.5мс).

Навантаження моделювалося за допомогою інструменту WRK, який відправляє 100000 запитів/с із випадковим розподілом ключів. 70% запитів спрямовано до «холодних» (нових) ключів, 30% — до «гарячих», обраних із попереднього журналу звернень.

Таблиця 1

Результати замірів

Метод	***False Positive Rate***	***Час відповіді(мс)***	***Пропускна здатність(запитів/с)***
ABF	2,5%	1,8	95 000
Scalable BF	2,8%	1,9	94 000
TEABF	1,2%	1,6	98 450

Було заміряно такі показники: ймовірність хибних позитивів (FPR) шляхом порівняння відповідей фільтра з реальною присутністю ключів; середній час відповіді клієнтського запиту; пропускну здатність кешу в запитах/с. Метрики збиралися через Prometheus-агенти з інтервалом 1с. Кожен експеримент проводився в три фази: прогрів (30с), вимірювання (60с) та охолодження (30с). Усі тести повторювалися тричі для усереднення результатів(табл 1).

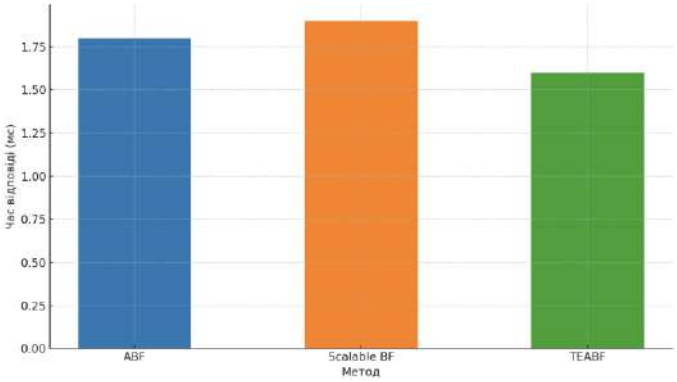

Рис. 1. Порівняння часу відповіді різних методів

Результати можна звести до наступних висновків:

- TEABF знизив FPR на ~50 % відносно ABF завдяки адаптації k до фактичного «гарячого» навантаження.
- Час відповіді скоротився на ~11 % за рахунок меншої кількості хибних звернень до бекенду (рис. 1)
- Пропускна здатність збільшилася, оскільки зменшено навантаження на мережу та БД.

Висновки

У роботі представлено Telemetry-Enhanced Adaptive Bloom Filter (TEABF) — інноваційну структуру даних, здатну динамічно підлаштовувати параметри (кількість хеш‑функцій та розмір бітового масиву) на основі системної телеметрії. Експериментальні результати показали зниження ймовірності хибних позитивів на понад 50 % та скорочення середнього часу відповіді на 11 %, що підтверджує ефективність підходу. Запропонована методика плавного оновлення через подвійне буферування забезпечує відсутність простоїв під час адаптації. Таким чином, TEABF є перспективним рішенням для високонавантажених розподілених систем із вимогами до мінімізації затримок та оптимального використання пам'яті.

Додатково, впровадження TEABF дозволяє скоротити витрати на інфраструктуру за рахунок зменшення навантаження на бекенд-системи та мережу, що в кінцевому підсумку покращує масштабованість архітектури. Висока гнучкість налаштувань робить TEABF придатним для широкого спектру застосувань — від комерційних хмарних платформ до телекомунікаційних мереж.

Література

1. Almeida P., Baquero C., Preguiça N., Hutchison D. *Scalable Bloom Filters*. IEEE Transactions on Knowledge and Data Engineering, 19(3): 730–740, 2007.
2. Lu H., Dong G., Yuan L. *Dynamic Bloom Filters*. ACM Transactions on Database Systems, 33(3): 15, 2008.
3. Chen X., Gao J. *Adaptive Bloom Filters for Network Applications*. Proceedings of the 2015 USENIX Annual Technical Conference, Santa Clara, CA, USA, 2015.
4. Kraska T., Beutel A., Chi E. H., Dean J., Polyzotis N. *The Case for Learned Index Structures*. Proceedings of ACM SIGMOD International Conference, 2018.

ШТУЧНИЙ ІНТЕЛЕКТ

ARTIFICIAL INTELLIGENCE

Computer Vision-Based Approach for UAV Landing Zone Identification

Danylo Novykov, Viktoriia Onyshchenko
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
danilnovikov759@gmail.com, v.onyshchenko@kpi.ua

Abstract. The aim of this study is to develop a computer vision–based model using YOLOv5s for the autonomous identification of safe UAV landing zones from onboard camera images. Several model modifications were proposed and compared, and evaluated on urban scenes with variations in lighting, noise, and geometric distortions. In the best case, improvements in correct detections, precision, and recall were achieved: Precision = 0,938, Recall = 0,8231, F_1 = 0,8768.

***Keywords*: object detection, image processing, unmanned aerial vehicles, UAV, autonomous landing, autonomous delivery.**

Підхід на основі комп'ютерного зору для визначення зони посадки БПЛА

Новиков Данило, Онищенко Вікторія
КПІ імені Ігоря Сікорського
м.Київ, Україна
danilnovikov759@gmail.com, v.onyshchenko@kpi.ua

Анотація. Метою дослідження є розробка моделі комп'ютерного зору на базі YOLOv5s для автономного визначення безпечної зони посадки БПЛА за зображеннями з бортової камери. Запропоновано та порівняно кілька модифікацій моделі, які були апробовані на урбаністичних кадрах із варіаціями освітлення, шуму та геометричних спотворень. У найкращому випадку досягнуто покращення кількості коректних розпізнавань, точності та повноти, а саме: Precision = 0,938, Recall = 0,8231, F_1 = 0,8768.

Ключові слова: виявлення об'єктів, обробка зображень, безпілотні літальні апарати, БПЛА, автономна посадка, автономна доставка.

Вступ

У сучасному світі логістика стає дедалі складнішою через постійне зростання обсягів замовлень, які потребують швидкої та надійної доставки. Інтеграція безпілотних літальних апаратів (БПЛА) у процес останньої милі дозволяє звільнити людські ресурси для виконання складніших завдань та покращити досвід кінцевого споживача.

Система автономної доставки вантажів за допомогою БПЛА умовно складається з наступних модулів:

- Навігація – планування маршруту польоту й руху;
- Визначення кінцевої точки доставки;
- Механізм фіксації та вивільнення вантажу.

Ця робота зосереджена на модулі визначення безпечної зони посадки, яка фактично є кінцевою точкою в ланцюгу доставки. Наразі не існує універсальної системи, здатної автоматично визначати зону посадки БПЛА у типовому міському середовищі – з урахуванням приватних будинків, офісних будівель та багатоповерхових житлових комплексів.

Мета дослідження – розробка моделі на основі комп'ютерного зору для виявлення безпечних майданчиків на відкритих ділянках приватних або громадських просторів із достатнім вільним простором. Для цього у якості ключового тренувального сигналу обрано спеціалізовані маркери, що чітко визначають межі зони посадки.

Слід також зазначити, що більшість існуючих рішень покладаються на GPS‑сигнал для уточнення місцезнаходження БПЛА. Однак GPS може бути неточним (типова похибка GPS ~2–5 м) або зовсім недоступним. Використання ж алгоритмів комп'ютерного зору забезпечує незалежність від супутникового зв'язку та підвищує надійність системи в критичних умовах, мінімізуючи ризики під час доставки.

Аналіз останніх досліджень і публікацій

Аналіз сучасної літератури свідчить про активний розвиток теми автономної доставки за допомогою БПЛА, вод-

ночас виявляючи низку обмежень існуючих підходів. У дослідженні [1] пропонується алгоритм пошуку безпечної зони посадки, що ґрунтується на виявленні людини як потенційного отримувача посилки за допомогою методів комп'ютерного зору. Такий підхід корисний, коли отримувач присутній на місці, але непридатний для випадків безконтактної доставки на приватну територію, де людина може бути відсутня.

Автори дослідження [2] пропонують комплексний підхід до виявлення безпечної зони посадки без використання маркерів для екстреної посадки на непідготовленій ділянці. Проте середній час обробки одного кадру (600×400 пікселів) становить близько 2,5 с, що ускладнює його застосування в режимі реального часу, де швидкодія критично важлива.

Серед комерційних прикладів слід відзначити Amazon Prime Drone Delivery [3]. Незважаючи на значний прорив у сфері автономної безпілотної доставки, система має обмеження. Зокрема, посадкова зона повинна бути попередньо визначена користувачем на основі супутникового знімка його двору, що створює залежність від GPS-сигналу. Додатково, БПЛА не приземляється, а скидає посилку з висоти 3,5 м, обмежуючи тип вантажів і знижуючи універсальність рішення.

Зазначені приклади підкреслюють актуальність розробки нового підходу, орієнтованого на підвищення продуктивності та автономності (незалежність від системи GPS) при збереженні високої точності.

Виклад основного матеріалу

В якості базової моделі обрано YOLOv5s [4] як сучасну модель комп'ютерного зору що поєднує гарну швидкість детекції та точність з відносно невеликою обчислювальною складністю та простотою налаштування. Для навчання використано набір тренувальних даних міського середовища з платформи Roboflow [5], у якому частина зображень містить спеціалізовані маркери, що позначають безпечну зону посадки і які в контексті дослідження виступають ціллю розпізнавання. Під час першого раунду тренування з використанням стандартних гіперпараметрів отримано контрольні результати: Precision (точність) = 0,781; Recall (повнота) = 0,889; mAP@0,5 = 0,851; mAP@0,5:0,95 = 0,538. Ці показники підтвердили ефективність базової конфігурації на ідеалізованих урбаністичних зображеннях.

Для підвищення стійкості до варіацій реального середовища розширено пайплайн обробки даних за допомогою бібліотеки Albumentations. Із ймовірністю 60% обирається та застосовується до зображення один із трьох методів контрастної/кольорової корекції: CLAHE (адаптивне вирівнювання гістограми з обмеженням контрасту), ToGray (перетворення в чорно-біле) або Equalize (еквалізація гістограми зображення). Тренування модифікованої моделі дало наступні результати: Precision = 0,884; Recall = 0,847; mAP@0,5 = 0,949; mAP@0,5:0,95 = 0,463. Результат навчання модифікованої версії демонструє помітне зростання точності й mAP@0,5 за рахунок незначного зниження повноти та mAP@0,5:0,95.

Щоб оцінити адаптивність обох моделей до реальних умов, проведено апробацію на тестовому наборі тренувальних даних [6], що включає: стандартні кадри (a), зображення зі зміненою яскравістю (±50%) (b, c), з Gaussian-розмиттям (d), шумом (e), чорно-білі (f) версії та повернуті на ±45° (g). Приклад тестового зображення і його модифікацій наведено на рис. 1:

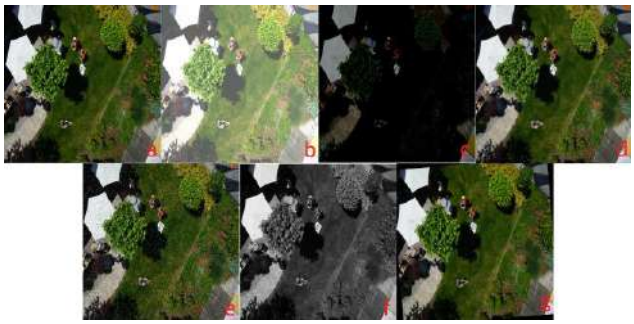

Рис. 1. Приклад тестового набору тренувальних даних

Оскільки метою цієї роботи є вибір кінцевої точки посадки за найвищою ймовірністю детекції маркеру, після розпізнавання кінцевою точкою вважається область з найбільшою впевненістю. Для порівняння базової й модифікованої версій моделей проведено три фази тестування з різними мінімальними порогами впевненості: без порогу, з порогом 50% і з порогом 75%. Для розрахунку ключових метрик використано:

- True Positives (TP) – кількість коректно виявлених маркерів;
- False Positives (FP) – сума випадків дублікатів детекції одного маркеру та некоректних передбачень;
- False Negatives (FN) – кількість маркерів, що не були виявлені.

Метрики Precision (1), Recall (2) і F_1 (3), гармонійне середнє точності та повноти, обчислювалися за стандартними формулами.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$F_1 = 2\frac{Precision \times Recall}{Precision+Recall} \quad (3)$$

Результати апробації наведено в табл. 1. Базову модель позначено як default, модифіковану – modified, через тире вказано мінімальний поріг впевненості, якщо застосовано. Модифікована модель у порівнянні з базовою демонструє:

- без порогу конфіденційності – зростання кількості коректних детекцій на 43,5% (99 проти 69 TP);
- при порозі 50% – в 13,8 раз більше коректних виявлень (83 проти 6 TP);
- при порозі 75% – базова модель не виявила жодного маркеру, тоді як модифікована – 7 зон.

Таблиця 1

Результат апробації базової (default) та модифікованої (modified) моделей

Назва моделі	TP	FP	FN	P	R	F_1
default	69	5	78	0,9324	0,4694	0,6244
modified	99	23	48	0,8115	0,6735	0,7361
default–50	6	0	141	1	0,408	0,0784
modified–50	83	3	64	0,9651	0,5646	0,7124
default–75	0	0	147	0	0	0
modified–75	7	0	140	1	0,0476	0,0909

Отримані результати свідчать, що запропонована стратегія аугментації значно підвищує продуктивність моделі в умовах варіативності освітлення, шумів та геометричних спотворень, зберігаючи прийнятний рівень швидкодії для застосування в реальному часі.

Хоча модифікована модель показала значне покращення порівняно з базовою, для підвищення її здатності до

узагальнення та покриття нестандартних ситуацій було проведено раунд донавчання (fine-tuning) із оптимізованими гіперпараметрами, використовуючи ваги з першого етапу. Результати наведено в табл. 2. Було опрацьовано три конфігурації:

- default* – базова модель, донавчена з початкових ваг (з першого раунду) без змін гіперпараметрів (контрольна версія);
- ft-default – базова модель, донавчена з новими підібраними гіперпараметрами;
- ft-modified – модифікована модель із розширеною аугментацією, донавчена з новими гіперпараметрами.

Таблиця 2

Показники моделей після донавчання

Назва моделі	**Precision**	**Recall**	**mAP@0,5**	**mAP@0,5:0,95**
default*	1	0,889	0,975	0,588
ft-default	0,852	1	0,995	0,635
ft-modified	1	0,96	0,995	0,667

Також проведено апробацію отриманих моделей за різними порогами впевненості на тестовому наборі даних. Результати апробації наведені в табл. 3:

Таблиця 3

Результат апробації після донавчання

Назва моделі	**TP**	**FP**	**FN**	**P**	**R**	**F_1**
default*	100	15	47	0,8696	0,6803	0,7634
ft-default	133	400	14	0,2495	0,9048	0,3911
ft-modified	132	263	15	0,3342	0,898	0,4871
default*–50	65	4	82	0,942	0,4422	0,6019
ft-default–50	124	87	23	0,5877	0,8435	0,6927
ft-modified–50	125	60	22	0,6757	0,8503	0,753
default*–75	2	0	145	1	0,0136	0,0268
ft-default–75	119	13	28	0,9015	0,8095	0,853
ft-modified–75	121	8	26	0,938	0,8231	0,8768

За результатами використання гіперпараметрів можна зробити наступні висновки:

- Навчання з оптимізованими гіперпараметрами (ft-default) підвищує кількість коректних розпізнань (TP), але погіршує precision через збільшення кількості помилкових спрацьовувань (FP) за умови низького мінімального порогу впевненості;
- Контрольна версія (default*) демонструє збалансованішу роботу у порівняні зі стандартною моделлю (default), проте поступається порівняно з модифікованою (modified) та fine-tuned (ft-*) моделями;
- Модифікована модель з оптимізованими гіперпараметрами (ft-modified) поєднує високий precision і recall, досягаючи найкращого mAP@0,5:0,95 та F_1 серед усіх конфігурацій;
- Зіставлення фаз із порогами 50% і 75% підтверджує здатність ft-моделей ефективно відкидати хибні спрацьовування без втрати критичного числа коректних детекцій.

Висновки

У цій роботі запропоновано алгоритм, який використовує лише зображення з бортової камери БПЛА для визначення безпечної зони посадки за допомогою спеціалізованих маркерів. Модель YOLOv5s була модифікована для детекції таких маркерів, проведено її навчання та тестування на різних тестових наборах даних. Основні результати:

- Базова модель без аугментації практично не виявляє маркери при встановленому наближеному до реальних умов порозі мінімальної впевненості;
- Аугментація даних на рівні моделі значно підвищує стійкість моделі до змін освітлення, шуму та геометричних спотворень, зберігаючи прийнятний час виконання;
- Підбір оптимальних гіперпараметрів покращує показники моделі порівняно зі стандартним навчанням;
- Поєднання аугментації з оптимальними гіперпараметрами забезпечує збалансоване співвідношення між точністю та повнотою;
- За мінімального порога впевненості на рівні 75% ft-modified демонструє найкращу комбінацію показників: P = 0,938; R = 0,8231 та F_1 = 0,8768.

Для практичного застосування обрано модель ft-modified як квінтесенцію найкращих метрик та апробації. Запропонований підхід не вимагає використання GPS-сигналу, що робить його придатним для умов з відсутнім або заглушеним супутниковим зв'язком.

Застосування запропонованої моделі в поєднанні з алгоритмами навігації створює передумови для створення автономної інтелектуальної системи доставки останньої милі за допомогою БПЛА, що може значно підвищити ефективність і надійність процесу доставки вантажів.

Література

1. Safadinho D. UAV Landing Using Computer Vision Techniques for Human Detection / D. Safadinho et al. // Sensors. – 2020. – vol. 20, is. 3.

2. Turan V. Image processing based autonomous landing zone detection for a multi-rotor drone in emergency situations / V. Turan et al. // Turkish Journal of Engineering. – 2021. – vol. 5, is. 4. – P. 193-200.

3. Gutierrez P. Amazon Prime Drone Delivery at XPONENTIAL Europe / P. Gutierrez // Inside Unmanned Systems. – 2025. – URL: https://insideunmannedsystems.com/amazon-prime-drone-delivery-at-xponential-europe (access date: 19.04.2025).

4. Jocher G. YOLOv5 by Ultralytics (Version 7.0) [Computer software] / G. Jocher et al. // zenodo. – 2022. – URL: https://doi.org/10.5281/zenodo.3908559 (access date: 19.04.2025).

5. Drone Urban Computer Vision Project. Roboflow. – 2022. – URL: https://universe.roboflow.com/bits-pilani-hyderabad/drone-urban (access date: 19.04.2025).

6. AR-marker evaluation Computer Vision Project. Roboflow. – 2025. – URL: https://universe.roboflow.com/test-tqcr4/ar-marker-evaluation (access date: 28.04.2025).

Method for Wheat Disease Detection and Recognition Using Image Analysis

Serhii Smovzh, Halushko Dmytro
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

Abstract. **The paper investigates the use of deep neural networks for automatic detection and recognition of wheat diseases based on images. A review of state-of-the-art approaches was conducted and an effective system was implemented using the EfficientNetV2B0, ConvNeXtTiny and U-Net models. The average classification accuracy for all 13 classes was 88%, while for some classes, the accuracy reached 94-95%, which is a high result. The use of XAI (Grad-CAM) ensured the interpretability of the model's solutions, increasing its reliability and suitability for practical use in the agricultural sector.**

Keywords*: *Deep learning, Image recognition, Disease detection, Agriculture, Phytopathology.

Метод виявлення та розпізнавання хвороб пшениці на основі аналізу зображень

Смовж Сергій, Галушко Дмитро
КПІ імені Ігоря Сікорського
Київ, Україна

Анотація. **У роботі досліджено застосування глибоких нейронних мереж для виявлення та розпізнавання захворювань пшениці на основі зображень. Проведено огляд сучасних підходів і реалізовано ефективну систему з використанням моделей EfficientNetV2B0, ConvNeXtTiny та U-Net. Середня точність класифікації по всіх 13 класах склала 88 %, при цьому для окремих класів, точність досягала 94 – 95 %, що є високим результатом. Використання XAI забезпечило інтерпретованість рішень моделі, підвищуючи її надійність і придатність до практичного застосування в аграрному секторі.**

Ключові слова: Глибоке навчання, Розпізнавання зображень, Виявлення захворювань, Сільське господарство, Фітопатологія.

Вступ

Нещодавній стрімкий розвиток штучного інтелекту та глибокого навчання надав нові можливості для вирішення проблем у багатьох галузях, зокрема в сільському господарстві. Значно полегшити життя фермерів може своєчасне та якісне виявлення та розпізнавання хвороб рослин, які є надзвичайно важливими для продовольчої безпеки та сталого розвитку сільського господарства. Пшениця є однією з основних сільськогосподарських культур, яка перебуває під постійною загрозою низки хвороб, що спричиняють значні втрати врожайності [1]. Раннє і точне виявлення цих хвороб має вирішальне значення, оскільки їхня профілактика повинна бути спрямована на уникнення значних втрат врожаю за допомогою управлінських заходів, що максимально обмежують шкоду завдану хворобами у найкоротші терміни.

Сучасні дослідження в галузі сільського господарства продемонстрували значний потенціал штучного інтелекту [2] в аналізі таких культур, як рис, картопля, кава, цукрова тростина та інші [3].

Представлено та порівняно результати навчання різних CNN, розглянуто найпоширеніші хвороби пшениці, проаналізовано існуючі підходи для вирішення аналогічних проблем (ResNet, EfficientNet, Inception [4], VGG), доступні набори даних та описано стандартні методи аналізу та ідентифікації заражених рослин. Дослідження фокусує увагу на доцільності використання Explainable AI [5] для досягнення поставлених завдань і вказує на перспективні напрямки подальшого розвитку.

Дані та їх попередня обробка

Розглянуто розвиток згорткових нейронних мереж, однієї з найпопулярніших архітектур глибокого навчання, для виявлення та розпізнавання хвороб пшениці. Основними методами навчання CNN є вивчення великих наборів зображень з анотаціями, що містять назви хвороб, що дозволяє моделі автоматично класифікувати різні типи захворювань на основі візуальних симптомів.

Описано процес збору даних і методи їхньої попередньої обробки, а також розроблено архітектуру моделі CNN. Вимірювання продуктивності моделі проводилося шляхом

розрахунку accuracy, precision, recall, and F1-score на незалежному наборі даних, шляхом перехресної валідації.

У ході дослідження зібрано зображення пшениці, розділені на 12 категорій із захворюваннями та одну категорію, яка представляє здорову рослину. До типових уражень віднесено: бурі, жовті й чорні іржі, фузаріоз колоса, плямистості, борошнисту росу, тлю, стеблового комарика, пірикуляріоз та сажкові захворювання. Кожне з них має унікальні візуальні прояви: від пустул і бляшок до нальотів і деформацій.

Остаточний датасет сформовано на основі декількох відкритих джерел і очищено вручну. Серед розглянутих датасетів найбільш повним є Wheat Plant Diseases [6], який включає понад 14 тисяч зображень і до 15 класів хвороб пшениці. Також розглянуто Wheat Disease Detection [7] та Wheat Leaf Disease [8], що мають менший обсяг і кількість класів, однак часто використовуються як база для fine-tuning моделей. Було видалено дублікати, зображення низької якості, а також вирівняно баланс між класами шляхом цілеспрямованого добору прикладів. Усього зібрано 12 491 зображення, і розподілено між 13 класами. На рис. 1 наведено візуалізацію розподілу даних.

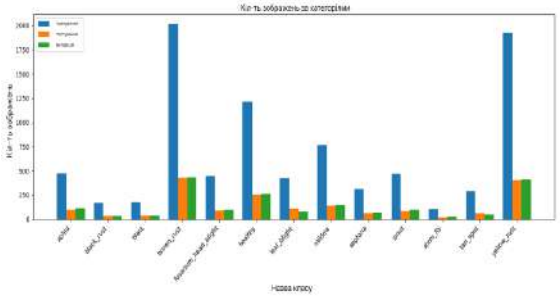

Рис 1. Розподіл даних

Проведений аналіз дозволив зібрати достатньо візуальних даних для повноцінного навчання, валідації та тестування системи, орієнтованої на реальні умови роботи в агропромисловому середовищі. Особливу увагу приділено хворобам, які мають подібні візуальні риси або зустрічаються рідко, оскільки саме вони становлять найбільший виклик для систем розпізнавання.

СУТНІСТЬ ЗАПРОПОНОВАНОГО МЕТОДУ

Після утворення датасету та використання його для початкового навчання моделі виникла проблема дисбалансу класів, який суттєво знижував точність визначення хвороб із малою кількістю зображень. Для подолання цієї проблеми було розроблено метод дворівневої класифікації. На етапі підготовки до використання на основі кількості зображень у тренувальній вибірці визначено групи класів: великі, середні, малі. У даному дослідженні до великих належали класи представлені 1000 зображень та більше, до середніх класи з кількістю зображень між 1000 та 300, всі інші були віднесені до малих. На першому етапі використано модель, яка вирішує до якої групи належить зображення, яке піддається аналізу. На другому етапі після того, як визначено групи, викликано відповідну модель, яка безпосередньо розпізнає конкретне захворювання всередині групи.

Для покращення ефективності розпізнавання візуально схожих або малочисельних класів додатково впроваджено функцію втрат Focal Loss і клас-специфічну аугментацію даних, що дозволило суттєво підвищити точність класифікації і майже повністю нівелювати вплив домінантних класів. У межах дослідження також було застосовано методи XAI для інтерпретації рішень класифікаційної моделі. Основну увагу зосереджено на використанні Grad-CAM, який дає можливість візуально локалізувати ділянки зображення, що найбільше вплинули на передбачення, що дозволило перевіряти чи знаходиться фокус на справді уражених ділянках листя та стебла, а не на фоновому шумі.

РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

Розроблений метод було протестовано та порівняно з іншими реалізаціями, експерименти довели, що поєднання архітектури EfficientNetV2B0 з методом ієрархічної класифікації показало найкращий результат.

Також у межах дослідження було застосовано методи напівконтрольованого навчання: спочатку моделі навчались на анотованих вручну даних, після чого прогнозували мітки для неанотованих зображень з умовою наявності високої впевненості моделі у рішенні (>90%). Це дозволило оптимізувати ресурси і отримати сегментований датасет без ручного маркування, результати навчання з використанням цього датасету показали приріст точності на 3–6% порівняно з попереднім навчанням на базі архітектури EfficientNetB0. На рис. 2 наведено графік порівняння методів на основі різних архітектур.

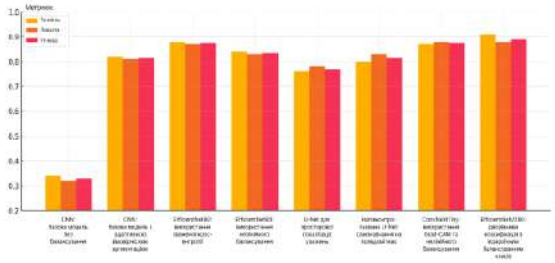

Рис 2. Порівняння експериментальних методів

Запропонований метод дворівневої класифікації дозволив підвищити точність роботи, що демонструється на графіку, підхід з двоетапною класифікацією майже повністю уникнув впливу домінантних класів, які наведено на рис.1, використання XAI додало прозорості роботі моделі. На рис. 3 зображено приклад роботи Grad-CAM, для кожної категорії ліворуч знаходиться оригінальне зображення з тестового датасету, по центру розташовано результат накладання теплової карти на оригінал, а праворуч власне саму теплову карту, яка демонструє зони, які модель вважає значними для висування прогнозу. Червоним кольором позначено найбільш важливу зону для моделі, а синім найменш значну.

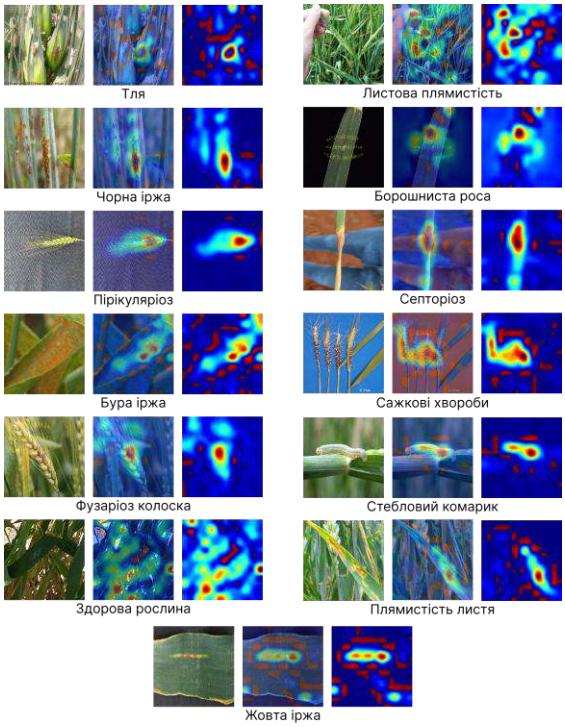

Рис 3. Використання XAI

Висновки

У цьому дослідженні було реалізовано, протестовано та порівняно низку архітектур глибокого навчання для задач виявлення та класифікації хвороб пшениці. Основна увага була зосереджена на підвищенні точності розпізнавання, особливо для малопредставлених класів, а також на інтерпретованості рішень моделі за допомогою Explainable AI.

Найвищу ефективність продемонстрував ієрархічний метод на основі EfficientNetV2B0, який за рахунок дворівневої класифікації дозволив досягти середньої тестової точності 88%. Значне покращення було досягнуто і для слабко представлених категорій, зростання recall у яких сягало 20 – 30% після впровадження Focal Loss та клас-специфічної аугментації.

Модель U-Net, натренована на сегментаційних масках, забезпечила високу якість локалізації уражень (IoU ≈ 0,99), дозволяючи використовувати її як попередній етап для подальшого підвищення точності класифікації. Крім того, було протестовано напівконтрольоване навчання з донавчанням моделі на неанотованих зображеннях, що дало приріст загальної точності класифікації до 6%.

Особливу роль у роботі відіграло застосування XAI – зокрема, методів Grad-CAM – для пояснення рішень моделі. Візуалізація уваги моделі на зображеннях кожного класу показала, що система фокусує розпізнавання саме на характерних симптомах хвороб, ігноруючи фонові або неінформативні області.

Таким чином, результати роботи підтверджують ефективність запропонованого методу в задачах виявлення хвороб пшениці за зображеннями, де є сильний дисбаланс класів. Метод демонструє високу точність, здатність до генералізації та пояснюваність рішень, що відкриває можливості для інтеграції подібних систем у реальні аграрні виробничі процеси. У подальших дослідженнях доцільним є фокус на мультиміткові сценарії, фазовість захворювання та роботу з реальними зображеннями з польових камер.

Література

1. Global Challange. *International Wheat Yield Partnership*. URL: https://iwyp.org/about-us/global-challenge/ (дата звернення: 01.03.2025).

2. Mohanty S. P., Hughes D. P., Salathé M. Using deep learning for image-based plant disease detection. *Frontiers in plant science*. 2016. Т. 7. URL: https://doi.org/10.3389/fpls.2016.01419 (дата звернення: 17.03.2025).

3. Sobuj M. S. I., Others. Leveraging pre-trained cnns for efficient feature extraction in rice leaf disease classification. *Proceedings of the international conference on advances in computing, communication, electrical, and smart systems (icaccess)*. 2023. URL: https://doi.org/10.48550/arXiv.2405.00025.

4. AI-powered banana diseases and pest detection / M. G. Selvaraj та ін. *Plant methods*. 2019. Т. 15, № 1. URL: https://doi.org/10.1186/s13007-019-0475-z (дата звернення: 17.03.2025).

5. Mehedi M. H. K., Others. Plant leaf disease detection using transfer learning and explainable AI. *2022 IEEE 13th annual information technology, electronics and mobile communication conference (IEMCON)*. Vancouver, BC, Canada, 2022. С. 166–170. URL: https://doi.org/10.1109/IEMCON56893.2022.9946513

6. Agarwal K., Yadav V., Suthar T. Wheat plant diseases. *Kaggle*. URL: https://www.kaggle.com/datasets/kushagra3204/wheat-plant-diseases (дата звернення: 20.03.2025).

7. Dunk S. Wheat disease detection. *Kaggle*. URL: https://www.kaggle.com/datasets/sinadunk23/behzad-safari-jalal (дата звернення: 20.03.2025).

8. KASHPONDY J. Wheat leaf disease. *Kaggle*. URL: https://www.kaggle.com/datasets/jayaprakashpondy/wheat-leaf-disease (дата звернення: 20.03.2025).

Efficiency of Reinforcement Learning Algorithms in the Mountain Car Task

Mykhailo Drahan, Andrii Pysarenko
Department of Information Systems and Technologies
Igor Sikorski Kyiv Polytechnic Institute
Kyiv, Ukraine
mykhailodrahan@gmail.com

Abstract. **This study explores the Deep Deterministic Policy Gradient (DDPG) algorithm's performance in solving the swing-up control task within the MountainCarContinuous-v0 environment, a simplified model of energy-accumulating maneuvers in cyber-physical systems (CPS) such as robotic manipulators, drones, and autonomous vehicles. The task requires an agent to build momentum through oscillatory motion to overcome a gravitational pit and reach a hilltop, despite receiving infrequent and delayed feedback. We evaluated four neural network architectures (64-64, 128-64, 256-128-64, 128-128-128) using Stable-Baselines3 over 200 episodes to examine how policy complexity influences learning efficiency. None of the architectures consistently succeeded, with the 128-64 model displaying the most stable yet limited performance. Final rewards remained near zero, indicating DDPG's difficulty in mastering the swing-up strategy. Learning curves showed instability and stagnation, highlighting DDPG's limitations in CPS tasks with deferred reinforcement. These findings suggest that architectural tweaks alone are insufficient. Future work should investigate reward shaping, demonstration learning, or alternative algorithms like TD3 or SAC to improve outcomes in complex CPS control scenarios.**

Keywords: Reinforcement Learning; Cyber-Physical Systems; Deep Deterministic Policy Gradient; Continuous Control; Mountain Car; Swing-Up Strategy; Neural Network Architecture, Information System.

INTRODUCTION

Cyber-physical systems (CPS) integrate computational algorithms with physical processes through sensing, actuation, and feedback, enabling intelligent behaviors in applications like autonomous navigation, robotic manipulation, and energy-efficient motion planning. As CPS tasks grow in complexity, they demand advanced control strategies that adapt to dynamic and uncertain environments without relying on explicit system models. Reinforcement learning (RL) has emerged as a powerful approach, allowing agents to learn optimal behaviors through trial-and-error interactions, making it highly relevant for CPS applications. [1].

The MountainCarContinuous-v0 environment serves as a benchmark for evaluating RL in energy-based control tasks. In this setting, an underpowered car must swing back and forth to accumulate momentum and reach a hilltop – a puzzle of momentum that mirrors energy accumulation in CPS, such as stabilizing a drone or pumping a mechanical actuator. This oscillatory strategy requires long-term planning, as rewards are only received upon success, posing a significant test for RL algorithms. The task's dynamics, involving inertia and non-linear state transitions, reflect challenges in real-world CPS, such as balancing inverted pendulums or navigating uneven terrain.

Among RL methods, the Deep Deterministic Policy Gradient (DDPG) algorithm is well-suited for continuous action spaces, combining Q-learning's stability with deterministic policy gradients' expressiveness [2]. DDPG has shown promise in control tasks, but struggles with infrequent feedback, delayed rewards, and complex temporal dependencies – common in CPS scenarios [3]. This study evaluates DDPG's ability to learn the swing-up strategy in MountainCarContinuous-v0, focusing on how neural network architecture impacts convergence, stability, and computational efficiency. By testing four multilayer perceptron configurations (64-64, 128-64, 256-128-64, 128-128-128), we aim to uncover trade-offs in policy complexity and provide insights into DDPG's suitability for CPS, contributing to the design of robust RL strategies for autonomous systems.

The MountainCarContinuous-v0 task poses unique obstacles for model-free RL algorithms like DDPG. Unlike typical goal-reaching scenarios with incremental rewards, this task provides feedback only upon reaching the hilltop, creating a challenging credit assignment problem. The agent must infer which actions, taken over hundreds of steps, contributed to a distant reward. The optimal strategy – swinging in the opposite direction to build momentum – is counterintuitive and difficult to discover through random exploration, especially without intermediate guidance. The task's dynamic nature, driven by inertia and non-linear state transitions, further complicates learning. These properties, prevalent in CPS, result in a non-convex policy space that is highly sensitive to exploration noise.

EXPERIMENT DESIGN AND IMPLEMENTATION

Overview

DDPG, an off-policy actor-critic algorithm, is designed for continuous action spaces, making it suitable for CPS tasks requiring precise, real-valued control, such as robotic actuation or motion planning. It integrates Q-learning for value estimation with deterministic policy gradients for policy optimization, offering a balance of stability and expressiveness. However,

DDPG's performance often degrades in environments with infrequent or delayed rewards, as seen in tasks involving energy accumulation. The MountainCarContinuous-v0 environment, where an underpowered car must accumulate momentum through oscillatory motion, mimics these CPS scenarios, making it an ideal testbed for evaluating DDPG's capabilities.

We tested four multilayer perceptron architectures to explore the influence of policy complexity: a compact 64-64 (two hidden layers of 64 units), a moderate 128-64, a deeper and wider 256-128-64, and a symmetric 128-128-128. These configurations were chosen to balance model capacity with the environment's observation and action spaces. Each architecture was implemented in Stable-Baselines3, with agents trained for 200 episodes of 250 timesteps. Exploration was facilitated using NormalActionNoise, and GPU acceleration was employed where available. Performance metrics included final average reward (last 10 episodes), reward variance, and training time.

Hyperparameter considerations

DDPG's performance is highly sensitive to hyperparameters, including actor and critic learning rates, soft update coefficients, discount factors, and action noise variance. To ensure a fair comparison, we kept these parameters constant across experiments. However, their settings significantly influence learning dynamics. For instance, Gaussian action noise affects exploration; insufficient amplitude may prevent the agent from discovering the swing-up strategy in sparse-reward settings like MountainCarContinuous-v0. Similarly, a small soft update coefficient stabilizes training but may slow policy updates, impacting learning speed. Although we did not systematically optimize hyperparameters, the observed performance stagnation suggests that refined tuning could unlock DDPG's potential. Future work will include sensitivity analysis to identify optimal settings for CPS-relevant tasks.

The methodology sets the stage for a comprehensive evaluation of DDPG's learning behavior, with results expected to reveal how architecture influences efficiency, stability, and robustness in sparse-reward environments.

Results and Discussion

None of the tested architectures mastered the swing-up task within 200 episodes, with final average rewards remaining near zero across all models. This indicates that DDPG struggled to learn effective momentum-building strategies in the MountainCarContinuous-v0 environment. The 128-64 architecture performed most consistently, with the lowest reward variance, but its gains were modest. In contrast, the 256-128-64 model, with the largest parameter space, exhibited significant fluctuations, including a sharp performance drop mid-training, suggesting erratic learning behavior.

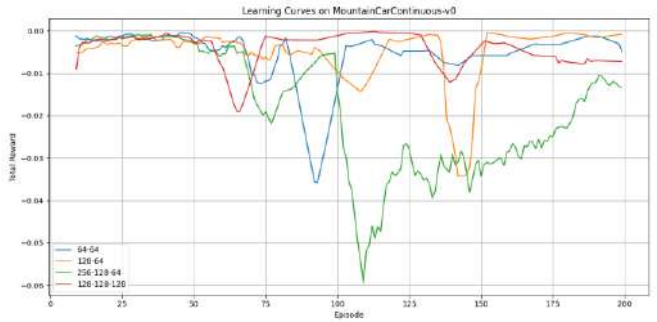

Fig. 1. Episode-wise learning curves of DDPG agents

Figure 1, which plots episode-wise rewards reveals early improvements followed by stagnation or decline across all models. For example, the 256-128-64 model's rewards plummeted around episode 100, while other architectures plateaued, reflecting difficulty sustaining progress. The 128-64 model showed the flattest curve, indicating stability but limited learning. Training times varied slightly, suggesting that computational cost was not the primary barrier.

The 128-64 model's stability suggests that moderate architectures may balance exploration and learning better than deeper ones. However, the lack of progress across all configurations underscores DDPG's limitations in handling infrequent reinforcement and momentum-based control. The 256-128-64 model's fluctuations raise concerns about overfitting or inefficient exploration in high-dimensional policy spaces. These findings align with prior research highlighting DDPG's sensitivity to reward structure and its dependence on dense feedback for effective policy updates.

Reward shaping could provide smoother feedback, guiding the agent toward the swing-up strategy. Pretraining with expert demonstrations might bootstrap initial policies, reducing reliance on random exploration. Algorithms like TD3 or SAC, designed for improved stability and exploration, may also outperform DDPG in similar tasks. These insights are particularly relevant for CPS applications, where energy-based behaviors and long-term planning are common, informing the development of robust RL solutions.

Conclusion and Future Work

The results confirmed known limitations of DDPG in sparse-reward environments, where the agent must learn to associate sequences of actions with delayed positive outcomes. The limited progress across all tested architectures, including deeper models, suggests that neither increased network capacity nor more complex representations sufficiently improve learning in such settings. Moreover, the instability and performance collapse observed in certain configurations (e.g., 256-128-64) highlight the risks of overfitting and poor exploration in high-dimensional policy spaces.

These findings underscore the importance of algorithmic adaptations when applying deep reinforcement learning to CPS tasks involving inertia, momentum, and energy-based behaviors. The standard DDPG framework may benefit from enhancements such as reward shaping to provide smoother feedback, curriculum learning to incrementally introduce task complexity, and expert-guided learning to bootstrap the initial policy.

In future work, we aim to use alternative algorithms like TD3 and SAC will help identify more robust solutions for CPS control, paving the way for smarter autonomous systems. We also plan to introduce visual diagnostics, such as trajectory visualization in the state space, to better understand behavioral patterns during learning. Benchmarking

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1, no. 1. Cambridge, MA: MIT Press, 1998.

[2] M. Sewak, "Deterministic policy gradient and the DDPG: Deterministic-policy-gradient-based approaches," in *Deep Reinforcement Learning: Frontiers of Artificial Intelligence*, Singapore: Springer Singapore, 2019, pp. 173–184.

[3] M. Drahan and A. Pysarenko, "Reinforcement learning algorithms for intelligent control in cyber-physical systems," in Proc. XXVII Int. Conf. Automatics 2024, Dnipro, Ukraine, Nov. 20–22, 2024, pp. 13–14, ISBN 978-966-2344-74-5.

Automatic Detection of Price Tags in Retail Product Images

Anatolii Ivanov, Viktoriia Onyshchenko
Department of Information Systems and Technologies
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
anatolii.0700@gmail.com, v.onyshchenko@kpi.ua

Abstract. **This paper investigates the problem of price tags automatic detection on retail product images under various lighting and placement conditions. An adapted YOLOv5 architecture is proposed to improve the accuracy of their localization on retail shelves. The model was trained on a dataset containing real-world images with diverse environmental conditions. A comparative analysis showed an accuracy improvement of 4,1% in the mAP@[.5:.95] metric compared to baseline models on the selected dataset.**

Keywords*: *Computer vision, image processing, pattern recognition, YOLOv5.

Автоматичне виявлення цінників на зображеннях товарів

Іванов Анатолій Ігорович, Онищенко Вікторія Валеріївна
КПІ імені Ігоря Сікорського
Київ, Україна
anatolii.0700@gmail.com, v.onyshchenko@kpi.ua

Анотація. **У данній роботі досліджується проблема автоматичного виявлення цінників на зображеннях роздрібних товарів у різних умовах освітлення та розташування. Пропонується адаптована архітектура моделі YOLOv5 для підвищення їх точності локалізації на торгових вітринах. Модель навчено на датасеті, що містить реальні зображення з різними умовами середовища. Порівняльний аналіз показав приріст точності за метрикою mAP@[.5:.95] на 4,1% відносно базових моделей на обраному наборі даних.**

Ключові слова: комп'ютерний зір, YOLOv5, обробка зображень, ритейл

Вступ

Автоматичне виявлення цінників на зображеннях товарів є актуальною задачею в роздрібній торгівлі та інвентаризації, що дозволяє суттєво спростити оновлення інформації про товари. З іншого боку, вирішення даної задачі може дозволити покупцям використовувати інформаційні технології для швидкого пошуку необхідних товарів. Проте цінники – це типово дрібні об'єкти, які мають обмежену представленість у пікселях зображення, що ускладнює виділення їх суттєвих ознак [1]. Для вирішення задач такого роду зазвичай використовуються передусім глибокі згорткові мережі.

Одноетапні детектори, зокрема серія моделей YOLO (You Only Look Once), відомі дуже високою швидкістю роботи при збереженні високої точності [2]. У цій роботі обрано модель YOLOv5, яка демонструє добре збалансовані показники точності та швидкодії та має широку підтримку тонкого налаштування (custom training).

Модель було донавчено на наборі даних PriceTag (Roboflow), що містить 326 реальних зображень товарів з анотованими рамками цінників. Метою даної роботи є підвищення точності локалізації цінників за рахунок спеціалізованих методик аугментації і архітектурних поліпшень.

Огляд літератури

Існують декілька класичних моделей для знаходження певних патернів на зображенні. Перший з них — одноетапні методи: YOLO-детектори здійснюють обробку «в один прохід» та показують високу швидкість детекції зображень з відносно високою точністю [2]. Ці моделі підходять для задач, де критично важливий час обробки, наприклад, у системах розпізнавання в режимі реального часу. SSD (Single Shot Detector) також належить до одноетапних методів, що дозволяють швидко виявляти об'єкти одним проходом зображення, жертвуючи частиною точності для підвищення швидкодії [3]. SSD, зокрема, широко застосовується у мобільних рішеннях та веб-застосунках, де обчислювальні ресурси обмежені.

Інший варіант – це двоетапні методи (Faster R-CNN): вони спочатку генерують області зацікавлення (region proposals), а лише потім класифікують їх. Це забезпечує високу точність, особливо для дуже дрібних об'єктів, але при цьому, це значно уповільнює роботу системи [2]. Такі моделі зазвичай використовуються в офлайн-аналітиці або в системах, де точність критично важлива та допустима висока затримка.

Також існують більш спеціалізовані рішення для цінників і схожих об’єктів. Наприклад, у задачу сегментації зображень цінників був залучений YOLOv4, що забезпечив точність біля 96,9%. Це підтверджує ефективність архітектур на базі YOLO для подібних завдань.

Застосовуються також методи на основі U-Net, MobileNet, VGG і т.д., проте їх докладний огляд виходить за рамки цієї роботи. В цілому, вибір між одно-та двоетапними методами зумовлюється балансом між точністю і швидкодією, необхідним для конкретного застосування. Оскільки основою даної задачі є можливість роботи на слабких пристроях в майже реальному часі, має сенс використовувати одноетапні моделі, зменшуючи час роботи, але при цьому трохи погіршуючи результати.

Методологія

В даній роботі як базу було використано модель YOLOv5, яка довела свою ефективність у задачах детекції завдяки поєднанню високої швидкодії та точності [4]. Модель донавчена на наборі PriceTag, що містить 134 зображення у форматі YOLO (вхідні зображення та текстові файли з координатами рамок цінників).

Першою зміною була аугментація зображень, тобто застосовано низку таких технік як масштабування, випадкові зміни яскравості, контрасту, додавання шуму для більш різноманітної моделі навчання з підібраними гіперпараметрами. Це дозволило збільшити різноманітність даних і допомогло моделі бути стійкішою до варіацій положення і вигляду цінників.

Другою зміною було використання якорів, оскільки YOLOv5 надає можливість використання набору попередньо визначених «якірних» (anchor) рамок різних розмірів та форм [5]. Вони накладаються на зображення на різних масштабах (сітках) і служать початковими припущеннями для позицій об’єктів. У застосованій конфігурації параметри анкоров підібрано під розміри цінників на зображеннях, що сприяє поліпшенню локалізації дрібних об’єктів.

Третьою зміною було застосування модулю уваги (CBAM) [6]. Convolutional Block Attention Module (CBAM), поєднує канал та просторову увагу. Він дозволяє моделі краще фокусуватися на релевантних ознаках зображення і підсилює корисні для детекції фрагменти, що покращує точність локалізації дрібних цінників.

Останньою зміною було поєднання методик, тобто комплексне застосування аугментації та налаштованих анкорових рамок, що надало синергетичний ефект, підвищуючи чутливість моделі до наданих даних. Важливо зазначити, що для оцінки не використовувалися інші набори даних – всі результати отримані на основі датасету PriceTag, без залучення додаткових джерел.

Результати

Для оцінки ефективності підходів було проведено серію експериментів на тестовій підмножині набору PriceTag, яка містить 134 зображень з 327 об'єктами. Основними метриками слугували Precision (P), Recall (R), mAP@0.5 та mAP@0.5:0.95. Було протестовано базову конфігурацію YOLOv5s, а також модифіковані варіанти з аугментацією, еволюційним підбором anchors (*Evolve*), CBAM-модулем і їхньою комбінацією.

Базова модель показала хороші результати (mAP@0.5 = 0.829), однак подальші модифікації дозволили покращити точність. Додавання лише аугментації призвело до зростання precision до 0,914, але recall суттєво зменшився. Це свідчить про покращення визначення окремих, добре помітних об'єктів, але із втратою деяких менш виразних.

Таблиця 1

Результат тестування моделей

Конфігурація	P	R	mAP@0.5	mAP@0.5:0.95
Default (базова)	0,850	0,782	0,829	0,472
Augmentation	0,914	0,631	0,799	0,481
Evolve (anchors)	0,817	0,492	0,644	0,314
CBAM	0,870	0,800	0,839	0,482
Evolve + Augmentation	0,936	0,754	0,870	0,521

Використання лише еволюційно підібраних anchors (Evolve) призвело до найнижчих результатів серед усіх конфігурацій (mAP@0.5:0.95 = 0,314), що свідчить про недостатність цієї зміни без додаткових покращень. Водночас CBAM-модуль, інтегрований у архітектуру, забезпечив стабільне підвищення метрик, особливо recall.

Найкращих результатів вдалося досягти при комбінуванні двох підходів — Evolve та аугментації. Так, mAP@0.5 зросла до 0,870, а mAP@0.5:0.95 — до 0,521, що є найвищими значеннями серед усіх експериментів. Таким чином вдалося отримати покращення на 4,1% у порівнняні з базовою моделлю під час використання даної модифікації.

Висновки

У даній роботі було досліджено задачу автоматичного виявлення цінників на зображеннях товарів у роздрібному середовищі. Для її вирішення було адаптовано модель YOLOv5 з урахуванням особливостей цільового класу об'єктів. Впровадження модулю CBAM, а також цілеспрямоване налаштування якорів та аугментації дозволили суттєво покращити результати. Найвищі значення точності (mAP@0.5 = 0,870, mAP@0.5:0.95 = 0,521) були досягнуті при комбінуванні кількох підходів, що підтверджує доцільність їх інтеграції. Отримані результати демонструють перспективність удосконаленої архітектури для локалізації об’єктів у складному візуальному середовищі. При цьому варто зазначити, що кращі результати було б досягнуто за умови використання датасету більшого об’єму та різноманітності зображень.

Література

1. Zhyuan W. Improved Small Object Detection Algorithm CRL-YOLOv5 / W. Zhyuan, M. Shujun, B. Yuntian et al. // Sensors. – 2024. – Vol. 24, № 19. – P. 6437.

2. Laptev P. Neural Network-Based Price Tag Data Analysis / P. Laptev, S. Litovkin et al. // Future Internet. – 2022. – Vol. 14, № 3. – P. 88.

3. Shobaki W.A. Comparative Study of YOLO, SSD, Faster R-CNN, for Optimized Eye-Gaze Writing / W.A. Shobaki, M. Milanova // Smart Cities. – 2024. – Vol. 7, № 2. – P. 47.

4. Diwan T. Object detection using YOLO: challenges, architectural successors, datasets and applications / T. Diwan, G. Anirudh, J.V. Tembhurne // Multimedia Tools and Applications. – 2023. – Vol. 82. – P. 9243–9275.

5. Zhong Y. Anchor Box Optimization for Object Detection / Y. Zhong, J. Wang, J. Peng, L. Zhang // Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). – 2020. – P. 1286–1294.

6. He L. CBAM-YOLOv5: A Promising Network Model for Wear Particle Recognition / L. He, H. Wei // Journal of Advanced Mechanical Design, Systems, and Manufacturing. – 2023. – Article №2520.

ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ

PROGRAMMING TECHNOLOGIES

Generating graphic visualizations using programming language tools with direct access to video memory

Oleksandr Zhyrytovskyi, Roman Zubko
Open International University of Human Development "Ukraine"
Kyiv, Ukraine
i.am.zhirik@gmail.com, rzubko@ukr.net

Abstract. Previously, programming languages such as BASIC and PASCAL allowed direct access to video memory. This made it possible to experiment with various visualizations on the computer screen. Today, processors are much faster, and video adapters can display images at higher resolutions. However, due to the lack of direct access to video memory in modern programming languages, algorithms for working with graphics have not seen significant development. The purpose of this article is to demonstrate the types of visualizations that can be created with direct access to video memory.

***Keywords*: computer graphics, visualizations, fractals, 3d rendering, texture mapping, mesh, triangualtion.**

Формування графічних візуалізацій за допомогою засобів мови програмування з прямим доступом до відеопам'яті

Жиритовський Олександр, Зубко Роман
Відкритий міжнародний університет розвитку людини «Україна»
Київ, Україна
i.am.zhirik@gmail.com, rzubko@ukr.net

Анотація. Раніше мови програмування такі, як BASIC чи PASCAL, мали можливість прямого доступу до відеопам'яті. Як наслідок, можна було експериментувати з відображенням різноманітних візуалізацій на екрані комп'ютера. На сьогодні процесори стали набагато швидшими і відеоадаптери здатні відтворювати зображення з більшою роздільною здатністю. Проте, за відсутності засобів прямого доступу до відеопам'яті у сучасних мовах програмування, алгоритми для роботи з графікою не зазнали значного розвитку. Метою даної статті є показ візуалізацій, які можуть бути розроблені за наявності прямого доступу до відеопам'яті.

Ключові слова: комп'ютерна графіка, візуалізації, фрактали, 3d рендеринг, накладання текстури, каркасна сітка, триангуляція.

Вступ

Формування графічної візуалізації передбачає задання певного кольору кожному пікселю на екрані. Маючи мову програмування з можливістю прямого доступу до відеопам'яті, можна створювати візуалізації на екрані комп'ютера у реальному часі.

Фрактали

Фрактал — це геометрична фігура або структура, яка має властивість самоподібності, тобто повторюється в зменшеному масштабі на різних рівнях. Це означає, що якщо збільшити якусь частину фракталу, вона виглядатиме подібно до цілого. Розглянемо один із фракталів, який називається множина Мандельброта [1]. В основі хаотичності даного фракталу лежить послідовність

$$Z_i = Z_{i-1}^2 + C \qquad (1)$$

При умові, що початкове значення Z_0 та константа C лежать в межах від -2 до 2, ця послідовність починає поводити себе дуже хаотично. А саме, вона спочатку коливається на цьому проміжку певний час, а уже потім наближається до 0 чи прямує до нескінченності. Кількість ітерацій, за яку відбувається вихід за межі діапазону від -2 до 2, можна відобразити у вигляді градацій яскравості кольору. Для створення двовимірної картинки, замість цілих чисел, потрібно використати комплексні числа. Приклад множини Мандельброта наведено на рис. 1.

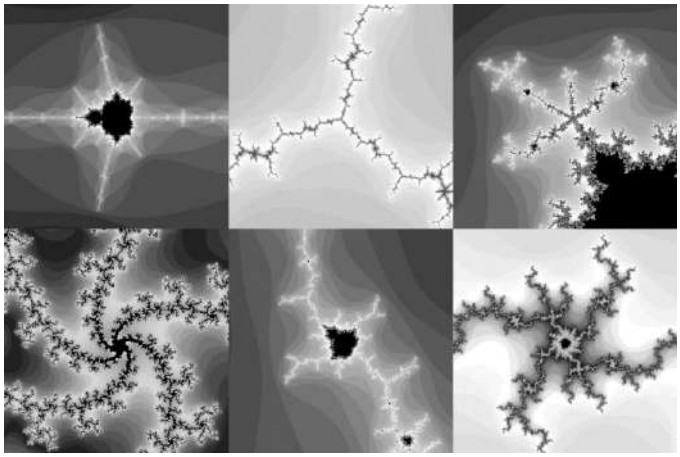

Рис. 1. Множина Мандельброта

Множина Жюліа будується так само, як і множина Мандельброта. Відмінністю є тільки те, що константа *C* у ній задається не поточною координатою точки, а певним сталим значенням. Приклад множини Жюліа [2] наведено на рис. 2.

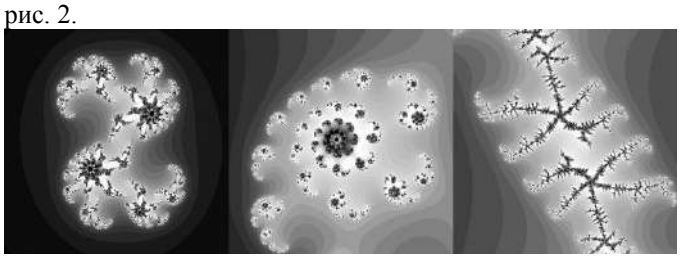

Рис. 2. Множина Жюліа

Фрактальні дерева

В основу побудови фрактальних дерев покладена функція побудови гілки дерева. Ця функція приймає координати *x* та *y* початку гілки, кут нахилу гілки та рівень гілки. Стовбур має перший рівень, довгі гілки мають другий рівень, коротші гілки – третій рівень і так далі. Знаючи координати та кут відносно цих координат, можна намалювати нову гілку уже вищого рівня під потрібним кутом відносно поточного. Кожна з гілок дерева будується подібним чином. Приклад етапів побудови фрактального дерева зображено на рис. 3. Кожен наступний етап тут доповнюється наступним рівнем гілок.

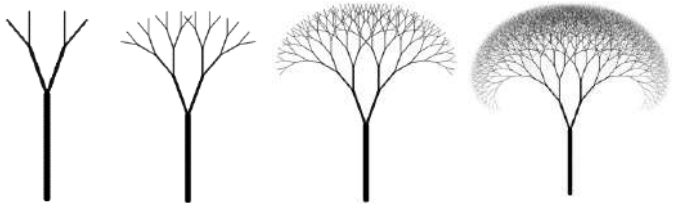

Рис. 3. Етапи побудови фрактального дерева

Як результат, можна побудувати фрактальне дерево з різними довжинами гілок, що нахилені під різними кутами та використовуючи дві або три криві лінії. Криві лінії тут описуються кубічними сплайнами. Кожна наступна гілка є сплайном, що будується як продовження попереднього сплайну. Приклади таких дерев зображені на рис. 4.

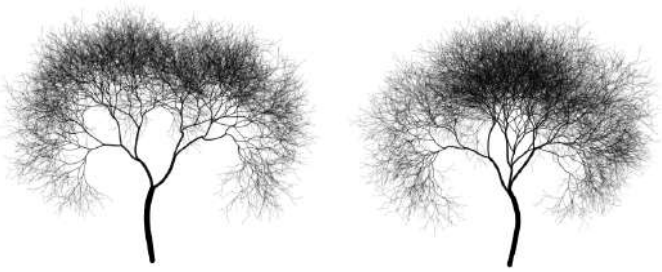

Рис. 4. Фрактальні дерева

Для створення більшої реалістичності доцільно намалювати дерево з більш товстим стовбуром. Для цього можна використати сплайни, які мають різну товщину ліній на їх кінцях. Приклади зображень таких дерев з товстим стовбуром побудованих на основі даного підходу наведено на рис. 5.

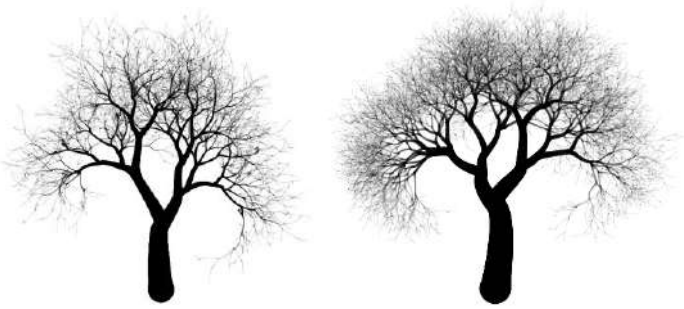

Рис. 5. Фрактальні дерева побудовані на основі кривих змінної товщини

Ще одним типом фрактальних дерев є Н-дерево. Воно має таку назву, оскільки схоже на літеру «Н». Його дві гілки можна уявити у вигляді двох годинникових стрілок. На кінцях кожної з цих двох годинникових стрілок знаходяться ще дві годинникові стрілки, повернуті на такі самі кути і так далі. В залежності від того під якими кутами знаходяться гілки будуть отримані різноманітні фігури. Приклади фігур, що базуються на обертанні гілок Н-дерева зображено на рис. 6.

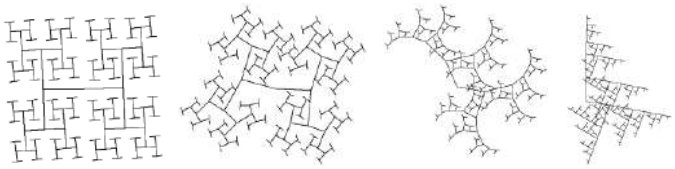

Рис. 6. Фігури побудовані на основі обертання гілок Н-дерева

Хаотичні функції

Особливістю хаотичних функцій є те, що кожне наступне значення функції рахується на основі попереднього. В основі хаотичного розподілу значень таких функцій лежить формула кола, але змінна *y* тут рахується за допомогою її попереднього значення:

$$x_i = \cos(\alpha)$$
$$y_i = \sin(\alpha + y_{i-1} \cdot t) \quad (2)$$

де:

α – кут, що змінюється від 0 до 2π;

t – певна константа, яка за необхідності може змінюватися у часі;

x, y – координати кожної точки графіка функції, що відображається на екрані.

Функцію за такою формулою зображено на рис.7.

Рис. 7. Хаотична функція за формулою (2)

На рис. 8 наведено приклад хаотичної функції, що будується за формулою:

$$x = \cos\left(\alpha + 2 \cdot \sqrt{0.1 + (x+t)^2 + (y + t/(x+y+t))^2}\right)$$
$$y = \sin\left(\alpha + 2 \cdot \sqrt{0.1 + (x+t)^2 + (y+t)^2}\right) \quad (3)$$

де:

α – кут, що змінюється від 0 до 2π;

t – певна константа, яка за необхідності може змінюватися у часі;

x, y – координати кожної точки графіка функції, що відображається на екрані.

Рис. 8. Хаотична функція за формулою (3)

Фігури Ліссажу

В основі побудови фігур Ліссажу [3] лежить формула кола, у яку додані два коефіцієнти. Вона має наступний вигляд:

$$x = \cos(k_1 \alpha)$$
$$y = \sin(k_2 \alpha) \quad (4)$$

де:

α – кут, що змінюється від 0 до 2π;

k_1, k_2 – коефіцієнти, що задають зображення функції;

x, y – координати кожної точки графіка функції, що відображається на екрані.

Комбінації двох коефіцієнтів даватимуть різні фігури. Приклади таких фігур зображено на рис. 9.

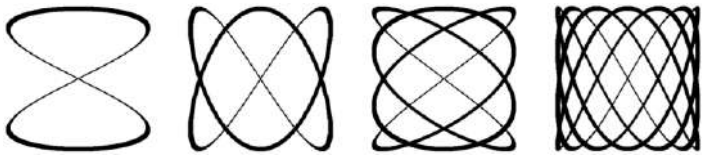

Рис. 9. Фігури Ліссажу

Троянди Гвідо Гранді

Троянди Гвідо Гранді, так само як і фігури Ліссажу, будуються за двома коефіцієнтами, але їх формула інша і має наступний вигляд:

$$x = \cos(k_1 \alpha) \cdot \cos(k_2 \alpha)$$
$$y = \sin(k_2 \alpha) \cdot \cos(k_1 \alpha) \quad (5)$$

де:

α – кут, що змінюється від 0 до 2π;

k_1, k_2 – коефіцієнти, що задають зображення функції;

x, y – координати кожної точки графіка функції, що відображається на екрані.

Приклади троянд Гвідо Гранді зображено на рис. 10.

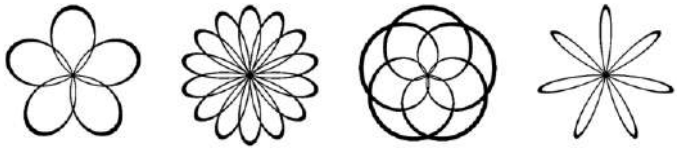

Рис. 10. Троянди Гвідо Гранді

Генерація лабіринтів

Для генерації лабіринту спочатку будується рамка, що є чотирма стінами по краям. В середині цієї рамки кожна нова стінка малюється починаючи від довільної існуючої стінки таким чином, щоб її кінець ні до якої іншої стінки не примикав. Результат побудованого таким способом лабіринту зображено на рис. 11.

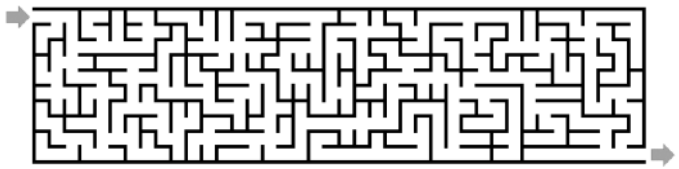

Рис. 11. Лабіринт

Візуалізація руху хвиль на воді

В даній моделі кожна точка екрану має колір, що характеризує висоту хвилі в цій точці. Оскільки хвилі накладаються, то висота хвилі в точці є сумою висот хвиль які її перетинають (рис. 12).

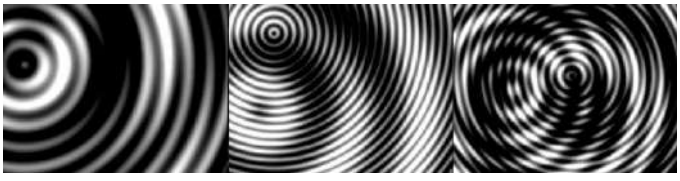

Рис. 12. Візуалізація руху хвиль на воді

Побудова тривимірних фігур на основі точок

В основі побудови фігур у тривимірному просторі лежить формула тривимірного повороту, яка має вигляд:

$$x' = x_c + x\cos\alpha - y\sin\alpha$$
$$y' = y_c + (x\sin\alpha + y\cos\alpha)\sin\beta + z\cos\beta \quad (6)$$

де:

x, y, z – координати точки у тривимірному просторі;

α, β – кути повороту тривимірної площини;

x_c, y_c – координати центру екрану;

x', y' – двовимірні координати точки на екрані.

Використовуючи формулу (6) можна ставити точку на екрані у довільних координатах. Як приклад на рис. 13 намальовано тривимірні зображення квіток жоржини та троянди.

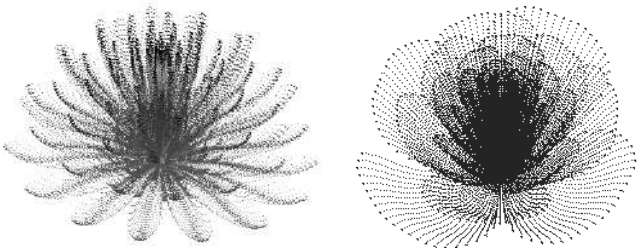

Рис. 13. Зображення квіток жоржини та троянди побудованих за допомогою малювання точок у тривимірному просторі

Побудова тривимірної фігури за допомогою каркасної сітки

В основі побудови тривимірної фігури є сітка з точок. Для побудови каркасу фігури сусідні точки у ній з'єднуються прямими лініями. Для побудови суцільної фігури кожна клітинка сітки заміщується двома трикутниками. Результати таких перетворень зображено на рис. 14.

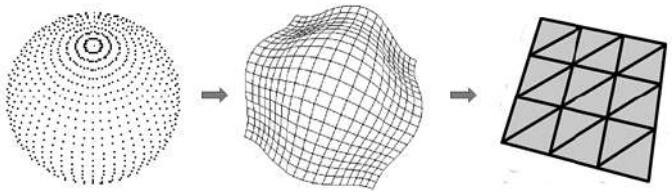

Рис. 14. Побудова тривимірної фігури за допомогою каркасної сітки

Накладання текстури на грань

Кожна трикутна грань фігури, у найпростішому випадку, може мати суцільний колір. Більш складним варіантом є плавне переливання трьох кольорів між вершинами трикутника. Накладання текстури на трикутну грань виконується повністю ідентично за тими самим формулами, як і переливання трьох кольорів. Тільки замість плавної зміни кольорів відбувається плавна зміна координат точок зображення текстури (рис. 15).

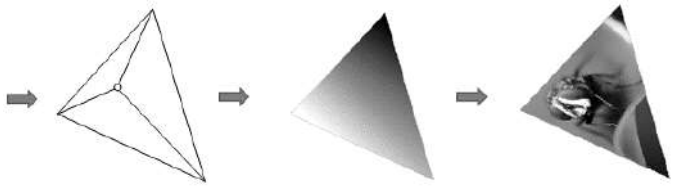

Рис. 15. Накладання текстури на грань

Накладання текстури на довільну форму

Якщо побудувати сітку з трикутників, що має певну форму, то для кожного трикутника можна окремо вказати координати текстури. Як наслідок, зображення текстури можна накласти на довільну форму [4] (рис. 16).

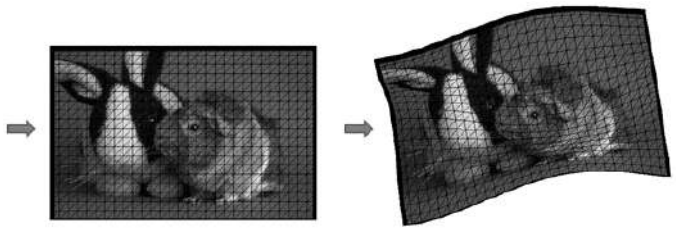

Рис. 16. Накладання текстури на довільну форму

Системи часток

Система часток [5] передбачає наявність великої кількості часток, які поводять себе за певними законами. Частка представляє собою певну нечітку фігуру, що має визначений колір, розмір, поточні координати, напрямок руху і швидкість. Також частка має фіксовану тривалість життя, після чого може просто зникнути, або перетворитися на набір інших часток. Для симуляції вибухів феєрверків будемо вважати, що є певна частка (ракета), яка у процесі польоту створює за собою слід із часток. У певний момент ракета зникає перетворюючись на набір часток у вигляді іскор, які розлітаються у різні сторони. Приклад симуляції вибухів феєрверків, використовуючи систему часток, зображено на рис. 17.

Рис.17. Симуляція вибухів феєрверків

Імітація вогню

В основі імітації вогню є безперервний цикл малювання помаранчевих крапок внизу екрана та розмиття. Розмиття відбувається присвоєнням точці кінцевого зображення середнього значення сусідніх точок початкового зображення. Що правда у такому випадку розмір полум'я буде лише у декілька пікселів. Для того щоб підняти полум'я угору, можна задати певні коефіцієнти для розмиття. Тоді замість формули середнього арифметичного (7)

$$\frac{a + b + c}{3} \quad (7)$$

використовується формула середнього зваженого (8).

$$\frac{1 \cdot a + 2 \cdot b + 3 \cdot c}{1 + 2 + 3} \quad (8)$$

Приклад коефіцієнтів та зображення полум'я наведено на рис. 18.

$$\begin{pmatrix} 10 & 1 & 10 \\ 1 & 1 & 1 \\ 1 & 100 & 1 \end{pmatrix}$$

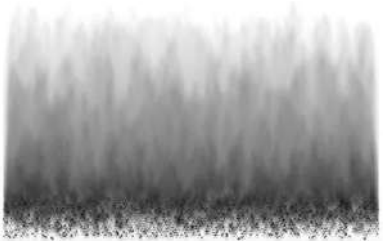

Рис. 18. Коефіцієнти розмиття та зображення імітованого вогню

Псевдо-тривимірні деформаційні візуалізації

Ідея створення псевдо-тривимірних деформаційних візуалізацій [6] полягає у тому, що кожній точці кінцевого зображення ставиться у відповідність певна точка початкового зображення. Потім кінцеве зображення стає початковим і цей цикл повторюється. Координати точки кінцевого зображення можуть бути дійсними числами. Тому можна використати техніку білінійного згладжування. А саме, для цього у кожної з координат треба відокремити цілу та дійсну частини чисел. Після чого потрібно взяти фрагмент початкового зображення розміром 2x2 пікселі з цілими координатами. Потім змішати ці чотири кольори пікселів в один у пропорції, що задаватиметься дійсними частинами чисел поточних координат. Циклічне малювання та зміщення точок у нові координати, разом з білінійним згладжуванням, може сформувати візуалізацію галактики (рис. 19).

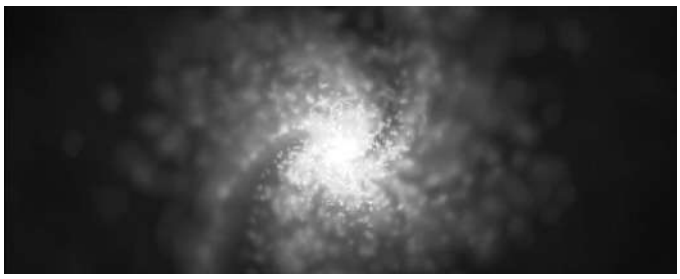

Рис. 19. Візуалізація галактики

Графічне відображення сніжинок

Сніжинка генерується як шість променів, рівномірно розташованих під кутом 60°. Кожен промінь має чотири відгалуження з випадковою довжиною, що розміщені під таким самим кутом. Саме ці відгалуження визначають унікальну форму сніжинки. Приклад сніжинок, побудованих за таким алгоритмом, наведено на рис. 20.

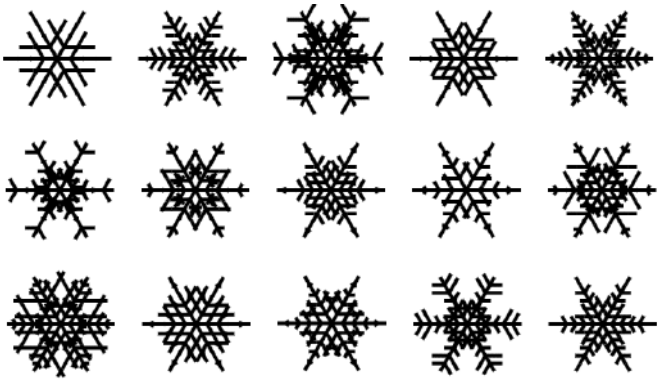

Рис. 20. Графічне відображення сніжинок

Висновки

Візуалізації у даній статті були згенеровані за допомогою прямого доступу до відеопам'яті. Можна зробити висновок, що написання алгоритмів, які виконуються на процесорі комп'ютера з прямим доступом до відеопам'яті є потужним механізмом для формування різноманітних візуалізацій на екрані. Для подальшого розвитку алгоритмів побудови візуалізацій актуальним є наявність у мовах програмування вбудованих засобів для прямого доступу до відеопам'яті.

Література

1. Mandelbrot B.B. The Fractal Geometry of Nature / B.B. Mandelbrot. – San Francisco: W.H. Freeman and Company, 1982. – 468 p.

2. Barnsley, M. F. Fractals Everywhere. – London : Academic Press Inc., 1988. – 396 p. – P. 269.

3. Greenslade, T. B. Jr. Adventures with Lissajous Figures / T. B. Jr. Greenslade. – San Rafael, CA : Morgan & Claypool Publishers, 2018. – 89 p.

4. Botsch, M., Kobbelt, L., Pauly, M., Alliez, P., & Lévy, B. Polygon Mesh Processing. – Natick, Massachusetts: AK Peters, 2010. – 364 p.

5. https://shawnhargreaves.com/egg/ (дата звернення 05.05.2025).

6. https://www.geisswerks.com/geiss/ (дата звернення 05.05.2025)