НАЦІОНАЛЬНА
АКАДЕМІЯ НАУК УКРАЇНИ
ІНСТИТУТ ПРОГРАМНИХ СИСТЕМ

ПРОБЛЕМИ ПРОГРАМУВАННЯ

науковий журнал

№ 3 липень – вересень 2025

Заснований у березні 1999 р.

ЗМІСТ

Свідоцтво про державну реєстрацію КВ № 7490 від 01.07.2003

Науковий журнал "Проблеми програмування" занесений до переліку наукових фахових видань України, в яких можуть публікуватися основні результати дисертаційних робіт.

ISSN 1727-4907

NATIONAL ACADEMY
OF SCIENCES OF UKRAINE
INSTITUTE OF SOFTWARE SYSTEMS

PROBLEMS IN PROGRAMMING

scientific journal

№ 3 ***July – September*** ***2025***

Founded in March, 1999

CONTENT

УДК 004.41 http://doi.org/10.15407/pp2025.03.003

Г.Б. Мороз

МЕТАЕВРИСТИЧНІ МЕТОДИ ОПТИМІЗАЦІЇ ЯКОСТІ ОБСЛУГОВУВАННЯ КОМПОЗИТНИХ ВЕБСЕРВІСІВ

З появою сервіс-орієнтованих архітектур стало можливим реєструвати, викликати та об'єднувати вебсервіси за їхніми ідентичними атрибутами якості обслуговування для створення композитних вебсервісів з доданою вартістю, які відповідають потребам користувачів. Проте швидке впровадження нових вебсервісів у динамічне бізнес-середовище може негативно вплинути на їхню якість обслуговування. Тому питання про те, як залучити, агрегувати та використати інформацію про якість обслуговування окремих *вебсервісів* для отримання оптимальної наскрізної якості обслуговування композитного вебсервіса, є наразі одним із приорітетних напрямків дослідження в програмній інженерії та сервіс-орієнтованих обчисленнях. У цій роботі представлені базова теоретична інформація, необхідна для розуміння важливості, багатогранності та складності проблеми композиції вебсервісів з урахуванням їхньої якості обслуговування, а також репрезентативний огляд використання метаевристичних методів глобальної оптимізації, які протягом останніх двох десятиліть є домінуючими методами вирішення даної проблеми. Мета роботи - привернути увагу студентів та наукової спільноти до актуальних проблем композиції вебсервісів, які виникають в Інтернеті речей, хмарних обчисленнях, соціальних мережах, технологіях мобільних комп'ютерів, смартфонів тощо, і залучити їх до активної участі в розв'язанні цих проблем.
Ключові слова: Сервісно-орієнтоване обчислення, якість обслуговування, композиція вебсервісів, композитний вебсервіс, метаевристика.

H.B. Moroz

METAHEURISTIC METHODS FOR OPTIMIZING THE QUALITY OF SERVICE OF COMPOSITE WEB SERVICES

With the advent of service-oriented architectures, it has become possible to register, invoke, and aggregate web services based on their identical quality of service attributes to create composite web services with added value that meet user needs. However, the rapid introduction of new web services into a dynamic business environment can negatively affect their quality of service. Therefore, the question of how to capture, aggregate, and use information about the quality of service of individual web services to obtain an optimal end-to-end quality of service of a composite web service is currently one of the priority research areas in software engineering and service-oriented computing. This paper presents the basic theoretical information necessary to understand the importance, multifacetedness, and complexity of the problem of web service composition taking into account their quality of service, as well as a representative overview of the use of methods global optimization metaheuristic, which have been the dominant methods for solving this problem over the past two decades. The purpose of the work is to draw the attention of students and the scientific community to the current problems of web service composition that arise in the Internet of Things, cloud computing, social networks, mobile computer and smartphone technologies, etc., and to involve them in active participation in solving these problems.
Key words: Service-oriented computing, quality of service, web service composition, composite web service, metaheuristics.

Вступ

Із кінця XX століття сервіс-орієнтоване обчислення (**SOC**) є важливою обчислювальною парадигмою, яка змінила спосіб розробки та використання програмних додатків [1], а також стала підгрунтям для еволюції компонентно-орієнтованої інженерії програмного забезпечення до більш прогресивної сервіс-орієнтованої програмної інженерії. У цій парадигмі *вебсервіси* (або *сервіси*) розглядаються як фундаментальні будівельні блоки для підтримки швидкої, гнучкої та економічно ефективної розробки розподілених програм у гетерогенних середовищах [2]. Із появою сервіс-орієнтованих архітектур (**SOA**) стало можливо сервіси реєструвати, виявляти, викликати в розподілених середовищах, публікувати та вільно об'єднувати між собою в один потуж-

ніший, так званий, композитний вебсервіс (**CWS**, composite web service**).** Об'єднання (або композиція) сервісів є однією з ключових проблем для технології SOA та сервісів. Воно дозволяє організаціям створювати альянси, передавати функціональні можливості аутсорсингу та надавати своїм клієнтам комплексне обслуговування. З точки зору бізнесу композиція вебсервісів (**WSC**, *web services composition*) різко знижує вартість і ризики створення нових бізнес-додатків у тому сенсі, що існуючі бізнес-логіки представлені як сервіси та можуть бути повторно використані [3].

З роками популярність і використання сервісів експоненційно зростає. Швидке впровадження соціальних мереж, хмарних обчислень, мереж речей тощо тільки сприяє стрімкому збільшенняю доступних в Інтернеті сервісів. Щоденно клієнти по всьому світу генерують міл'ярди запитів на різноманітні сервіси, а тисячі провайдерів (постачальників, розробників) пропонують нові або модифіковані існуючі сервіси. Серед величезної кількості наявних нині сервісів від різних провайдерів сотні, а то і тисячі можуть мати однакову або дуже схожу функціональність. За такого розвитку подій першорядного значення, як для провайдерів, так і для клієнтів, набуває якість обслуговування сервісів (**QoS**, *quality of service*) або, у більш вузькому сенсі, нефункціональні властивості сервісів. З одного боку, диференціація сервісів відносно QoS надає можливість провайдерам відрізняти свої сервіси від інших функціонально подібних сервісів, а, отже, QoS стає ключовим інструментом конкуренції провайдерів. З іншого боку, клієнти, оцінюючи QoS сервісів з однаковою функціональностю, отримують можливість визначити, який сервіс, а отже і провайдер, їх найбільше влаштовує. Проте в реальному житті це зробити незавжди просто, оскільки QoS, що надається клієнту, залежить від багатьох як внутрішніх, так і зовнішних факторів. Зокрема, таких як обчислювальні ресурси провайдера, продуктивність самого сервісу, хостингової платформи, географічне положення користувача, швидкість підключення до Інтернету між користувачами та сервісами тощо [2]. На цьому фоні критичною для технології SOA та сервісів стає *проблема композиції вебсервісів з урахуванням QoS* (**QWSC**), суть якої зводиться до пошуку серед величезної кількості можливих CWS з однаковою функціональністю такого, наскрізна QoS якого буде оптимальною (з точки зору клієнта) [4]. На ранніх етапах розвитку SOC, коли кількість провайдерів і розміри репозиторію сервісів були незначні, проблему QWSC можна було вирішити за допомогою точних та евристичних методів [5,6]. Ефективність цих методів, які погано масштабвалисяся, почала швидко зменшуватись при стрімкому зростанні як кількості сервісів, так і їх користивачів. Наразі для проблеми QWSC характерні багатовимірність, нелінійність цільових функцій, конфліктуючі атрибути QoS, величезний пошуковий простір та значний об'єм вхідних даних тощо. Отже ця проблема є NP-складною і для свого вирішення потребує нових підходів та методів. Наразі серед найбільш успішних розглядаються метаевристичні методи глобальної оптимізації. На відміну від точних методів оптимізації, метаевристика не гарантує оптимальність отриманих рішень. У порівнянні з методами наближення вони не забезпечують доказову точність розв'язку та доказові межі часу виконання. Проте вони здатні забезпечити "практично прийнятні" рішення протягом задовільної тривалості часу.

У статті подано базові знання, необхідні для розуміння особливостей проблеми QWSC, а також представлені сучасні метаевристичні методи її вирішення. Завершується робота висновком.

Передмова

В даному розділі надається коротка базова інформація, яка допоможе краще зрозуміти особливості та складність проблеми QWSC.

Сервіси та їхні властивості. Сервіси — це самоописні, самодостатні, слабко пов'язані, незалежні від платформи та багаторазово використовувані компоненти програмного забезпечення, призначені для підтримки взаємодії між машинами в мережі [3,4]. Вони описуються, публікуються, виявляються та викликаються в розподілених середовищах за допомогою набору станда-

ртів на основі XML (розширювана мова розмітки), включаючи WSDL (мова опису сервісів), SOAP (простий протокол доступу до об'єктів) і UDDI (універсальне виявлення та інтеграція описів).

SAO, яка підпримує життєвий цикл сервісів, складається з трьох основних компонентів: провайдера (постачальника), реєстру сервісів та замовника (користувача, клієнта). На Рис.1 представлена абстрактна модель такої архітектури та взаємозв'язки між її компонентами. Провайдер розробляє сервіс, генерує його опис (WSDL) і публікує його в загальнодоступному реєстрі сервісів (UDDI), роблячи його доступним для виклику. Реєстр містить інформацію для ідентифікації сервісу, включаючи URL-адресу, яка вказує на розташування файлу WSDL, а також технічні відомості про сервіс. Клієнт отримує інформацію про сервіс з реєстру і, якщо вона його влаштовує, використовує відповідний файл WSDL для взаємодії з сервісом через повідомлення SOAP [4].

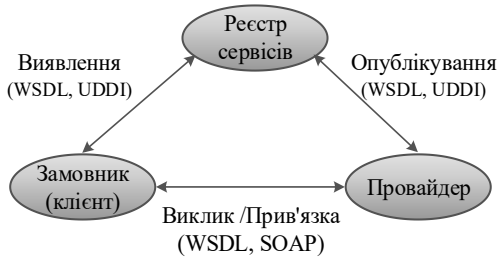


Рис.1. Модель SOA, що використовується сервісами (адаптовано з [2])

Сервіс вважається повністю представлений, якщо добре описані його як функціональні, так і нефункціональні властивості, виражені через певні атрибути QoS [3]. Функціональні властивості описують операційну поведінку сервісів, тобто, яка інформація потрібна для успішного виклику сервісу та яка інформація буде повернена після його виконання**.** З формальної точки зору функціональність сервісу як чорного ящика можна розглядати як певне відображення $\Im$ набору вхідних даних I в набір вихідних даних O, ($\Im$:I→O). Оскільки користувача передусім цікавить, *що робить сервіс*, то пари (I, O) здебільшого достатньо для розуміння цього. В більшості літературних джерел дотримуються саме такого погляду на функціональність сервісів.

Нефункціональні властивості сервісів або атрибути QoS є обмеженнями, визначеними над їхньою функціональністю і використовуються для ранжування сервісів, які мають однакову функціональність. Існує безліч атрибутів QoS для оцінки сервісів з точки зору їхніх нефункціональних властивостей. Проте є декілька атрибутів QoS, які вважаються типовими та поширеними для переважної більшості сервісів від різних провайдерів [4,7]. Це, зокрема:

***вартість, C(Si)*,** що описує, скільки коштує виконання *i*-им сервісом завдання S_i;

***час відповіді, T*(*Si*)** - тобто час, протягом якого можна отримати відповідь від *i*-го сервісу після відправлення запиту на завдання S_i. Значення цієї метрики фактично залежать не тільки від QoS цільового сервісу, а й від якості проміжної мережі;

***надійність, R*(*Si*)**, що описує, наскільки ймовірно, що *i*-ий сервіс дасть очікувану відповідь на завдання S_i;

***доступність, A*(*Si*)**, що описує, наскільки вірогідно отримати доступ до *i*-го сервіса при необхідності виконання завдання S_i;

***репутація, Rep*(*Si*)**, що описує, наскільки завдання *i*-го сервісу S_i , користується перевагою або довірою серед користувачів.

Композиція сервісів з урахуванням QoS. Композиція сервісів — це керований QoS процес об'єднання (агрегації) кількох існуючих в мережі Інтернет сервісів в новий CWS з доданою вартістю, який має необхідні клієнту функціональність та наскрізну QoS. Наразі найбільш поширеним підходом до (напів)автоматичної WSC є підхід на основі робочого бізнес-процесу [4,5]. Підгрунтям цього підходу є подібність абстрактного CWS до абстрактного бізнес-процесу, який являє собою набір загальних сервісів (службових) завдань із визначеними залежностями між потоком управління та потоком даних.

На Рис. 2 показано два ключові етапи життєвого циклу WSC, а саме етапи ***Виявлення*** та ***Вибору*** сервісів. На етапі ***Виявлення*** для кожного *i*-го абстрактного сервіса з функціональністю O_i із множини всіх

доступних в Інтернет сервісів створюється клас конкретних сервісів-кандидатів, які також мають функціональністю O_i, але відрізнятися між собою за QoS. Вхідними даними етапу ***Вибір*** є набір класів сервісів-кандитатів, отриманих на попередньому етапі. Вибираючи з кожного класу по одному сервісу-кандитату, отримаємо множину всіх екземплярів CWS, серед яких необхідно знайти такий, що найкраще відповідає заявленим вимогам та обмеженням користувача до наскрізної QoS.

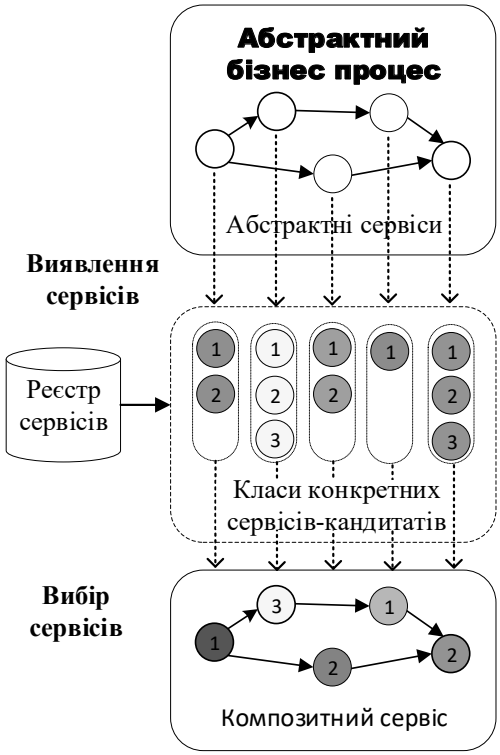


Рис.2. Ключові етапи життєвого циклу композиції сервісу (адаптовано з [4]).

Функції агрегації. В існуючих мовах моделювання бізнес-процесу, таких як BPEL-WS, OWL-S тощо виділяються чотири структури управління завданнями, а саме: послідовна, паралельна, розгалужена та циклічна [8]. Ці ж структури є базовими і для абстрактних CWS (Рис. 3). При п*ослідовній структурі* завдання S_i виконуються в послідовному порядку, i=1,…,m. В п*аралельній структурі* всі паралельні завдання виконуються одночасно. Перехід до завдання за межами структури відбувається після завершення всіх паралельних завдань. В *розгалуженій структурі* з імовірністю p_i виконується завдання S_i і після його завершення відбувається перехід до завдання за межами структури, Σp_i=1. В *структурі циклу* сервіси, створені за її допомогою, виконуються неодноразово, доки не буде виконано певну умову, наприклад, кількість циклів l.

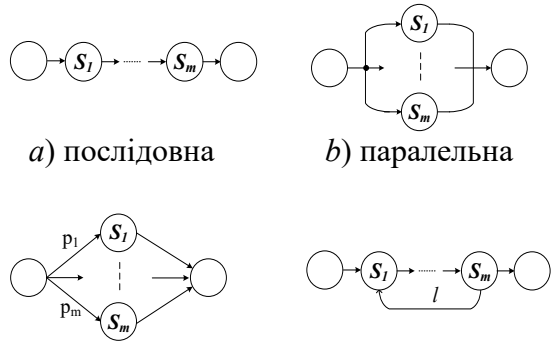


Рис. 3. Базові структури абстрактних CWS

В Табл. 1 наведені функції агрегації для базових структур абстрактних CWS і основних атрибутів QoS [4].

Таблиця 1.

Функції агрегації для QoS, де l — кількість циклів

Атрибут QoS	Структури абстрактних композитних сервісів			
	Послідовна	*Паралельна*	*Розгалужена*	*Циклічна*
Час відповіді	$\sum_{i=1}^{m} T(S_i)$	$max_{i=1}^{m} T(S_i)$	$\sum_{i=1}^{m} T(S_i)\times p_i$	$l\times\sum_{i=1}^{m} T(S_i)$
Надійність	$\prod_{i=1}^{m} R(S_i)$	$\prod_{i=1}^{m} R(S_i)$	$\sum_{i=1}^{m} R(S_i)\times p_i$	$\prod_{i=1}^{m} R(S_i)$
Доступність	$\prod_{i=1}^{m} A(S_i)$	$\prod_{i=1}^{m} A(S_i)$	$\sum_{i=1}^{m} A(S_i)\times p_i$	$\prod_{i=1}^{m} A(S_i)$
Вартість	$\sum_{i=1}^{m} C(S_i)$	$\sum_{i=1}^{m} C(S_i)$	$\sum_{i=1}^{m} C(S_i)\times p_i$	$l\times\sum_{i=1}^{m} C(S_i)$
Репутація	$\frac{1}{m}\sum_{i=1}^{m} Rep(S_i)$	$\frac{1}{m}\sum_{i=1}^{m} Rep(S_i)$	$\sum_{i=1}^{m} P(S_i)\times p_i$	$l\times\sum_{i=1}^{m} P(S_i)$

Формальна постановка задачі QWSC. Нехай

m – кількість абстрактних сервісів в абстрактному композитному сервісі;

k – кількість атрибутів QoS;

G_i – упорядкована множина конкретних сервісів-кандидатів, що мають функціональність i-го абстрактного сервіса, $1 \le i \le$ m;

n_i – кількість сервісів в класі G_i;

x_i – порядковий номер сервісу в класі G_i,

$(1 \le x_i \le n_i)$;

$v_j(x_i)$ – значення j-го атрибута x_i-го конкретного сервісу із G_i, $1 \le j \le$ k, $1\ 1 \le x_i \le n_i$);

$\Re^m$ – множина всіх можливих CWS $X = [x_1, x_2, \ldots, x_m]$, $1 \le x_i \le n_i$, i=1,…,m;

$f_j(X) = f_j\left(v_j(x_1), \ldots, v_j(x_m)\right)$ - агрегатна функція j-го атрибута QoS CWS, $1 \le j \le$ k;

Необхідно знайти такий вектор

$$X^* = [x_1^*, x_2^*, \ldots, x_m^*],\ 1 \le x_i \le n_i,$$

який оптимізує вектор агрегатних функцій

$$F(X) = [f_1(X), f_2(X), \ldots, f_k(X)]$$

У разі наявності обмежень

$$g_i(X) \le 0,\ i = 1,\ldots,\ \text{r},\quad \text{r} \le \text{k};$$
$$h_j(X) \ge 0,\ j = \mathbf{0},\ldots,\ \text{k-r},$$

де $X^* \in \Re^m$ – (суб)оптимальний розв'язок (тобто прийнятний композитний сервіс).

Метаевристичні методи оптимізації QoS композитних сервісів

Одними з перших метаевристик, використаних для вирішення проблеми QWSC, були метаевристики на основі траєкторії, такі як табу пошук [9] та моделювання відпалу [10]. Проте згодом вони поступились потужнішим метаевристикам на основі популяції, зокрема, алгоритмам рійового інтелекту, еволюційним алгоритмам тощо. Оскільки проблема QWSC має тенденцію ускладнюватися, то з часом виявилось, що застосування будь-якого одного алгоритму оптимізації далеко не завжди приводить до успіху. Теоретичним підгрунтям для розуміння подібних емпіричних фактів є відома NFL-теорема Волперта - Макреді про відсутність безкоштовного обіду, яка стверджує, що не існує універсального алгоритму оптимізації, який може розв'язувати всі проблеми краще за інші алгоритми. Одним із очевидних шляхів подолання негативних наслідків NFL-теореми і підвищення ефективності вирішення завдань глобального оптимізації є розробка гібридних алгоритмів, які об'єднують різні або однакові алгоритми, але з різними значеннями вільних параметрів. Мотивацією гібридизації є можливість подолання недоліків окремих алгоритмів, не втрачаючи їхніх переваг, що робить гібридні алгоритми ефективнішими за автономні.

Еволюційні алгоритми

Зазвичай під термінами «еволюційні алгоритми» або «еволюційне обчислення» розуміють велику групу алгоритмів до якої входять, зокрема, генетичні алгоритми (genetic algorithm, **GA**), еволюційні стратегії, еволюційне та генетичне програмування (genetic programing, **GP**), диференціальна еволюція тощо. Вони мають загальну концептуальну базу моделювання еволюції окремих структур і відрізняються за способом формулювання задачі, процесами відбору і використання операторів репродукції (відтворення).

Генетичні алгоритми. Canfora G. та ін. [11] першими застосували генетичний алгоритм для композиції сервісу з урахуванням QoS. Ними був описаний підхід для швидкої, грубозернистої WSC на основі GA, який виявився більш маштабованим, однак повільнішим за цілочисельне програмування. Мета підходу полягала в забезпеченні швидкого способу знаходження субоптимального композитного сервісу. В роботі [12] запропоновано GA, що характеризується схемою кодування хромосом спеціальною матрицею відношень і обробкою різноманітності популяції з моделюванням відпалу. Матриця відношень має здатність одночасно представляти перепланування композитного сервісу, циклічні шляхи та багато сценаріїв сервіса, таких як імовірні-

сний виклик, паралельний виклик, послідовна активація тощо. Zhang C., Ma Y. [13] для вибору сервісів із глобальними обмеженнями QoS запропонували швидкий конвергентний GA з розширеною політикою початкової популяції та еволюційною політикою на основі схеми кодування матриці відношень. А також динамічний GA, *представлений* політикою оцінки динамічної еволюції на основі схеми кодування генома матриці відношень, що забезпечує кращі плани CWS. Vanrompay Y. та ін. [14] для мобільних систем, що складаються з кількох вузлів, запропонували GA для ефективного пошуку майже оптимального щодо загальних атрибутів QoS складу композитного сервісу та його розгортанні на наборі підключених вузлів таким чином, щоб розподіл відповідав заданим обмеженням якості обслуговування (QoS) та мінімізував вартість зв'язку між вузлами.

Gong X.R. та ін. в [15] представили GA композиції сервісу, який підтримує глобальне оптимальне та динамічне перепланування. Алгоритм використовує схему кодування матриці позицій, щоб одночасно виражати всі композитні шляхи та інформацію про перепланування QoS. Jiang Z.Y. та ін. [16] запропонували модель оптимізації запитів і відповідний GA на основі багатоатрибутного агрегування QoS різних сервісів для індивідуальності та ефективності в оцінках. Модель налаштовує QoS з глобальними обмеженнями та уподобаннями користувачів, динамічною схемою рейтингу та багаторівневою відповідністю. Авторами роботи [17] для вирішення проблеми QWSC по-перше, представлено GA відновлення з мінімальними конфліктами, в якому окремі особини можуть бути швидко виправлені за допомогою евристики впорядкування значень, коли вони порушують обмеження міжсервісних залежностей і конфліктів. По-друге, для вирішення проблеми розподілення CWS запропоновано GA на основі штрафів, оскільки розумне розподілення композитних сервісів між паралельними серверами може збільшити пропускну здатність. У GA вбудований швидкий локальний оптимізатор та ранговий відбір із елітарністю. Cao J. та ін. [18] запропонували удосконалений GA для оптимізації якості моделі процесу обслуговування сервісів, яка спочатку перетворюється на дерево структури процесу для розрахунку якості, а потім GA використовується для уточнення процесу обслуговування. В роботі [19] представлено інтегровану структуру запитів до сервісів, яка включає в себе модель запитів до сервісів і двофазову стратегію оптимізації. Модель запиту визначає сервісні спільноти для організації великого та неоднорідного простору обслуговування. Двофазова стратегія оптимізації з жадібним алгоритмом і GA здатна «спільно розвивати» кілька можливих планів виконання одночасно. Klein A. та ін. [20] описали мережевий підхід до композиції сервісів у хмарі, що складається з мережевої моделі розрахунку QoS та алгоритму вибору, який ґрунтується на GA. Ukor R. та ін. в [21] представили задачу оптимізації вибору сервісу з поняттям маркерів запиту та трьома підходами (цілочисельне програмування, евристичний пошук по дереву та GA) для вирішення цієї задачі. Автори роботи [22], розглядаючи одночасно функціональні та нефункціональні вимоги, запропонували семантичний, заснований на GA, підхід до автоматичної композиції сервісів. У роботі [23] запропоновано коеволюційний GA для створення сервісу на основі QoS, який повністю враховує індивідуальні відносини між популяціями.

Багатоцільові GA. Claro D.B. та ін. [24] для вибору сервісів для оптимальної композиції реалізували багатоцільовий еволюційний підхід, заснований на NSGA-II (генетичний алгоритм недомінованого сортування), який ідентифікує набір оптимальних рішень за Парето без введення ранжирування серед різних параметрів QoS. В [25] також використовано структуру NSGA-II для композиції сервісів з урахуванням QoS. Водночас для полегшення ухвалення рішення споживачам сервісів надавали кілька різних можливих розв’язків. Hashmi K. та ін. [26] на основі NSGA-II представили сервіс переговорів, який використовується як споживачем, так і сервісом провайдера для проведення переговорів щодо залежних параметрів QoS. Авторами [27] запропоновано алгоритм глобальної багатоцільової оптимізації WSSPEA2. Суть його полягає в

тому, що задача вибору вебсервісів на основі QoS трансформується в багатоцільову задачу оптимізації композиції сервісів з обмеженнями QoS. Силовий еволюційний алгоритм Парето (SPEA2) використовується для отримання набору оптимальних за Парето рішень за допомогою одночасної оптимізації ряду цільових функцій, і користувачі можуть вибрати одне з цих рішень за своїми перевагами. В [28] представлено алгоритм GODSS (глобальний оптимальний вибір динамічних сервісів) для вирішення динамічного вибору сервісів із глобальними оптимальними QoS у композиції сервісів. Суть алгоритму полягає в тому, що задача динамічного вибору сервісу з глобальними оптимальними QoS трансформується в багатоцільову оптимізацію композиції сервісів з обмеженнями QoS. Wang J. та ін. [29] запропонували багатоцільовий GA для оптимального вибору сервісів на основі QoS, враховуючи такі обмеження у процесі вибору сервісу, як структура управління в плані композиції, взаємозв'язок між конкретними сервісами та компроміс між кількома атрибутами QoS. В [30] запропоновано багатоцільовий GA, на ефективність якого не впливає еволюціонуючий набір рішень на кожній ітерації. Набір оптимальних рішень за Парето будується за правилом псевдобінарного дерева. Потім оптимальні за Парето рішення упорядковуються, і придатність кожного оптимального рішення за Парето визначається подібністю окремих рішень. Wagner F. та ін. [31] описали розширену операцію відновлення та надали багатоцільовий алгоритм оптимізації, який використовує базові знання для виявлення надійних оптимізованих за QoS виборів сервісів у відкритому сервісному середовищі. Алгоритм враховує вартість потенційних збоїв та ранжує рішення на основі переваг ризику особи, яка ухвалює рішення. Ramirez A. та ін. [32] досліджували придатність еволюційного алгоритму з багатьма цілями для вирішення проблеми зв'язування сервісів на основі реального тесту з дев'ятьма атрибутами QoS.

Гібридні GA. Ma X. та ін. [33] вперше обговорили обмеження методу простих адитивних ваг через те, що ці ваги не мають фізичного значення і їх важко встановити. Потім пропонується новий ефективний обчислювальний підхід, лінійне фізичне програмування у співпраці з GA для вибору сервісу, керованого якістю. Liang W. та та ін. [34] запропонували гібридний алгоритм для композиції сервісів, який включає GA та грубу теорію множин. Цей алгоритм може: (i) вирішувати проблеми, які можна розкласти на функціональні вимоги, і (ii) покращувати продуктивність GA, виявляти нездійснені рішення та винятки шляхом зменшення діапазону домену початкової популяції та обмеженого перетину, використовуючи грубу теорію множин. Liu Z. та ін. [35], базуючись на декомпозиції глобальних обмежень QoS, запропонували метод динамічної композиції, який складається з трьох етапів: (i) глобальні обмеження QoS розкладаються на локальні обмеження культурного GA; (ii) правила визначення QoS визначають значення QoS сервісів-кандидатів; (iii) найкращі сервіси, що задовольняють локальні обмеження, вибираються для кожного завдання протягом часу виконання. Que Y. та ін. [36] базуючись на ідеї декомпозиції QoS, реалізували гібридне рішення з інформаційною ентальпією, успадкованою від штучної імунної системи, для вирішення проблеми детермінованої ймовірності оператора GA, яка призводить до проблеми передчасної конвергенції. В [37] розроблено керований кластером GA шляхом впровадження k-середніх у процес генерації початкової популяції. Крім того, Jatoth C. та ін. [38] представили оптимальну композицію хмарного сервісу з використанням GA на основі адаптивної еволюції генотипу, що має справу з кількома атрибутами QoS і забезпечує рішення, які задовольняють баланс параметрів QoS і обмежень підключення композиції сервісу

Генетичне програмування. В роботі [39] запропоновано алгоритм GP композиції сервісів, який може об'єднувати сервіси, використовуючи різні керуючі структури, генерувати композиції відповідно до контекстно-вільної граматики та явно керувати оновленням атрибутів. А також мінімізує кількість сервісів і шукає композиції з мінімальним шляхом виконання. Авторами роботи [40] запропоновано алгоритм адаптивного GP, спрямова-

ний на створення бажаних виходів на основі доступних вхідних даних, а також на забезпечення того, щоб композитний сервіс мав оптимальне значення QoS. В роботі [41] представлено підхід на основі GP до композиції розподіленого сервісу з урахуванням QoS. Вплив каналів зв'язку на атрибути QoS вибраних сервісів розглядається явно. Щоб впоратися з розподіленим середовищем сервісів, була прийнята мережева система координат для оцінки часу, витраченого на зв'язок між сервісами, а також між сервісами та кінцевими користувачами.

Гібридне GP — це особливе застосування GA, яке широко використовується у повністю автоматизованій стратегії композиції з графовим поданням. У повністю автоматизованій композиції абстрактні робочі процеси створюють вихідно-вхідні зв'язки між сервісами, де оператор алгоритму використовується в еволюційному процесі для вибору сервісу композиції без обмежень. В [42] пропонується гібридний підхід до складання сервісів, який поєднує GP та випадковий жадібний пошук. Жадібний алгоритм використовується для генерації допустимих та локально оптимізованих особин для GP та для виконання операцій мутації під час генетичного програмування. Yu Y. та ін. [43] пропонують гібридний підхід, який об'єднує використання GP та табу-пошуку для композиції сервісу з інтенсивним використанням даних з урахуванням QoS. Ефективність запропонованого підходу оцінюється за допомогою загальнодоступних наборів даних. У статті [44] запропоновано комбінацію GP та випадкового жадібного пошуку для композиції сервісу. Жадібний алгоритм використовується для створення дійсних і локально оптимізованих особин для заповнення початкового покоління для GP та для виконання операцій мутації під час генетичного програмування.

Алгоритми ройового інтелекту

Це парасольковий термін для великої групи алгоритмів, до якої, зокрема, входять алгоритми *оптимізації рою частинок* (particle swarm optimisation, **PSO**), *оптимізації мурашиної колонії* (ant colony optimisation, **ACO**), оптимізації штучної бджолиної колонії, (artificial bee colony algorithm, **ABC**) тощо.

Оптимізація рою частинок. Wang W. та ін. [45] для вирішення проблеми QWSC запропонували вдосконалений алгоритм оптимізації рою частинок (iPSOA) із стратегіями нерівномірної мутації та адаптивного коригування ваги для покращення швидкості конвергенції на глобальному та локальному рівнях відповідно. А також представлено модель пулу сервісів, яка допомогає користувачам зберегти більше композитних планів за один процес вибору та ефективний алгоритм побудови пулу сервісів, заснований на вдосконаленій оптимізації дискретного рою частинок (IDPSO). Водночас стратегія нерівномірної мутації була введена в глобальну найкращу частинку в IDPSO для підвищення якості складних планів у пулі сервісів. Wang S. та ін. [46] запропонували швидкий PSO у середовищі хмарних обчислень для створення хмарних CWS. Спочатку оператор горизонту повинен видалити надлишкові кандидати CWS, а потім PSO вибрає CWS із решти кандидатів для створення композитних сервісів. В роботі [47] розроблено PSO для полегшення динамічного вибору сервісу, в якому запропоновано три типи операторів швидкості та одне рівняння еволюції положення. Критерій no-hope/re-hope гарантує різноманітність PSO та покращує можливість глобального пошуку. Wang L. та ін. [48] розробили автоматичний алгоритм композиції сервісів, заснований на глобальній оптимізації QoS та хаотичному PSO. Водночас модель вибору сервісів із глобальною оптимізацією QoS перетворюється на задачу багатоцільової оптимізації. В роботі [49], для реалізації ефективної інтеграції та уніфікованого управління ресурсами *логістичного центру* запропоновано метод вираження логістичних ресурсів та інкапсуляції сервісів, щоб знайти найкращий конкретний сервіс, прив'язаний до кожного відповідного абстрактного сервісу. Автори [50] запропонували групування PSO на основі переваг користувача у виборі сервісу. Алгоритм не тільки досягає балансу між швидкістю конвергенції та розмаїттям роїв, а й дозволяє максимально задовольнити користувачів за рахунок стратегій угруповання частинок, розвитку різних кластерів окремо,

своєчасного оновлення кожного кластера та використання нечітких обмежень для вираження уподобань користувача. В роботі [51] об'єднано PSO з алгоритмом присвоєння Мункреса для композиції сервісу, який відрізняється від споріднених робіт двома аспектами: (i) для вирішення проблеми оптимізації вводяться два метаевристичні методи, засновані на PSO та (ii) одночасно обробляються декілька запитів робочого процесу.

Багатоцільова PSO. Cao J. та ін. [52] запропонували багатоцільовий алгоритм вибору сервісів на основі PSO, який моделює проблему вибору сервісів як обмежену оптимізаційну задачу з кількома цілями. Сервіси в кожному наборі суб-сервісів спочатку сортуються відповідно до концепції домінування Парето. Потім створюється новий набір суб-сервісів, розмір якого набагато менший за початковий, і, нарешті, виводиться оптимальний набір за Парето. Автори роботи [53] перетворили задачу ранжирування top-k у задачу багатокритеріального програмування, коли переваги користувачів розглядаються разом із запитами користувачів. Потім пропонується вдосконалений дискретний PSO, щоб зведені рейтинги задовольняли як запити користувачів, так і їхні переваги. Guha T. та ін. [54] запропоновано та порівняно два алгоритми вибору сервісів в Grid: багатоцільовий алгоритм оптимізації рою частинок з кількома цілями з використанням методу відстаней скупченості (MOPSO-CD) до алгоритму встановлення відповідності на основі задоволення обмежень (CS-MM). Fan X. [55] побудував ефективний багатоцільовий алгоритм PSO з кількома обмеженнями за допомогою нішевої технології для вирішення проблеми вибору та композиції персоналізованого сервісу. Авторами [56], враховуючи кореляції між ресурсними сервісами, запропоновано багатоцільовий PSO для вирішення проблеми вибору та композиції ресурсів множинної мережі. Основні особливості: (i) він поєднує техніку недомінованого сортування для вибору найкращих глобальної та локальної позицій; (ii) формула оновлення частинок динамічно змінюється для досягнення компромісу між глобальним дослідженням та локальною експлуатацією; і (iii) для підтримки різноманітності популяції застосовуються оператори обрізання популяції на основі перестановок і цілей.

Гібридна PSO — це одна із найвідоміших груп алгоритмів ройового інтелекту, яка у існуючій літературі згадується як друга за частотою гібридна метаевристика. Xu X. та ін. [57] для збільшення ефективності пошуку оптимального композитного сервіса запропонували гібридний алгоритм оптимізації хаосу (PS-CTPSO), який є поєднанням PSO та стратегії хижацького пошуку. Liu Y. та ін. [58] для вирішення проблеми композиції сервісів пропонують гібридний алгоритм оптимізації рою квантових частинок (HQPSO), який поєднує PSO з деякими принципами квантової механіки. Yin H. та ін. [59] для покращення різноманітності роїв впроваджують генетичні оператори в PSO. Крім того, Wang S. та ін. [60] запропонували покращити ефективність PSO за допомогою методів горизонту для видалення надмірностей у сховищі *сервісів*. У цьому ж контексті Hossain M. S. та ін. [61] для розв'язання проблеми отримання оптимальної композиції за мінімальний час із численних наборів динамічних сервісів, запропонували гібридний алгоритм, який поєднує PSO і кластеризацію k-середніх. Він працює паралельно за допомогою MapReduce на платформі Hadoop. Це важливо для обробки великих обсягів різнорідних даних і сервісів із різних джерел у мобільному середовищі. Chifu V. та ін. [62] застосували метод кластеризації даних для визначення щільної області у просторі пошуку для створення надійної можливості пошуку для знаходження глобальних оптимумів та уникнення локальних пасток. Gharbi M. та ін. [63] запропонували метод під назвою реляційний аналіз концепції (RCA) для зменшення простору пошуку. Евристичні методи, такі як алгоритм клонального відбору [64] і гармонійний пошук [65], об'єднані з оператором горизонту та стратегією хижацького пошуку [57], для підвищення можливостей пошуку традиційного PSO. В [66] реалізувано повністю автоматичну композицію сервісів за допомогою алгоритму планування. Гібридний PSO з метаевристиками, включаючи алгоритм штучної імунної системи [67] і гармонійний пошук [65], запропонований

для гарантування тонкого балансу між дослідженням та експлуатацією. Haytamy S. та ін. [68] розробили двоступінчасту гібридну модель, в якій вихідні дані рекурентної нейронної мережі LSTM подавалися в PSO для перетворення даних QoS у послідовну інформацію. Крім того, Hosseinzadeh M. та ін. [69] розробили гібридний PSO зі штучною нейронною мережею для покращення параметрів QoS. Sun та ін. [70] запропонували підхід швидкого вибору сервісів, який повністю використовує PSO з адаптивними стрибками та нечітким логічним керуванням для адаптивного розкладання глобальних обмежень QoS на локальні з адаптивним рівнем якості. В [71] для проблеми QWSC запропоновано гібридний PSO з принципом клонального відбору та кількома корисними стратегіями. Удосконалена локальна стратегія «кращий перший» представлена для досягнення однакових впливів на локальну придатність компонентного завдання та функцію придатності композитного сервісу, коли замінюється інший сервіс-кандидат. У роботі [72] запропоновано новий алгоритм кооперативної еволюції на основі PSO та SA, який успадковує швидку конвергенцію PSO та глобальну конвергенцію SA, а також долає недоліки схильності до захоплення локального оптимуму PSO та повільної конвергенції SA. Багатоцільова PSO розроблена також для багатоцільової композиції сервісів із глобальною оптимізацією QoS. У роботі [73] представлена модель вибору сервісів на основі PSO для визначення оптимальної комбінації сервісів шляхом оцінки атрибутів QoS для динамічної програми електронного бізнесу. Модель реалізована багатоагентною системою, де агенти можуть представляти автономних запитувачів сервісів, постачальників сервісів та інші функції в програмах електроного бізнесу. Fethallah H. та ін. [74] для проблеми QWSC запропонували реактивне багатоагентне рішення на основі локальної версії PSO, у якій кожна група має повну сітчасту топологію, яка більш несприйнятлива до локальних оптимумів, ніж глобальна версія.

Оптимізація мурашиної колонії. Zheng X. в [75] представив алгоритм вибору сервісу з урахуванням QoS, до якого включена модель ступеня задоволеності користувача для грід-сервісів і розширений ACO з новим правилом мурашиного клонування. Авторами роботи [76] запропоновано модель мінімізації вартості композиції сервісів для додатків з інтенсивним об'ємом даних і розроблено відповідний алгоритм ACO для її реалізації. Сервіси навколишнього медіа в середовищі моніторингу розумного будинку повинні бути повсюдними, адаптивними та надійними щодо доступу та доставки. Hossain M. та ін. [77] запропоновано структуру вибору сервісів у "розумних" середовищах, яка використовує потенціал підходу до вибору сервісів на основі мурашок, враховуючи динамічність уподобань та задоволеності користувачів. А це є ключовим фактором для вибору медіа-сервісів, щоб знайти найкращий спосіб вибору відповідних медіа-сервісів. Це також дозволяє різним категоріям мешканців отримувати доступ до різноманітних медіа-сервісів таким чином, що їхній досвід оптимізується з огляду на навколишнє середовище. В роботі [78] запропоновано натхненний мурашками метод вибору семантичного сервісу, який використовує граф композиції та багатокритеріальну функцію для визначення оптимального рішення композиції відповідно до вподобань користувача щодо QoS та семантичної якості. Щоб підвищити ефективність і точність процесу виявлення сервісів, вони організували набір доступних сервісів зі схожою функціональністю в кластери сервісів. Wang R. та ін. [79] у поєднанні з механізмом позитивного зворотного зв'язку ACO спочатку проаналізували проблему вибору сервісів із базовим принципом ACO, а потім перетворили проблему вибору сервісів з урахуванням QoS у проблему найкоротшого шляху. Нарешті вони вказали кроки вирішення проблеми вибору сервісу на основі ACO та порівняли ефективність алгоритму за різними параметрами. В роботі [80] представлено жадібний алгоритм під назвою Greedy-WSC і алгоритм на основі оптимізації колонії мурашок під назвою ACO-WSC, які намагаються вибрати можливі комбінації хмар і використовувати мінімальну кількість хмар. Експериментальні результати показують, що запропонований метод оптимізації мурашиної

колонії може ефективно та результативно *знаходити хмарні комбінації з мінімальною кількістю хмар.*

Багатоцільова ACO. Fang Q. та ін. [81] застосували багатоцільовий алгоритм *оптимізації мурашиної* колонії для вирішення проблеми вибору *динамічного сервісу*. Модель вибору сервісу з глобальними обмеженнями QoS перетворюється на задачу багатоцільової оптимізації з обмеженнями користувача. Zhang W. та ін. [82] запропонували багатоцільове моделювання вибору оптимального шляху для композиції *динамічних сервісів* на основі QoS. Представлено стратегію декомпозиції композитного сервісу із загальною структурою потоку на паралельні шляхи виконання, яка є масштабованою для підтримки композиції дуже складних сервісів. Dahan F. та ін. в [83] розробили алгоритм оптимізації колонії літаючих мурашок під назвою FACO (Flying Ant Colony Optimization), який модифікує ACO для вирішення проблеми композиції сервісу з урахуванням QoS. Вони також представили алгоритм EFACO (Enhanced Flying Ant Colony Optimization), який зменшує складність обчислень FACO, запровадивши в ньому три вдосконалення 1) щоб уникнути проблеми з часом виконання, EFACO обмежує процес польоту лише тоді, коли якість рішення покращується; 2) щоб уникнути сканування всіх сусідніх вузлів застосували метод вибору сусідніх вузлів; 3) ввели третю модифікацію, яка перетворила алгоритм на мультиферомонний алгоритм.

Гібридна ACO. Yang Z. та ін. [84] розробили гібридний ACO, за допомогою якого використовувався генетичний алгоритм для вибору критичного параметра для ACO. Генетичний алгоритм відомий своїм гнучким, надійним механізмом пошуку, але інколи він страждає від відсутності надійних можливостей пошуку глобальних оптимумів. З іншого боку, ACO має репутацію глобального пошукового алгоритму, але страждає від повільної конвергенції, особливо великомасштабних проблем. Отже, Yang Y. та ін. [85] запропонували гібридизацію, де алгоритм мурашиної колонії служив зачатком генетичної операції. Liu Z.-Z. та ін. [86] інтегрували систему мурахи в алгоритм культури. Погана стагнація була описана як підводне каміння ACO в існуючій літературі. Alayed H. та ін. [87] покращили алгоритм ACO, щоб збільшити різноманітність і уникнути стагнації, втіливши процес обміну. Гібридний метод композиції динамічних сервісів [88] представлено на основі методу оптимального шляху зваженого орієнтованого ациклічного графа, який поєднує ACO та GA для проблеми QWSC. Пропонується новий динамічний генетичний гібридний алгоритм мурашиної колонії [89] для визначення часу запуску генетичних алгоритмів і алгоритмів мурашиної колонії для найкращої стратегії оцінки злиття. Отже, здатність до оптимізації максимізується, а загальна швидкість конвергенції прискорюється. В роботі [90] для проблеми QWSC пропонується алгоритм оптимізації C-MMAS (culture max-min ant system), отриманий шляхом інтеграції системи Max–Min ant у структуру культурного алгоритму. Алгоритм складається з простору популяцій, простору переконань і протоколів зв'язку між ними. Популяційний простір містить ряд рішень. Простір переконань зберігає кращі рішення, отримані в еволюційному процесі популяційного простору; ці рішення розглядаються як досвід і знання та використовуються для скерування еволюції C-MMAS у популяційному просторі. Окрім того, розроблено комплексну модель (DGQoS) оцінки CWS, заснованій на загальній QoS та доменній QoS, і новий алгоритм оптимізації C-MMAS для вирішення проблеми вибору сервісу на основі DGQoS.

В останні роки зі швидким розвитком мереж мобільного зв'язку продовжують з'являтися деякі нові сервіси, такі як хмарна віртуальна реальність, голографічний зв'язок тощо. Тому потрібні більш гнучкі та інтелектуальні алгоритми композиції сервісів. Виходячи з цього, в [91] запропоновано стратегію композиції сервісу в багатохмарному середовищі, а також алгоритм ACO на основі механізму мультиферомонів для оптимізації QoS. Щоб уникнути появи локальних оптимумів, ми додатково вводимо операцію мутації генетичного алгоритму.

Інші метаевристики композиції сервісів

Jiang B. та ін. [92] для підвищення ефективності композиції сервісу запропонували підхід на основі вдосконаленого алгоритму феєрверків (fireworks algorithm). В [93] представлено розширений багатоцільовий алгоритм диференціальної еволюції для пошуку репрезентативного набору рішень із хорошою близькістю та дистрибутивністю. У [94] для вирішення проблеми композиції сервісу використовується варіант алгоритму пошуку гармонії (Harmony Search), який знаходить оптимальний композитний сервіс, що задовольняє локальні та глобальні обмеження користувача щодо атрибутів якості. Ghobaei-Arani M. та ін. [95] представили алгоритм *пошуку зозулі* (cuckoo search, CS), який здійснює композицію сервісу для покращення QoS у розподіленому хмарному середовищі. Dahan F. [96] запропонував покращений алгоритм оптимізації кита (IWOA), який містить три різні стратегії для підвищення продуктивності базового алгоритму оптимізації кита (Whale Optimization Algorithm, WOA) за рахунок збільшення швидкості збіжності, яка створює низьку точність рішення для проблеми композиції сервісів**.** У [97] для знаходження композиції хмарного сервісу з оптимальними значеннями QoS запропоновано гібридний алгоритм, який поєднує алгоритми стратегії орла (Eagle Strategy Algorithm) з WOA, що покращало швидкість збіжності та забезпечило належний баланс між дослідженням та експлуатацією. Dahan F. [98] пропонує алгоритм, заснований на поведінці мікрокажанів під час полювання на здобич. Запропонований алгоритм усуває деякі недоліки класичного алгоритму кажана (Bat Algorithm, BA) і визначає оптимальну комбінацію сервісів для задоволення потреб користувача. У роботі [99] вдосконалено алгоритм штучної бджолиної колонії, щоб зробити його більш придатним для проблеми композиції сервісів. Запропоноване вдосконалення контролює стратегії експлуатації та дослідження таким чином, щоб заохочувати дослідження на ранніх стадіях, а експлуатацію на більш пізніх стадіях. Pop C. та ін. [100] представили гібридний алгоритм, який поєднує навчання з підкріпленням із метаевристиками CS та табу пошук. У цьому алгоритмі навчання з підкріпленням відстежує заміну сервісу, тоді як табу пошук використовується для оптимізації процесу вибору з точки зору часу виконання та досліджуваного простору пошуку. Простір пошуку моделюється як структура розширеного графа планування, яка кодує всі можливі рішення композиції для заданого запиту користувача. Щоб встановити, чи є рішення оптимальним, атрибути QoS сервісів, а також семантична подібність між ними, розглядаються як критерії оцінки. Chifu V. та ін [101] запропонували алгоритм оптимізації спаровування медоносних бджіл у поєднанні з компонентами генетичного алгоритму, табу пошуку та навчання з підкріпленням. У роботі [102] запропоновано алгоритм пошуку вусиків жука (Beetle Antennae Searching Algorithm) інтегрувати в PSO для створення кращої початкової популяції та, як наслідок, досягнення швидшої конвергенції. Dahan F. та ін. [103] представили гібридний алгоритм між ABC та CS, щоб збільшити ефективність композиції сервісів. Алгоритм CS використовується для подолання повільної конвергенції ABC, дозволяючи покинутим бджолам покращити свій пошук і перекрити локальний оптимум. Ahanger T. та ін. [04] запропонували гібридний алгоритм між ABC і BA для досягнення кращого компромісу між локальним використанням і глобальним пошуком. Peng та ін. [105] запропонували багатокластерну адаптивну оптимізацію мозкового штурму, об’єднану з подвійною опорною векторною машиною для зменшення простору пошуку.

Висновки

З появою і розвитком SAO еволюція складності проблеми композиції сервісів проходила в напрямку її зростання у міру збільшення кількості та асортименту сервісів, обчислювальних парадигм та можливостей мереж Інтернет. Водночас методи її розв’язання переходили від точних (без евристичних) підходів до майже оптимальних евристик і, нарешті, до метаевристик глобальної отимізації, які є домінуючими протя-

гом останніх двох десятиліть. Проведений в даній роботі репрезентативний огляд і аналіз метаевристичних методів дозволяє зробити ряд висновків, зокрема:

Фундаментальна мета дослідження композиції сервісу полягала в досягнені швидкої конвергенції та стабільних методів, які можуть знайти високоякісні рішення для користувачів, в умовах динамічно змінного середовища сервісів, незалежно від скадності проблеми QWSC.

Метаевристики значною мірою виправдали покладені на них очікування. Проте поява останніми роками нових обчислювальних парадигм з обмеженими ресурсами, таких як інтернет речей, туманне та хмарне обчислення тощо суттєво ускладнили проблему QWSC. Емпіричні дані показують, що досягнення фундаментальної мети в рамках автономних метаевристик стає все складнішим. Швидка конвергенція, уникнення локального захоплення, робота з великим простором пошуку та досягнення високоякісного рішення зміщують дослідження композиції сервісів у бік розробки гібридних метаевристик, які можуть досягти тонкого балансу між дослідженням та експлуатацією. Гібридна метаевристика є багатообіцяючою спробою вийти за межі метаевристики через об'єднання декількох метаевристик таким чином, щоб подолати недоліки кожної з них, не втрачаючи водночас їхніх переваг. Поєднання машинного навчання з метаевристикою стало останньою тенденцією в цій галузі. Більше того, нещодавня гібридизація машинного навчання та генетичного програмування в рамках повністю автоматизованої композиції сигналізує про рух до парадигми гіперевристики.

References

1. Huhns M. N., Singh M. P., Service-oriented computing: Key concepts and principles, IEEE Internet Comput., vol. 9, no. 1, pp. 75–81, Jan. 2005.
2. Papazoglou M. P., Heuvel W.-J., Service oriented architectures: approaches, technologies and research issues, VLDB J. Vol.16 (3) (2007) 389–415.
3. O'Sullivan J., Edmond D., Hofstede A.T., What's in a service?, Distrib. Parallel Databases, 2002, 12, (2–3), pp. 117–133
4. Bouguettaya A., Sheng Q.Z., Daniel F. (Eds.), Web Services Foundations, Springer, 2013, P.739
5. Gabrel V., Manouvrier M., Murat C., Optimal and automatic transactional web service composition with dependency graph and 0-1 linear programming, in Proc. Int. Conf. Service-Oriented Comput. Berlin: Springer, Nov. 2014, pp. 108–122.
6. Li J., Zhao Y., Liu M., et al.: An adaptive heuristic approach for distributed QoS-based service composition, in Proc. IEEE Symp. Comput. Commun., Jun. 2010, pp. 687–694.
7. Dongre Y., Ingle R.: An investigation of QoS criteria for optimal services selection in composition, in Proc. 2nd Int. Conf. Innov. Mech. Ind. Appl. (ICIMIA), Mar. 2020, pp. 705–710.
8. Mili H., Tremblay G., et al.: Business Process Modeling Languages: Sorting Through the Alphabet Soup. ACM Computing Surveys, 11, 2010, P.57
9. Pop C., Vlad M., Chifu V., et al.: A tabu search optimization approach for semantic web service composition, in Proc. 10th Int. Symp. Parallel Distrib. Comput., Jul. 2011, pp. 274–277.
10. Liu Q., Zhang S.-L., Yang R., et al.: Web services composition with QoS bound based on simulated annealing algorithm, J. Southeast Univ., vol. 24, no. 3, 2008, pp. 308–311.
11. Canfora G., Penta M.D., Esposito R., et al.: A lightweight approach for QoS-aware service composition, Proc. 2nd Int. Conf. on Service Oriented Computing (ICSOC'04), NY, USA, 2004, pp. 36–47
12. Zhang C.W., Su S., Chen J.L.: DiGA: population diversity handling genetic algorithm for QoS-aware web services selection, Comput. Commun., 2007, 30, pp. 1082–1090
13. Zhang C.W., Ma Y.: Dynamic genetic algorithm for search in web service compositions based on global QoS evaluations. IEEE Int. Conf. on Scalable Computing and Communications, 2009, pp. 644–649
14. Vanrompay Y., Rigole P., et al.: Genetic algorithm-based optimization of service composition and deployment. 3rd Int. workshop on Services integration in pervasive environments, 2008, pp. 13–17
15. Gong X.R., Zhu Q.S., Wu C.L., et al.: Web services composition supporting global optimal and dynamic re-planningof QoS, Comput. Integr. Manuf. Syst.,2008, 14, (10), pp. 2068–2075
16. Jiang Z., Han J., Wang Z.: An optimization model for dynamic QoS-aware web services selection and composition, Chin. J. Comput., 2009, 32, (5), pp. 1014–1025
17. Ai L., Tang M., Fidge C.: Partitioning composite web services for decentralized execution using a genetic algorithm, Future Gener. Comput. Syst., 2011, 27, pp. 157–172
18. Cao J., Wang J., Zhao H., et al.: A service process optimization method based on model refinement, J. Supercomput., 2013, 63, pp. 72–88
19. Yu Q., Rege M., Bouguettaya A., et al.: A two-phase framework for quality-aware web service selection, Serv. Oriented Comput. Appl., 2010, 4, pp. 63–79
20. Klein A., Ishikawa F., Honiden S.: Towards network-aware service composition in the cloud. WWW 2012, Lyon, France, 2012, pp. 959–968

21. Ukor R.: Service selection and horizontal multi-sourcing in process-oriented capability outsourcing, J. Softw., Evol. Process, 2012, 24, pp. 259–283
22. Fanjiang Y.-Y., Syu Y.: Semantic-based automatic service composition with functional and non-functional requirements in design time: a GA approach, Inf. Softw. Technol., 2014, 56, (3), pp. 352–373
23. Li Y.Z., Hu J., et al.: Research on QoS service composition based on coevolutionary genetic algorithm, Soft Comput., 2018, 22, pp. 7865–7874
24. Claro D.B., Albers P., Hao J.K.: Selecting web services for optimal composition. Int'l Conf. Web Services (ICWS'05), Orlando, FL, USA, 2005
25. Yao Y.J., Chen H.P.: Qos-aware service composition using NSGA-II. 2nd Int. Conf. on Interaction Sciences, Seoul, Korea, 2009, pp. 358–363
26. Hashmi K., Alhosban A., Najmi E., et al.: Automated web service quality component negotiation using NSGA-2. ACS Int. Conf. on Computer Systems and Applications, Ifrane, 2013, pp. 1–6
27. Li J.Z., Luo W.L., Zeng J.T., et al.: Application of SPEA2 algorithm in web services selection. IEEE Youth Conf. on Information Computing and Telecommunications, Beijing, 2010, pp. 387–390
28. Liu S., Liu Y.: A dynamic web services selection algorithm with QoS global optimal in web services composition, *J. Softw.*, 2007, 18, (3), pp. 646–656
29. Wang J.L. *et al*.: Optimal web service selection based on multi-objective GA. Int. Symp. on Comp. Intellig. and Design, 2008, pp. 553–556
30. Hu H., Dong W., et al.: Pareto optimality based genetic algorithm in web services composition, *J. Xi'an Jiao Tong Univ.*, 2009, 43, (12), pp. 50–54
31. Wagner F., Klopper B., et al.: Towards robust service compositions in the context of functionally diverse services. WWW, 2012, pp. 969–978
32. RamĺŁrez A., Parejo J., Romero J., et al.: Evolutionary composition of QoS-aware web services: A many-objective perspective, *Expert Syst. Appl.*, 2017, 72, pp. 357–370
33. Ma X.N., Dong B.T.: Linear physical programming based approach for web service selection. 2008 Int. Conf. on Information Management, Innovation Management and Industrial Engineering, Xian, China, 2008, pp. 398–401
34. Liang W.Y., Huang C.C.: The generic genetic algorithm incorporates with rough set theory – An application of the web services composition, *Expert Syst. Appl.*, 2009, 36, pp. 5549–5556
35. Liu Z., Xue X., Shen J., *et al*.: Web service dynamic composition based on decomposition of global QoS constraints, *Int. J. Adv. Manuf. Technol.*, 2013, 69, pp. 2247–2260
36. Que Y., Zhong W., Chen H., *et al*.: Improved adaptive immune genetic algorithm for optimal QoS-aware service composition selection in cloud manufacturing, *Int. J. Adv. Manuf. Technol.*, vol. 96, nos. 9-12, pp. 4455-4465, Jun. 2018.
37. Sadeghiram S., Ma H., Chen G.: Cluster-guided genetic algorithm for distributed data-intensive web service composition, in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1-7.
38. Jatoth C., Gangadharan G.: Optimal fitness aware cloud service composition using an adaptive genotypes evolution based genetic algorithm, *Future Gener. Comput. Syst.*, vol. 94, pp. 185-198, 2019.
39. Rodriguez-Mier P., Mucientes M., Lama M., et al.: Composition of web services through genetic programming, *Evol. Intell.*, 2010, 3, pp. 171–186
40. Yu Y., Ma H., Zhang M.: An adaptive genetic programming approach to QoS-aware web services composition. IEEE Congress on Evolutionary Computation, Cancun, 2013, pp. 1740–1747
41. Yu Y., Ma H., Zhang M.: A Genetic Programming approach to distributed QoS-aware web service composition. IEEE Congress on Evolutionary Computation 2014: 1840-1846
42. Ma H., Wang A., Zhang M.: A hybrid approach using genetic programming and greedy search for QoS-aware web service composition, in *Transactions on Large-Scale Data-and Knowledge-Centered Systems XVIII*. Springer, 2015, pp. 180-205.
43. Yu Y., Ma H., Zhang M.: A hybrid GP-tabu approach to QoSaware data intensive web service composition, in *Proc. Asia-Pacific Conf. Simulated Evol. Learn.*: Springer, Dec. 2014, pp. 106-118.
44. Wang A., Ma H., Zhang M.: Genetic programming with greedy search for web service composition, in *Database and Expert Systems Applications*, vol. 8056, pp. 9-17.
45. Yang F.C., et al.: An improved particle swarm optimization algorithm for QoS-aware web service selection in service oriented communication, *Int. J. Comput. Intell. Syst.*, 2010, 4, (s), pp. 18–30
46. Wang S.G., Sun Q.B., Zou H., et al.: Particle swarm optimization with skyline operator for fast cloud-based web service composition, *Mobile Netw. Appl.*, 2013, 18, pp. 116–121
47. Fan X.Q., Jiang C.J., Fang X.W., et al.: Dynamic web service selection based on discrete PSO, *J. Comput. Res. Dev.*, 2010, 47, (1), pp. 147–156
48. Wang L., He Y.X.: Web service composition based on QoS with chaos particle swarm optimization. 6th Int. Conf. on Wireless Communications Networking and Mobile Computing, 2010, pp. 1–4
49. Li W.F., Zhong Y., Wang X., et al.: Resource virtualization and service selection in cloud logistics, *J. Netw. Comput. Appl.*, 2013, 36, pp. 1696–1704
50. Li S.Z., Shen P., Yang S.X.: A grouping particle swarm optimization algorithm for web service selection based on user preference. 2011 IEEE Int. Conf. on Computer Science and Automation Engineering, Shanghai, China, 2011, v3, pp. 427–431
51. Ludwig S.A.: Applying particle swarm optimization to quality-of-service driven web service composition. IEEE 26th Int. Conf. on Advanced Information Networking and Applications (AINA), Fukuoka, Japan, 2012, pp. 613–620
52. Cao J.X., Sun X.S., Zheng X., et al.: Efficient multi-objective services selection algorithm based on particle swarm optimization. IEEE Asia-Pacific Services Computing Conf., 2010, pp. 603–608
53. Yu W., Li S., Tang X., et al.: An efficient top-k ranking method for service selection based on ε-

ADMOPSO algorithm, *Neural Comput. Appl.*, 2018, 31, pp. 77–92
54. Guha T., Ludwig S.A.: Comparison of service selection algorithms for grid services: multiple objective PSO and constraint satisfaction based service selection. 20th IEEE Int. Conf. on Tools with Artificial Intelligence, 2008, pp. 172–179
55. Fan X.Q. A decision-making method for personalized composite service, *Expert Syst. Appl.*, 2013, 40, pp. 5804–5810
56. Tao F., Zhao D.M., Hu Y.F., et al.: Correlation-aware resource service composition and optimal-selection in manufacturing grid, *Eur. J. Oper. Res.*, 2010, v. 201, pp. 129–143
57. Xu X., Rong H., et al.: Predatory search-based chaos turbo particle swarm optimisation (PS-CTPSO): A new PSO algorithm for web service combination probems, *Future Gener. Comput. Syst.*, v. 89, pp. 375–386, 2018.
58. Liu Y., Miao H., Li Z., et al.: QoS-aware web services composition based on HQPSO algorithm, in *Proc. 1st ACIS/JNU Int. Conf.Comput., Netw., Syst. Ind. Eng.*, May 2011, pp. 400–405.
59. Yin H., Zhang C., et al.: A hybrid multiobjective discrete PSO algorithm for a SLA-aware service composition problem, *Math. Problems Eng.*, vol. 4, 2014, pp.1-14
60. Wang S., Sun Q., Zou H., et al.: Particle swarm optimization with skyline operator for fast cloud-based web service composition, *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 116-121, 2013.
61. Hossain M. S., Moniruzzaman M, et al.: Big data-driven service composition using parallel clustered PSO in mobile environment, *IEEE Trans. Services Comput.*, vol. 9, no. 5, pp. 806-817, Sep. 2016.
62. Chifu V. R., Pop C. B., et al.: Web service composition technique based on a service graph and PSO, in *Proc. IEEE 6th Int. Conf. Intell.Comput. Commun. Process.*, 2010, pp. 265-272.
63. Gharbi M., Mezni H.: Towards big services composition, *Int. J.Web Grid Services*, vol. 16, no. 4, pp. 393–421, 2020.
64. Zhao X., Huang P., et al.: A hybrid clonal selection algorithm for quality of service-aware web service selection problem, *Int. J. Innov. Comput. Inf. Control*, vol. 8, no. 12, pp. 8527–8544, 2012.
65. Fekih H., Mtibaa S., Bouamama S.: An efficient user-centric web service composition based on harmony particle swarm optimization, *Int. J. Web Services Res.*, vol. 16, no. 1, pp. 1–21, Jan. 2019.
66. da Silva A. S., Mei Y., Ma H., et al.: Particle swarm optimization with sequence-like indirect representation for web service composition, in Evolutionary Computation in Combinatorial Optimization, vol. 9595, Chicano F., Hu B., García-Sánchez P., Eds. Cham, Switzerland: Springer, 2016, 8-14.
67. Zhao X., Song B., et al.: An improved discrete immune optimization algorithm based on PSO for QoS-driven web service composition, *Appl. Soft Comput.*, vol. 12, no. 8, pp. 2208–2216, 2012.
68. Haytamy S., Omara F.: A deep learning based framework for optimizing cloud consumer QoS-based service composition, *Computing*, vol. 102, no. 5, pp. 1117–1137, May 2020.
69. Hosseinzadeh M., Tho Q., Ali S., et al.: A hybrid service selection and composition model for cloud-edge computing in the Internet of Things, *IEEE Access*, vol. 8, pp. 85939-85949, 2020.
70. Sun Q., Wang S., Yang F.: Quick service selection approach based on particle swarm optimization. IEEE Fifth Int. Conf. on Bio-Inspired Computing: Theories and Applications, 2010, pp. 278–284
71. Zhao X.C., Song B.Q., et al.: An improved discrete immune optimization algorithm based on PSO for QoS-driven web service composition, *Appl. Soft Comput.*, 2012, 12, (8), pp. 2208–2216
72. Fan X.Q., Fang X.W., Jiang C.J.: Research on web service selection based on cooperative evolution, *Expert Syst. Appl.*, 2011, 38, pp. 9736–9743
73. Yu C.X., Wang G.: A multi-agent based architecture for web service selection in E-business. Eighth IEEE Int. Conf. on e-Business Engineering, Beijing, China, 2011, pp. 245–250
74. Fethallah H., Chikh M.A., Mohammed M., et al.: Qos-aware service selection based on swarm particle optimization. Int. Conf. on Information Technology and e-Services, Sousse, 2012, pp. 1–6
75. Zheng X.: Ant colony intelligence based solution for grid services selection. 7th World Congress on Intelligent Control and Automation, 2008, pp. 2512–2517
76. Wang L.J., Shen J., et al.: Towards minimizing cost for composite data-intensive services. Proc. of the 2013 IEEE 17th Int. Conf. on Computer Supported Cooperative Work in Design, 2013, pp. 293–298
77. Hossain M.S., Hossain S.K.A., Alamri A., *et al.*: Ant-based services election framework for a smart home monitoring environment, *Multimed. Tools. Appl.*, 2013, 67, pp. 433–453
78. Pop C.B., Chifu V.R., Salomie I., et al.: Ant-inspired technique for automatic web service composition and selection. 12th Int. Symp. on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania, 2010, pp. 449–455
79. Wang R., Ma L., Chen Y.P.: The application of ant colony algorithm in web service selection. 2010 Int. Conf. on Computational Intelligence and Software Engineering (CiSE), Wuhan, China, 2010, pp. 1–4.
80. Yu Q., Chen L., Li B. Ant colony optimization applied to web service compositions in cloud computing, Computers & Electrical Engineering Vol. 41, 2015, pp. 18–27
81. Fang Q., Peng X., Liu Q., et al.: A global QoS optimizing web service selection algorithm based on MOACO for dynamic web service composition. Int. Forum on Information Technology and Application, Chengdu, China, 2009, v.1, pp. 37–42
82. Zhang W., Chang C.K., Feng T.M., et al.: Qos-based dynamic web service composition with ant colony optimization. IEEE 34th Annual Computer Software and Applications Conf., Seoul, Korea, 2010, pp. 493–502
83. Dahan F., Hindi K., Ghoneim A., et al.: An Enhanced Ant Colony Optimization Based Algorithm

to Solve QoS-Aware Web Service Composition, IEEE Access ,Vol.9, pp. 34098 – 4111, 2021
84. Yang Z., Shang C., Liu Q., et al.: A dynamic web services composition algorithm based on the combination of ant colony algorithm and genetic algorithm, *J. Comput. Inf. Syst.*, vol. 6, no. 8, pp. 2617–2622, 2010.
85. Yang Y., Yang B., Wang S., et al.: A dynamic ant-colony genetic algorithm for cloud service composition optimization, *Int. J. Adv. Manuf. Technol.*, vol. 102, (1-4), pp. 355–368, 2019.
86. Liu Z.-Z., Wang Z.-J., Zhou X.-F, et al.: A new algorithm for QoS-aware composite web services selection, in *Proc. 2nd Int. Workshop Intell. Syst. Appl.*, May 2010, pp. 1–4.
87. Alayed H., Dahan F., Alfakih T., et al.: Enhancement of ant colony optimization for QoS-aware web service selection, *IEEE Access*, vol. 7, pp. 97041–97051, 2019.
88. Yang Z.K., Shang C.W., Liu Q.T., et al.: A dynamic web services composition algorithm based on the combination of ant colony algorithm and genetic algorithm, *J. Comput. Inf. Syst.*, 2010, 6, (8), pp. 2617–2622
89. Yang Y., Yang B., Wang S., et al.: A dynamic ant-colony genetic algorithm for cloud service composition optimization, *Int. J. Adv. Manuf. Technol.*, 2019,102, (1–4), pp. 355–368
90. Liu Z., Wang Z., Zhou X., *et al*.: A new algorithm for QoS-aware composite web services selection. 2nd Int. Workshop on Intelligent Systems and Applications (ISA), Kyiv, 2010, pp. 1–4
91. Bei L., Wenlin L., Xin S., et al.: An improved ACO based service composition algorithm in multi-cloud networks. *Journal of Cloud Computing (2024)* Vol.13, №1, *pp.1-12*
92. Jiang B., Qin Y., Yang Y., *et al*.: Web Service Composition Optimization with the Improved Fireworks Algorithm. Mobile Information Systems. Volume 2022, pp.1-13
93. Peng S., Guo T.: Multi-Objective Service Composition Using Enhanced Multi-Objective Differential Evolution Algorithm. Computational Intelligence and Neuroscience. 2023(1), pp.1-10.
94. Jafarpour N., Khayyambashi M.: A new approach for QoS-aware web service composition based on harmony search algorithm. 11th IEEE Int. Symp. on Web Systems Evolution, 2009, pp. 75–78
95. Ghobaei-Arani M., Rahmanian A., *et al*.: CSA-WSC: cuckoo search algorithm for web service composition in cloud environments, *Soft Comput.*, 2018, 22, (24), pp. 8353–8378
96. Dahan F. An Improved Whale Optimization Algorithm for Web Service Composition: Axioms 2022, 11, 725. pp. 1-14
97. Gavvala S., Jatoth C., *et al*.: Qos-aware cloud service composition using eagle strategy, *Future Gener. Comput. Syst.*, 2019, 90, pp. 273–290
98. Dahan F. Neighborhood Search Based Improved Bat Algorithm for Web Service Composition. Computer Systems Science & Engineering CSSE, 2023, vol.45, no.2 pp.1343-1356
99. Dahan F., Hindi K.E.: Enhanced artificial bee colony algorithm for QoS-aware web service selection problem, Computing, 2017, 99, (5), pp. 507–517
100. Pop C. B., Chifu V. R., Salomie I., et al.: Cuckoo-inspired hybrid algorithm for selecting the optimal web service composition, in *Proc. IEEE 7th Int. Conf. Intell. Comput. Commun. Process.*, Aug. 2011, pp. 33-40.
101. Chifu V. R, Pop C. B., Salomie I., et al.: Hybrid honey bees mating optimization algorithm for identifying the near-optimal solution in web service composition, *Comput. Informat.*, vol. 36, no. 5, pp. 1143-1172, 2017.
102. Yang H., Xue F., Zhu H., et al.: Web service composition optimization based on adaptive mutant beetle swarm, *J. Phys., Conf. Ser.*, vol. 1651, no. 1, Nov. 2020, pp.347-501
103. Dahan F., Alwabel A.: Artificial Bee Colony with Cuckoo Search for Solving Service Composition. Intelligent Automation & Soft Computing IASC, 2023, vol.35, no.3 pp.3385-3402
104. Ahanger T. A., Dahan F.: Hybridizing Artificial Bee Colony with Bat Algorithm for Web Service Composition. Computer Systems Science and Engineering CSSE, 2023, vol.46, no.2, pp.2429-2445
105. Peng S., Wang H., Yu Q.: Multi-clusters adaptive brain storm optimization algorithm for QoS-aware service composition, *IEEE Access*, vol. 8, pp. 48822-48835, 2020.

Одержано: 14.07.2025
Внутрішня рецензія оримана: 07.08.2025
Зовнішня рецензія оримана: 15.08.2025

Про автора:

Мороз Григорій Борисович,
кандидат технічних наук,
с.н.с., провідний науковий співробітник.
https://orcid.org/0000-0001-8666-9503

Місце роботи автора:

Інститут програмних систем
НАН України

УДК 004.4:004.942 https://doi.org/10.15407/pp2025.03.019

Є.М. Дерев'янко, В.Л. Шевченко

ВИЗНАЧЕННЯ ОПТИМАЛЬНОГО НАБОРУ МЕТОДІВ ІТЕРАЦІЙНОГО ПОКРАЩЕННЯ ОПОРНОГО МАРШРУТУ

В статті розв'язується задача комівояжера для планування польоту безпілотного літального апарату в умовах усунення наслідків надзвичайної ситуації. Розглянутий підхід використовує комбінацію методу найближчої точки для створення опорного рішення і низки методів локальних варіацій точок маршруту для покращення опорного рішення. В сценарій покращення опорного рішення включені методи: 1. 1PM - переміщення 1 точки, 2. 2PE - обміну місцями 2 точок і 3. DC -усунення перехрещення відрізків маршруту. У разі поліпшення опорного рішення кращий результат дає комбінація декількох методів. Якість покращення маршруту залежить від виду маршруту та від сценарію (набору і послідовності методів, включених у сценарій). Результати аналізу різних сценаріїв використання методів покращення опорного рішення узагальнені у вигляді звичайних графіків і теплових діаграм. Побудовані карти маршрутів для різних сценаріїв покращення опорного рішення. Найбільш результативними комбінаціями методів у складі сценаріїв виявилися 1-3, 3-1, 1-1. Найгірші комбінації: 2-2 та повні повтори інших методів 1-1-1, 2-2-2, 3-3-3. Величина виграшу щодо покращення якості маршруту змінювалась у діапазоні від 1 до 28%, в більшості випадків від 6 до 19%. Це потребувало від 2 до 24 ітерацій, в більшості випадків від 20 до 24 ітерацій. Як вартість маршруту були розглянуті Евклідова відстань, коефіцієнт небезпеки та Евклідова відстань з мультиплікатором у вигляді логістичної функції від коефіцієнту небезпеки. Метод розрахований на обмежені обчислювальні можливості звичайних бізнес-комп'ютерів. За умови ітераційного уточнення рішення методичний підхід дозволяє в реальному масштабі часу контролювати обчислювальну процедуру і завершувати її або по досягненні заданої точності, або у разі вичерпання часу. Модель створена алгоритмічною мовою Матлаб.

Ключові слова: задача комівояжера, маршрут, вартість маршруту, опорне рішення, обчислювальна процедура, модель, оптимізація, ітераційний метод.

Ye. Derevianko, V. Shevchenko

DETERMINATION OF THE OPTIMAL SET OF METHODS FOR ITERATIVE IMPROVEMENT OF THE REFERENCE ROUTE

The article solves the problem of a traveling salesman for planning a flight of an unmanned aerial vehicle in the conditions of eliminating the consequences of an emergency. An approach is considered that uses a combination of the nearest point method to create a reference solution and a number of methods of local variations of route points to improve the reference solution. The following methods are included in the reference solution improvement scenario: 1. 1PM - moving 1 point, 2. 2PE - exchanging places of 2 points and 3. DC - eliminating intersections of route segments. When improving the reference solution, the best result is obtained by combining several methods. The quality of route improvement depends on the type of route and the scenario (the set and sequence of methods included in the scenario). The results of the analysis of different scenarios of using reference solution improvement methods are summarized in the form of ordinary graphs and heat diagrams. Route maps are constructed for different scenarios of improving the reference solution. The most effective combinations of methods in the scenarios were 1-3, 3-1, 1-1. The worst combinations: 2-2 and complete repetitions of other methods 1-1-1, 2-2-2, 3-3-3. The magnitude of the gain in improving the quality of the route varied in the range from 1 to 28%, in most cases from 6 to 19%. This required from 2 to 24 iterations, in most cases from 20 to 24 iterations. The Euclidean distance, the hazard coefficient and the Euclidean distance with a multiplier in the form of a logistic function of the hazard coefficient were considered as the cost of the route. The method is designed for the limited computational capabilities of conventional business computers. When iteratively refining the solution, the methodological approach allows you to control the computational procedure in real time and complete it either upon reaching the specified accuracy or when time runs out. The model is created in the algorithmic language Matlab.

Keywords: traveling salesman problem, route, route cost, reference solution, computational procedure, model, optimization, iterative method.

Вступ

В умовах надзвичайних ситуацій для збереження життя людей широко застосовують безпілотні літальні апарати (БПЛА). БПЛА доставляють у небезпечні зони рятувальні засоби, воду, їжу тощо. БПЛА під час польоту може відвідувати декілька точок маршруту з метою моніторингу та доставки вантажів. Ресурс польотного часу є обмеженим. Виникає задача прокладання найкращого маршруту з урахуванням дальності, перешкод, небезпек та інших факторів. Задача побудови найкращого маршруту з урахуванням характеристик (вартості) окремих ділянок маршруту є загальновідомою задачею комівояжера SMP (Sales Man Problem). В цій задачі комівояжер повинен побувати у всіх заданих точках і повернутися в точку старту. Маршрут має бути прокладений так, щоб його вартість була найменшою. Під час надзвичайних ситуацій умови міняються динамічно. Прорахувати наперед усі маршрути неможливо. Потрібно мати змогу оперативного розрахунку маршрутів на звичайних бізнес-комп'ютерах на місці події. Таким чином, актуальною є задача оперативного розрахунку оптимального маршруту безпілотного літального апарату за допомогою звичайних обчислювальних засобів.

1. Аналіз існуючих досліджень

Загальна проблематика використання БПЛА розглянута в [1]. Але ця робота присвячена загальному аналізу стану та перспектив розвитку БПЛА. Питання побудови моделей оптимальних маршрутів розглянуті не були. Вперше задача комівояжера була розв'язана досить давно. Для невеликих розмірностей задачі популярним є метод повного прямого перебору або, як його ще називають, метод грубої сили (Brute Force) [2, 3]. На жаль, метод споживає багато обчислювальних ресурсів, має обчислювальну складність (n-1)! і ефективно працює лише у випадку, якщо кількості точок маршруту не більше 10. Розвитком методів повного прямого перебору є методи впорядкованого перебору, наприклад, метод дискретного динамічного програмування (алгоритм Хелд-Карпа) [4, 5], з обчислювальною складністю n^2 * 2^n. Недолік – високе споживання пам'яті і обмеження кількості точок у маршруті – до 25. Методи цілочисельного лінійного програмування мають гарну математичну формалізацію. Але вимагають стільки ж або ще більше обчислювальних ресурсів. Ефективність методів зростає за обмежень, які ведуть до зниження порядку моделі (branch-and-bound метод) [6].

Як вже було сказано вище, рішення має бути отримане оперативно. В такій ситуації компромісною поступкою може бути точність рішення. Дуже добре для практичної справи, якщо максимально швидко отримати якесь опорне (наближене) рішення, яке потім можна покращувати в рамках доступних обчислювальних та часових ресурсів. Швидкими є жадібні алгоритми [6-8]. Недолік - низька точність. Похибка може сягати до 25-45%. Подальше покращення опорного рішення можна виконувати за допомогою генетичних, мурашиних алгоритмів, алгоритмів відпалу [7, 8]. Недолік - складність отримання готових рішень на кожній ітерації. Потрібний час на налаштування процесу еволюції. Найбільш сучасним вважається використання нейронних мереж [8-10]. Недолік –ненаочність процесу оптимізації.

В [11] маршрут будували за допомогою жадібного алгоритму (Nearest Point Method - Метод найближчої точки). Низька якість результату стала розплатою за оперативність. В [12, 13] так само знаходиться спочатку опорне рішення. Але потім це рішення уточняється за допомогою локальних варіацій маршруту різними методами (1PM - переміщення 1 точки, 2PE - обмін місцями 1 точок, DC - усунення перехрещень відрізків). Із цих методів локальних варіацій формувався стек послідовних методів (сценарій), який повторювався декілька разів доти, поки покращення рішень припинялося. Недолік – евристичність сценарію без обґрунтування.

Недоліки розглянутих методів:

1. Висока обчислювальна складність обмежує кількість точок маршруту.

2. Складність використання в польових умовах надзвичайної ситуації на звичайних комп'ютерах.

3. Ненаочність проміжних результатів пошуку рішення і неможливість штатного припинення ітераційної процедури покращення рішення за умови вичерпання часу.

4. Евристичність сценаріїв покращення опорних рішень.

Раціональним шляхом подолання даних недоліків є створення опорного рішення (перевага - швидкість) та подальше його уточнення за допомогою набору методів локальних варіацій (перевага - наочність та можливість зупинки ітерації без втрати проміжного рішення).

Метою статті є покращення якості маршрутів БПЛА за допомогою ітераційного покращення опорного рішення набором методів локальних варіацій, зміст якого буде обґрунтовуватися на основі статистики успішності рішень.

2. Побудова опорного маршруту

Опорний маршрут створюємо за допомогою жадібного алгоритму методу найближчої точки. Далі опорний маршрут уточнюється за допомогою методів локальних варіацій. За критерій якості рішення використовується вартість маршруту, яка дорівнює сумі вартостей всіх ділянок маршруту

$$CostRoute = \sum_{\substack{RouteSet,\\ i=2,N}} \mathrm{Cost}(i, i-1),$$

де $RouteSet$ – множина номерів точок маршруту, які упорядковані за послідовністю проходженням маршруту, N – загальна кількість точок маршруту, i – номер точки маршруту.

Найчастіше як вартість проходження ділянки маршруту використовують Евклідову відстань

$$\mathrm{Cost}(i,\mathrm{j}) = \\ = D_e(i,\mathrm{j}) = \sqrt{\left(x_\mathrm{i} - x_j\right)^2 + \left(y_\mathrm{i} - y_j\right)^2},$$

де i,j - номери точок, x, y - координати точок.

У загальному випадку крім відстані для обрахунку вартості ділянки маршруту можна використати показники у вигляді втрат або витрат певних ресурсів: заряд батареї, паливо, амортизація БПЛА. Також не включається ймовірність певних подій: пошкодження або втрати БПЛА, невиконання завдання, втрати вантажу, інцидентів льотного та іншого характеру. Більшу кількість факторів, що були перелічені, можна узагальнити під єдиним показником – коефіцієнт небезпеки k_{Hazard}. Для ділянки маршруту між кожною парою точок $k_{Hazard}(i,\mathrm{j})$ має своє значення, яке для зручності оцінюємо числами в діапазоні від 0 до 100. На основі величини рівня небезпеки формується мультиплікатор відстані, який знаходимо за допомогою логістичної функції [14] (рис.1)

$$SL(k_{Hazard}(i,\mathrm{j})) = \\ = d + \frac{a-d}{1 + e^{-\frac{2}{T}(k_{Hazard}(i,\mathrm{j}) - s)}},$$

де a, d – асимптоти, T – постійна аргументу логістичної залежності, s – точка симетрії,

[d a T s] = [1 5 12 70].

Залежність вартості ділянки маршруту з урахуванням небезпеки має вигляд

$$\mathrm{Cost}_H(i,\mathrm{j}) = SL(k_{Hazard}(i,\mathrm{j}))\, D_e(i,\mathrm{j}).$$

Для пришвидшення моделювання матриці Cost, k_{Hazard}, Cost_H для всіх можливих пар точок маршруту (рис.2, 3) були розраховані до початку моделювання.

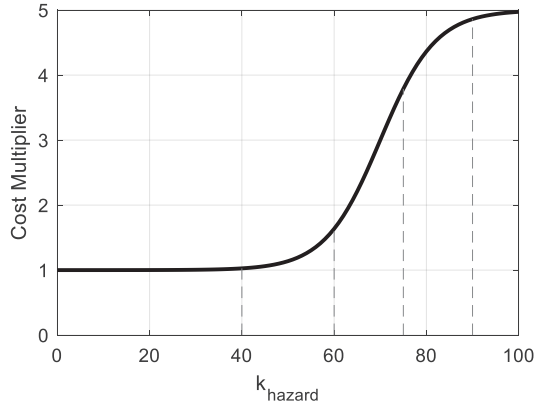


Рис. 1. Логістична залежність мультиплікатора вартості залежно від коефіцієнта небезпеки k_{Hazard}

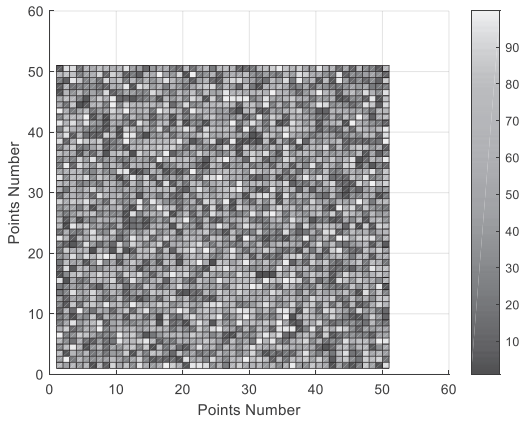


Рис. 2. Теплова карта масиву відстаней Cost

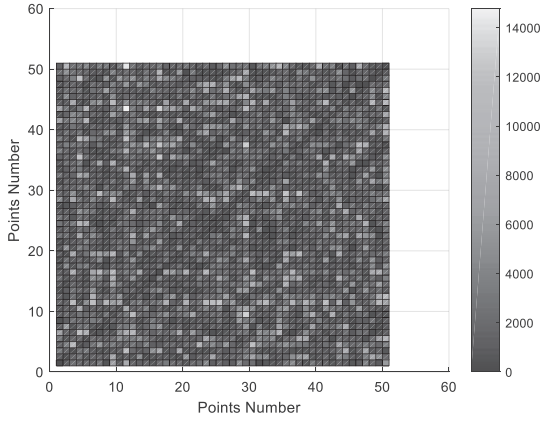


Рис. 3. Теплова карта масиву відстаней Cost_H з урахуванням коефіцієнту небезпеки k_{Hazard}

3. Локальні варіації маршруту

1PM (1 Point Moving) метод переміщення 1 точки полягає в тому, що кожну точку по черзі пробуємо перемістити на всі можливі нові позиції. Якщо вартість нового маршруту менша, то таке переміщення закріплюється. Якщо ні, то відкидається. Деякі покращення маршруту вимагають переміщення декількох точок або переміщення однієї точки декілька разів у комбінації з іншими точками. Тому процедура 1PM повторюється декілька разів. Але і в цьому випадку можуть бути такі розташування точок, що із них немає переходу до оптимального рішення лише за допомогою методу 1PM. Для виходу із такого тупика потрібно додатково використати якийсь інший метод. Комбінація методів мінімізує тупикові ситуації і підвищує якість рішень.

2PE (2 Point Exchange Method) метод обміну місцями двох точок маршруту. Аналогічно до попереднього методу аналізуються всі можливі варіанти пар обмінів. Недоліком є те, що оптимальність рішення залежить від порядку розгляду пар обмінів. Тому цей метод також повторюється декілька разів у циклі і комбінується з іншими методами. Метод 2PE, як і 1PM, поряд з покращеннями може утворювати перехрещення ділянок маршрутів, що є ознакою недостатньої якості маршруту. Для усунення перехрещень використовується окремий метод.

DC (Delete Crossing) метод усуває перехрещення ділянок маршруту від первинного стану (рис.4) до нового стану (рис.5). І перевага, і недолік методу в тому, що він шукає виключно перехрещення. Перевага в тому, що він це робить ефективно. Недолік в тому, що усунення одного перехрещення може утворювати інше. Крім того, метод не контролює інші негаразди маршруту. Тому цей метод повторюється у власному циклі, а також комбінується з іншими методами. Аналітичні викладки щодо виявлення та усунення перехрещень детально розглянуті в [12, 13].

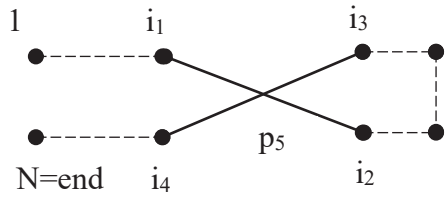


Рис. 4. Ділянки маршруту i1-i2 та i3-i4 до усунення перехрещення

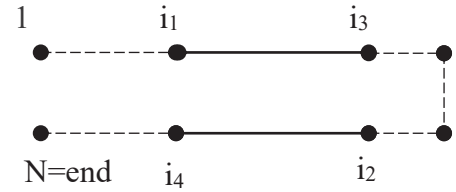


Рис. 5. Ділянки маршруту i1-i3 та i2-i4 після усунення перехрещення

Доведемо, що таке перетворення веде до зменшення довжини маршруту.

Довжини катетів менше довжини гіпотенузи (рис.6)

$$AD < AC, \quad DB < CB .$$

Додаємо праві і ліві частини нерівностей

$$AD + DB < AC + CB,$$
$$AB < AC + CB.$$

Це підтверджує, що довжина сторони трикутника менша за суму інших сторін.

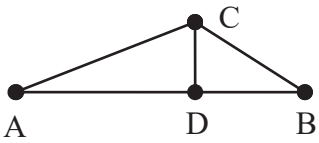


Рис. 6. Співвідношення сторін трикутника

Із цього випливає (рис.7), що

$$i_1 i_3 < i_1 p_5 + p_5 i_3,$$
$$i_4 i_2 < i_4 p_5 + p_5 i_2.$$

Додаємо праві та ліві частини нерівностей

$$i_1 i_3 + i_4 i_2 < (i_1 p_5 + p_5 i_2) + (i_4 p_5 + p_5 i_3),$$
$$i_1 i_3 + i_4 i_2 < i_1 i_2 + i_4 i_3.$$

Сума неперехрещених ділянок $i_1 i_3 + i_4 i_2$ менша за суму перехрещених ділянок $i_1 i_2 + i_4 i_3$.

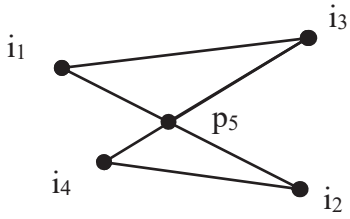


Рис. 7. Доведення виграшу від усунення перехрещення ділянок маршруту

Як і в попередніх методах, метод DC треба повторювати декілька разів і використовувати сумісно з іншими методами.

Загальний алгоритм ітераційного уточнення опорного маршруту.

1. Побудова опорного маршруту методом найближчого сусіда (NPM,Nearest Point Method).

2. Уточнення опорного маршруту.

2.1. Завантаження сценарію – комбінації методів локальних варіацій точок маршруту.

2.2. Цикл сценарію: повтор сценарію локальних варіацій у циклі.

2.2.1. Вибір наступного методу зі сценарію.

2.2.2. Цикл методу: повтор застосування поточного методу із сценарію.

2.2.3. Перевірка умови зупинки ітераційної процедури застосування методу (відсутність змін на 2 останніх ітераціях). Якщо умова виконана, то перехід на п.2.2.1.

2.2.4. Перевірка виконання заданої кількості ітерацій методу. Якщо умова не виконана, то повернення на етап 2.2.2.

2.2.5. Вихід із циклу методів.

2.3. Перевірка умови зупинки ітераційної процедури застосування сценарію (відсутність змін на 6 останніх ітераціях). Якщо умова не виконана, то повернення на етап 2.2.

2.4. Перевірка виконання заданої кількості ітерацій сценарію. Якщо умова не виконана, то повернення на етап 2.2.

2.5. Вихід із циклу сценарію.

3. Уточнення опорного маршруту завершене.

4. Пошук найкращої комбінації методів локальних варіацій у складі сценарію

Закодуємо методи локальних варіацій числами 1. 1PM, 2. 2PE, 3. DC. Сформуємо сценарії довжиною 3 методи із всіх можливих комбінацій трьох методів, розглянутих вище. Всього буде 3*3*3 = 27 варіанти. Сценарії були апробовані на вибірці на 30 різних наборах із 50 точок маршрутів, сформованих випадковим чином. Максимальне число повторів сценаріїв дорівнювало 4, максимальне число повторів методів у межах сценаріїв – 7. Окремі приклади результатів удосконалення опорних рішень представлені на рис. 8-10.

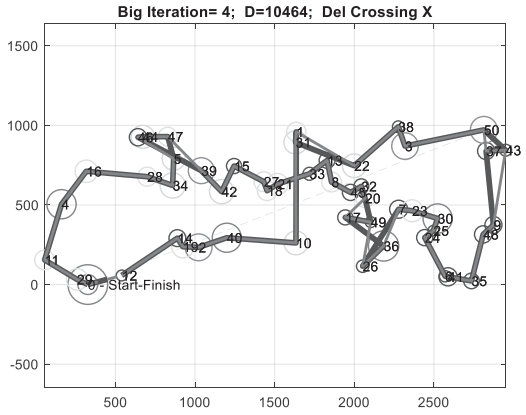


Рис. 8. Результати покращення маршруту для сценарію 14 (найгірша якість)

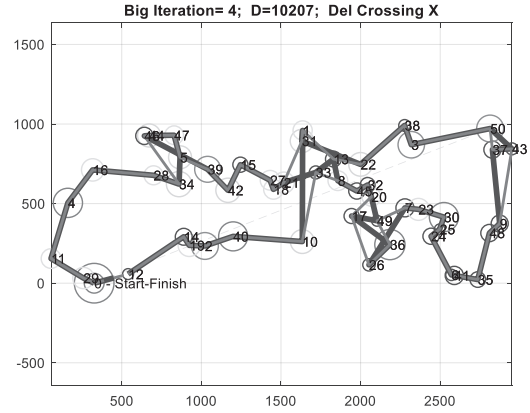


Рис. 9. Результати покращення маршруту для сценарію 6 (середня якість)

Тут подвійне червоне коло позначає точку старту (вона ж точка фінішу) маршруту. Опорний маршрут, створений за допомогою методу найближчого сусіда показаний синьою лінією. Покращений маршрут – червоною лінією. Сценарій 14 (методи [2 2 2]) надав одне із найгірших рішень (рис.8). Сценарій 6 (методи [1 2 3]) забезпечив середню якість (рис. 9). Сценарій 19 (методи [3 1 1]) надав найкращий результат (рис. 10).

Рис. 10. Результати покращення маршруту для сценарію 19 (найкращій результат)

Величина виграшу оцінювалась у відсотках відносно опорного рішення

$$CostMinimizationWin = \\ = \left(\frac{Cost_{NPM} - Cost_{Scenario}}{Cost_{NPM}}\right) 100\%,$$

де $Cost_{NPM}$ вартість опорного маршруту, $Cost_{Scenario}$ вартість маршруту, що був покращений за допомогою поточного сценарію. На рис.11 по вертикалі наведені номери (ліворуч) та зміст методів сценаріїв (праворуч). По горизонталі – виграш кожного сценарію. Одна лінія відповідає одному варіанту набору точок маршруту. Рис.11 показує, що сценарії мають свої особливості щодо спроможності покращувати опорне рішення незалежно від варіанту маршруту. Різні маршрути мають різний потенціал удосконалення. Рис.12 доповнений даними щодо кількості потрібних ітерацій.

Найбільша кількість пар сценарій-маршрут потребувала від 20 до 24 ітерацій. Усі пари - від 2 до 24. Величина виграшу в більшості випадків змінюється від 6 до 19%. Повний діапазон зміни величини виграшу від 1 до 28%. Лінійна регресія показує залежність виграшу від кількості ітерацій. Більш детально розподіл величини виграшу за конкретними номерами сценарію (ScenarioNumber) та номерами маршруту (TestNumber) показує теплова карта на рис.13.

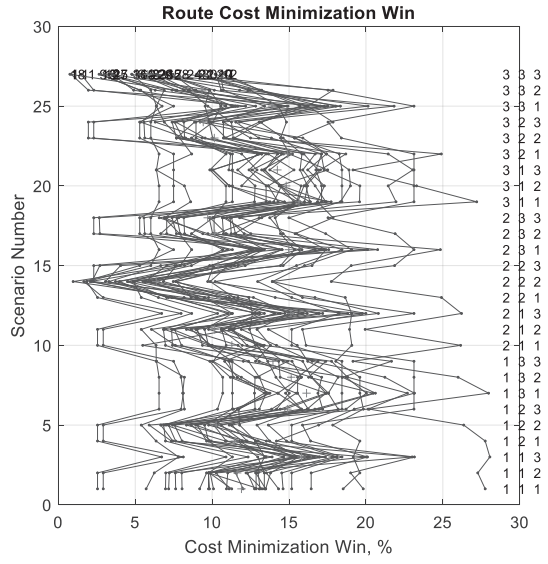


Рис. 11. Результативність покращення рішень для сценаріїв та маршрутів

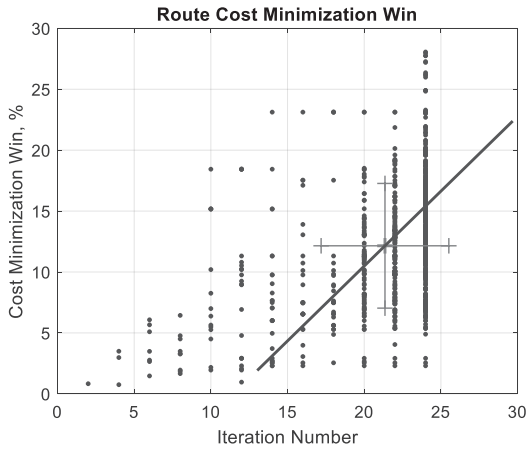


Рис. 12. Результативність покращення опорних рішень залежно від кількості ітерацій

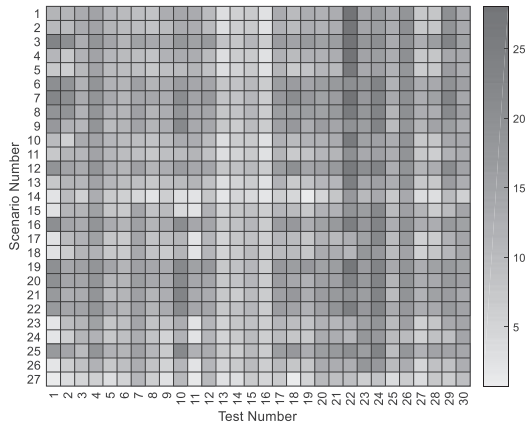


Рис. 13. Теплова карта виграшу залежно від номера сценарію (ScenarioNumber) та варіанта маршруту (TestNumber)

Аналіз отриманих даних виявив, що найкращі результати показували сценарії, в яких 1 (метод 1) перебуває на першій позиції. Трохи гірші, але теж непогані результати давали сценарії, в яких 1 перебуває на останній позиції. У будь-яких позиціях сценарію метод 1 працює найкраще, а метод 2 – найгірше. Водночас треба враховувати і зв'язки методів. Наприклад, метод 3 суттєво покращує свою ефективність у зв'язці з методом 1. Рейтинг пар методів на першій-другій позиції або на другій-третій позиції: найкращі в порядку рейтингу 1-3, 3-1, 1-1, найгірша пара 2-2. Повтор будь-яких методів 1-1-1, 2-2-2, 3-3-3 зменшує різноманіття, що суттєво погіршує результат.

В цілому метод 1 є домінуючим – підвищує ефективність сценарію. Дуже сильними є зв'язки 1-3, 3-1. Метод 2 майже завжди послаблює сценарій. Але він потрібний для швидкого усунення конкретної проблеми – перехрещення ділянок. І він може бути покращений через розташування поряд методу 1 або 3.

Висновки

В роботі розглядається розв'язання задачі комівояжера для планування польоту безпілотного літального апарату в умовах усунення наслідків надзвичайної ситуації.

Розглянутий підхід, що використовує комбінацію методу найближчої точки для створення опорного рішення і низки методів локальних варіацій точок маршруту для покращення опорного рішення. В сценарій покращення опорного рішення включені такі методи: 1. 1PM - переміщення 1 точки, 2. 2PE - обміну місцями 2 точок і 3. DC -усунення перехрещення відрізків маршруту.

Для поліпшення опорного рішення кращий результат дає комбінація декількох методів. Покращення якості маршруту залежить від виду маршруту та від сценарію, тобто від набору і послідовності методів, включених до сценарію.

Результати аналізу різних сценаріїв використання методів покращення опорного рішення були узагальнені у вигляді звичайних графіків і теплових діаграм. Були побудовані карти маршрутів для різних сценаріїв покращення опорного рішення.

Найбільш результативними комбінаціями методів у складі сценаріїв виявилися 1-3, 3-1, 1-1. Найгірші комбінації: 2-2 та повні повтори інших методів 1-1-1, 2-2-2, 3-3-3.

Величина виграшу щодо покращення якості маршруту змінювалась у діапазоні від 1 до 28%, в більшості випадків від 6 до 19%. Це потребувало від 2 до 24 ітерацій, в більшості випадків від 20 до 24 ітерацій.

Як вартість маршруту були розглянуті Евклідова відстань, коефіцієнт небезпеки та Евклідова відстань з мультиплікатором у вигляді логістичної функції від коефіцієнту небезпеки.

Метод розрахований на обмежені обчислювальні можливості звичайних біз-

нес-комп'ютерів. За умови ітераційного уточнення рішення методичний підхід дозволяє в реальному масштабі часу контролювати обчислювальну процедуру і завершувати її або після досягнення заданої точності, або у разі вичерпання часу.

Напрямами подальших досліджень є розширення методів уточнення і їх спрямування на конкретні вади маршрутів, які можуть бути діагностовані перед запуском методів.

Література

1. Мосов С.П. (2024) Роїння дронів військового призначення: реалії та перспективи. Збірник наукових праць Центру воєнно-стратегічних досліджень Національного університету оборони України, 2024, №1 (80), с.77-86. DOI: https://doi.org/10.33099/2304-2745/2024-1-80/77-86.
2. Dika Setyo Nugroho, Ahmad Ilham. Brute force algorithm application for solving traveling salesman problem (tsp) in semarang city tourist destinations. Jurnal Komputer dan Teknologi Informasi. Vol. 1, No. 2, Bulan 2023, pp. 79-86x. E-ISSN: 2986-7592, DOI: 10.26714/jkti.v3i1.13957. https://jurnal.unimus.ac.id/index.php/JKTI/article/view/16196/pdf
3. Gohil, A., Tayal, M., Sahu, T., and Sawalpurkar, V., "Travelling Salesman Problem: Parallel Implementations & Analysis", arXiv e-prints, Art. no. arXiv:2205.14352, 2022. doi:10.48550/arXiv.2205.14352. https://arxiv.org/abs/2205.14352
4. Parjadis, A., Bessiere, C., & Hebrard, E. (2024). Learning Lagrangian multipliers for the Travelling Salesman Problem. *Proceedings of the 30th International Conference on Principles and Practice of Constraint Programming (CP 2024)*, 22. https://drops.dagstuhl.de/storage/00lipics/lipics-vol307-cp2024/LIPIcs.CP.2024.22/LIPIcs.CP.2024.22.pdf?utm_source=chatgpt.com
5. Alanzi, E., Borchate, A., & co-authors. (2025). Solving the traveling salesman problem with machine learning: A review of recent advances and challenges. *Artificial Intelligence Review*. Advance online publication. https://doi.org/10.1007/s10462-025-11267-x https://www.researchgate.net/publication/392404863_Solving_the_traveling_salesman_problem_with_machine_learning_a_review_of_recent_advances_and_challenges?utm_source=chatgpt.com
6. Nataliya Boyko. An Overview of Existing Methods for Solving the Travelling Salesman Problem in Order to Find the Optimal Solution for Economic Problems. European scientific journal of Economic and Financial innovation. №1(7) (2021). DOI: http://doi.org/10.32750/2021-0108. https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://journal.eae.com.ua/index.php/journal/article/download/129/115/&ved=2ahUKEwjMnq-k0ZuNAxWyExAIHZqeG8QQFnoECB8QAQ&usg=AOvVaw0OzPk2F2kE3LoWlAP23rNG
7. Skakalina, O., & Kapiton, A. (2024). Comparative Analysis of Heuristic Algorithms for Solving the TSP. Системи управління, навігації та зв'язку. DOI: 10.26906/SUNZ.2024.2.144. https://www.researchgate.net/publication/380870329_COMPARATIVE_ANALYSIS_OF_THE_APPLICATION_OF_HEURISTIC_ALGORITHMS_FOR_SOLVING_THE_TSP_PROBLEM
8. Ivashchenko, H., Onyshchenko, O., Bondarenko, M., & Zdoryk, N. (2024). Methods for Solving the Traveling Salesman Problem Based on Computational Intelligence. Systems of Control and Navigation. DOI: 10.26906/SUNZ.2024.2.099
9. Yimeng Min, Yiwei Bai, Carla P Gomes. Unsupervised Learning for Solving the Travelling Salesman Problem. 37th Conference on Neural Information Processing Systems (NeurIPS 2023). 21 Sept 2023. https://openreview.net/forum?id=lAEc7aIW20&noteId=nfxOn45gNM
10. Шевченко В.Л. Імітаційне моделювання. Математичне моделювання процесів: Навч.посібник. / [В.І. Мірненко, В.Л.Шевченко, Д.С.Берестов, Р.М.Федоренко, А.В.Шевченко]; за ред. В.Л.Шевченка. – К.: КНУ ім.Т.Шевченка, 2020.– 164 с.
11. Igor Sinitsyn, Yevhen Derevianko, Stanislav Denysyuk, Volodymyr Shevchenko, Optimizing Reference Routes through Waypoint Sequence Variation in Emergency Events of Natural and Technological Origin, in: Proceedings of the Cybersecurity Providing in Information and Telecommunication Systems II (CPITS-II 2024), Kyiv, Ukraine, October 26, 2024 (online), CEUR Workshop Proceedings, ISSN 1613-0073, 2024, Vol-

3826, pp. 505-512, https://ceur-ws.org/Vol-3826/short20.pdf
12. Yevhen Derevianko, Nikita Krupa, Volodymyr Shevchenko, Viktor Shevchenko.Statistical Characteristics of the Reference Route Improvement Procedure Using a Stack of Iterative Methods // Proceedings of the Workshop on Intelligent Information Technologies (UkrPROG-IIT 2025), 15-th International Scientific and Practical Programming Conference, Kyiv, Ukraine, May 13-14, 2025, ISSN 1613-0073, 2025, Vol-4049, pp. 68-78, https://ceur-ws.org/Vol-4049/paper6.pdf
13. Volodymyr Shevchenko, Yevhen Derevianko, Oleksii Korniienko, Viktor Shevchenko. Improving the Reference Route Using a Stack of Iterative Local Optimization Methods // Proceedings of the 1st Workshop Software Engineering and Semantic Technologies (SEST 2025), Kyiv, Ukraine, May 13-14, 2025, ISSN 1613-0073, 2025, Vol-4053, pp. 218-226, https://ceur-ws.org/Vol-4053/paper13.pdf
14. Шевченко В.Л. Оптимізаційне моделювання в стратегічному плануванні.– К.: ЦВСД НУОУ, 2011. -283с.

References

1. Mosov S.P. (2024) Swarming of military drones: realities and prospects. Collection of scientific papers of the Center for Military and Strategic Studies of the National Defense University of Ukraine, 2024, №1 (80), c.77-86. DOI: https://doi.org/10.33099/2304-2745/2024-1-80/77-86.
2. Dika Setyo Nugroho, Ahmad Ilham. Brute force algorithm application for solving traveling salesman problem (tsp) in semarang city tourist destinations. Jurnal Komputer dan Teknologi Informasi. Vol. 1, No. 2, Bulan 2023, pp. 79-86x. E-ISSN: 2986-7592, DOI: 10.26714/jkti.v3i1.13957. https://jurnal.unimus.ac.id/index.php/JKTI/article/view/16196/pdf
3. Gohil, A., Tayal, M., Sahu, T., and Sawalpurkar, V., "Travelling Salesman Problem: Parallel Implementations & Analysis", arXiv e-prints, Art. no. arXiv:2205.14352, 2022. doi:10.48550/arXiv.2205.14352. https://arxiv.org/abs/2205.14352
4. Parjadis, A., Bessiere, C., & Hebrard, E. (2024). Learning Lagrangian multipliers for the Travelling Salesman Problem. Proceedings of the 30th International Conference on Principles and Practice of Constraint Programming (CP 2024), 22. https://drops.dagstuhl.de/storage/00lipics/lipics-vol307-cp2024/LIPIcs.CP.2024.22/LIPIcs.CP.2024.22.pdf?utm_source=chatgpt.com
5. Alanzi, E., Borchate, A., & co-authors. (2025). Solving the traveling salesman problem with machine learning: A review of recent advances and challenges. Artificial Intelligence Review. Advance online publication. https://doi.org/10.1007/s10462-025-11267-x https://www.researchgate.net/publication/392404863_Solving_the_traveling_salesman_problem_with_machine_learning_a_review_of_recent_advances_and_challenges?utm_source=chatgpt.com
6. Nataliya Boyko. An Overview of Existing Methods for Solving the Travelling Salesman Problem in Order to Find the Optimal Solution for Economic Problems. European scientific journal of Economic and Financial innovation. №1(7) (2021). DOI: http://doi.org/10.32750/2021-0108. https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://journal.eae.com.ua/index.php/journal/article/download/129/115/&ved=2ahUKEwjMnqk0ZuNAxWyExAIHZqeG8QQFnoECB8QAQ&usg=AOvVaw0OzPk2F2kE3LoWlAP23rNG
7. Skakalina, O., & Kapiton, A. (2024). Comparative Analysis of Heuristic Algorithms for Solving the TSP. Системи управління, навігації та зв'язку. DOI: 10.26906/SUNZ.2024.2.144. https://www.researchgate.net/publication/380870329_COMPARATIVE_ANALYSIS_OF_THE_APPLICATION_OF_HEURISTIC_ALGORITHMS_FOR_SOLVING_THE_TSP_PROBLEM
8. Ivashchenko, H., Onyshchenko, O., Bondarenko, M., & Zdoryk, N. (2024). Methods for Solving the Traveling Salesman Problem Based on Computational Intelligence. Systems of Control and Navigation. DOI: 10.26906/SUNZ.2024.2.099
9. Yimeng Min, Yiwei Bai, Carla P Gomes. Unsupervised Learning for Solving the Travelling Salesman Problem. 37th Conference on Neural Information Processing Systems (NeurIPS 2023). 21 Sept 2023. https://openreview.net/forum?id=lAEc7aIW20&noteId=nfxOn45gNM
10. Shevchenko V.L. Imitation modeling. Mathematical modeling of processes: Study guide. / [V.I. Mirnenko, V.L. Shevchenko, D.S.

Berestov, R.M. Fedorenko, A.V. Shevchenko]; under the editorship V.L. Shevchenko. – K.: KNU named after T. Shevchenko, 2020. – 164 p.
11. Igor Sinitsyn, Yevhen Derevianko, Stanislav Denysyuk, Volodymyr Shevchenko, Optimizing Reference Routes through Waypoint Sequence Variation in Emergency Events of Natural and Technological Origin, in: Proceedings of the Cybersecurity Providing in Information and Telecommunication Systems II (CPITS-II 2024), Kyiv, Ukraine, October 26, 2024 (online), CEUR Workshop Proceedings, ISSN 1613-0073, 2024, Vol-3826, pp. 505-512, https://ceur-ws.org/Vol-3826/short20.pdf
12. Yevhen Derevianko, Nikita Krupa, Volodymyr Shevchenko, Viktor Shevchenko.Statistical Characteristics of the Reference Route Improvement Procedure Using a Stack of Iterative Methods // Proceedings of the Workshop on Intelligent Information Technologies (UkrPROG-IIT 2025), 15-th International Scientific and Practical Programming Conference, Kyiv, Ukraine, May 13-14, 2025, ISSN 1613-0073, 2025, Vol-4049, pp. 68-78, https://ceur-ws.org/Vol-4049/paper6.pdf
13. Volodymyr Shevchenko, Yevhen Derevianko, Oleksii Korniienko, Viktor Shevchenko. Improving the Reference Route Using a Stack of Iterative Local Optimization Methods // Proceedings of the 1st Workshop Software Engineering and Semantic Technologies (SEST 2025), Kyiv, Ukraine, May 13-14, 2025, ISSN 1613-0073, 2025, Vol-4053, pp. 218-226, https://ceur-ws.org/Vol-4053/paper13.pdf
14. Shevchenko V.L. Optimization modeling in strategic planning.– K.: TsVSD NUOU, 2011. -283p.

Одержано: 03.10.2025
Внутрішня рецензія отримана: 07.10.2025
Зовнішня рецензія отримана: 07.10.2025

Про авторів:

[1]*Дерев'янко Євген Миколайович*,
аспірант.
http://orcid.org/0000-0003-2949-8896

[1]*Шевченко Віктор Леонідович*,
д.т.н., професор.
http://orcid.org/0000-0002-9457-7454.

Місце роботи авторів:

[1]Інститут програмних систем
НАН України,
тел. +38-044-522-62-42
E-mail: gii2014@ukr.net
Сайт: www.iss.nas.gov.ua

УДК 004.4'24 https://doi.org/10.15407/pp2025.03.029

В.О. Гайдукевич, А.Ю. Дорошенко

ЗАСТОСУВАННЯ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ ЕНЕРГОСПОЖИВАННЯ В СИСТЕМАХ РОЗУМНОГО ДОМУ

У статті досліджено застосування методів машинного навчання для прогнозування енергоспоживання в умовах функціонування систем розумного дому. Основу дослідження становить міжнародно відомий часовий набір даних PSML (Power System Machine Learning), який включає інформацію про споживання електроенергії, генерацію, балансування та прогнозування навантаження в умовах декарбонізованої енергетичної мережі. Набір даних PSML характеризується високою деталізацією та охоплює різні часові масштаби — від годинних до щоденних значень, що дозволяє оцінювати як короткострокові, так і середньострокові тенденції енергоспоживання.

У роботі проведено порівняльний аналіз класичних та сучасних методів машинного навчання, включно з регресією, класифікацією та кластеризацією, для прогнозування навантаження та виявлення шаблонів споживання електроенергії в побутовому середовищі. Особлива увага приділена оптимізації моделей для роботи з великими даними, обробці пропусків та аномалій, а також інтеграції прогнозів у системи автоматичного управління енергоресурсами розумного дому.

Ключові слова: машинне навчання, прогнозування, розумний дім, оптимізація енергоспоживання, PSML.

V.O. Haidukevych, A.Y. Doroshenko

APPLICATION OF MACHINE LEARNING MODELS TO PREDICT ENERGY CONSUMPTION IN SMART HOME SYSTEMS

The article investigates the application of machine learning methods to forecast energy consumption in the context of smart home systems. The research is based on the internationally renowned PSML (Power System Machine Learning) time series dataset, which includes information on electricity consumption, generation, balancing and load forecasting in the context of a decarbonized energy network. The PSML dataset is characterized by high detail and covers different time scales - from hourly to daily values, which allows assessing both short-term and medium-term trends in energy consumption.

The paper provides a comparative analysis of classical and modern machine learning methods, including regression, classification and clustering, for load forecasting and identifying patterns in electricity consumption in the domestic environment. Particular attention is paid to optimizing models for working with big data, processing gaps and anomalies, as well as integrating forecasts into automatic smart home energy management systems.

Keywords: machine learning, forecasting, smart home, energy optimization, PSML.

Вступ

Цифрова трансформація енергетичної галузі, зростання ролі децентралізованих джерел енергії та активна участь споживачів у процесах керування навантаженням актуалізують потребу у новітніх підходах до прогнозування енергоспоживання [1]. В умовах високої варіативності, сезонних коливань та метеозалежної генерації класичні статистичні методи дедалі частіше виявляються недостатньо ефективними [2]. У цьому контексті методи штучного інтелекту, зокрема, машинного навчання, відкривають нові можливості для побудови точних і гнучких моделей, здатних виявляти приховані нелінійні закономірності у часових рядах, обробляти великі обсяги даних та автоматизувати процеси прогнозування [3].

У статті представлено підхід до регресії, класифікації та кластеризації енергоспоживання на основі відкритого набору даних PSML [4], який охоплює багаторівневі часові ряди різної роздільності — від індивідуальних побутових споживачів до регіональних енергетичних систем. Особливу увагу зосереджено на моделюванні сценаріїв для розумного дому як типового елемента сучасної енергомережі. Такий

підхід дозволяє проаналізувати поведінку кінцевого споживача, протестувати ефективність алгоритмів у реалістичних умовах та оцінити їхню придатність для інтеграції в автоматизовані системи енергоменеджменту [5].

Метою дослідження є аналіз точності та генералізаційної спроможності сучасних моделей машинного навчання у вирішенні задач прогнозування навантаження, класифікації типових режимів роботи та виявлення схожих шаблонів споживання. У межах експериментальної частини було протестовано низку моделей, включаючи метод опорних векторів, алгоритми градієнтного підсилення, метод щільнісної кластеризації та інші, які оцінювалися за метриками точності, ефективності та інтерпретованості результатів [6].

Варто відзначити, що у статті [7] основний акцент зроблено на застосуванні методів машинного навчання для підвищення енергоефективності будівель шляхом оптимізації роботи інженерних систем. На відміну від цього, наше дослідження зосереджується на аналізі багаторівневих часових рядів споживання енергії, прогнозуванні навантаження та виявленні типових сценаріїв роботи побутових пристроїв у контексті розумного дому, що робить його більш орієнтованим на інтеграцію в системи енергетичного менеджменту з динамічними моделями прогнозування.

1. Системи розумного дому як контекст для прогнозування

Розумний дім — це інтегрований комплекс апаратних і програмних рішень, що забезпечують автоматизований контроль над освітленням, мікрокліматом, безпекою, мультимедійними пристроями та особливо енергоспоживанням у житловому просторі [3]. У центрі такої системи знаходиться інтелектуальний модуль керування, здатний отримувати дані з численних сенсорів, виконувати аналіз у реальному часі та ухвалювати оптимальні рішення для підвищення комфорту мешканців та ефективності використання ресурсів.

Одним із ключових напрямів розвитку розумного дому є прогнозування споживання електроенергії. Це завдання має низку стратегічно важливих переваг [8]:

- зменшення втрат енергії — точні прогнози дозволяють уникнути роботи обладнання у холостому режимі, оптимізувати цикли роботи техніки та знизити загальні витрати.
- автоматичне керування побутовими приладами — система може самостійно вмикати чи вимикати прилади, регулювати інтенсивність роботи кондиціонерів, нагрівачів та освітлення.
- адаптація до пікових навантажень у мережі — прогнозні моделі допомагають зміщувати споживання на години з меншими тарифами або меншими навантаженнями у мережі, що особливо важливо для енергетичних систем з обмеженими ресурсами.

На практиці це означає, що система розумного дому може:

- активувати обігрів або охолодження заздалегідь, з урахуванням прогнозу температури та графіку перебування мешканців;
- автоматично заряджати акумуляторні системи або електромобілі у період нічних тарифів;
- відключати непотрібні пристрої, коли мешканців немає вдома;
- інтегруватися з відновлюваними джерелами енергії, прогнозуючи потужність від сонячних панелей або вітрогенераторів.

Використання методів машинного навчання у цьому контексті дає змогу будувати адаптивні моделі, які враховують не лише часові патерни енергоспоживання, а й зовнішні фактори, зокрема, погодні умови, вологість, графік використання побутової техніки та навіть індивідуальні звички мешканців [9]. Крім того, прогнозування в розумному домі сприяє побудові мікромереж (microgrid) — локальних енергетичних систем, де баланс між генерацією та споживанням може підтримуватися автоматично. Це зменшує залежність від центральної мережі та підвищує енергетичну автономність житла [10].

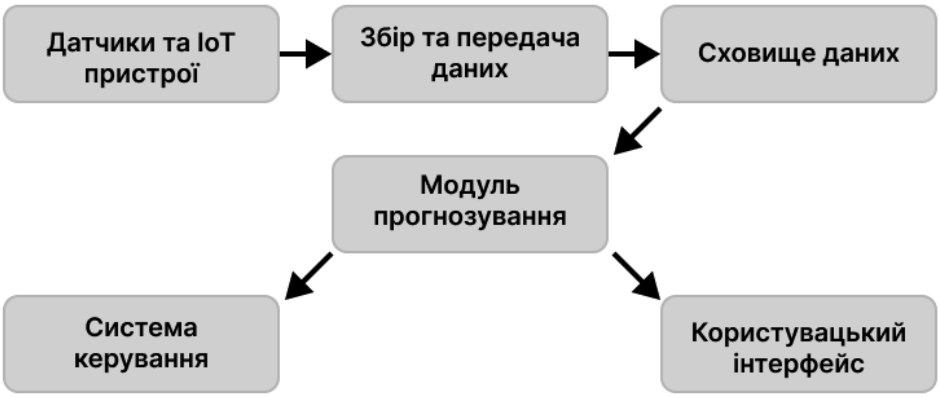


Рис. 1. Схема архітектури системи розумного дому з модулем прогнозування

2. Огляд і підготовка даних PSML

Набір PSML є спеціалізованою колекцією високодеталізованих часових рядів, які відображають динаміку побутового споживання електроенергії з інтервалом у 15 хвилин. Такі дані зазвичай надходять від сучасних розумних лічильників та систем енергомоніторингу, що інтегруються в інфраструктуру розумного дому. Цей набір містить як основні показники енергоспоживання, так і супутні фактори, що впливають на нього:

- timestamp — мітка часу вимірювання, яка дає змогу синхронізувати записи з іншими джерелами, наприклад прогнозами погоди або даними про генерацію електроенергії.
- house_id — унікальний ідентифікатор домогосподарства, що дозволяє відслідковувати індивідуальні або групові профілі.
- power_kwh— обсяг спожитої електроенергії (у кВт·год) за фіксований часовий інтервал; ключовий показник для побудови моделей прогнозування.
- temperature — температура зовнішнього середовища (°C), важлива для аналізу впливу кліматичних факторів.
- humidity — відносна вологість (%), параметр, що впливає на роботу кліматичних систем.
- day_of_week, hour, is_weekend — створені часові ознаки, що допомагають враховувати добові й тижневі патерни.

Завдяки своїй структурі, PSML є типовим прикладом IoT-збірника даних, що використовується для інтелектуального управління енергоспоживанням у приватних будинках та комерційних об’єктах. Він демонструє природні коливання навантаження, зумовлені часом доби, погодними умовами та поведінкою мешканців.

Використання PSML у аналітичних платформах відкриває широкі можливості для підвищення енергоефективності. На основі цих даних можна:

- прогнозувати рівень споживання з урахуванням сезонних і погодних змін;
- автоматично регулювати роботу побутових приладів, адаптуючи їх до пікових тарифів та знижуючи витрати;
- оптимізувати роботу домашніх акумуляторних систем, визначаючи найвигідніший час для заряджання й розряджання;
- виявляти аномалії — від потенційних несправностей обладнання до нераціонального використання ресурсів.

Таким чином, цей набір даних є не лише основою для побудови прогнозних моделей, а й служить базою для створення самоадаптивних алгоритмів управління енергетикою в розумних будинках.

Для забезпечення високої якості та надійності подальшого моделювання, дані PSML пройшли ретельну попередню обробку:

1. Заповнення пропусків — відновлення відсутніх значень за допомогою інтерполяції або медіан для уникнення викривлення трендів.
2. Розширення часових ознак — з часових міток згенеровано додаткові атрибути (година доби, день тижня, вихідний день).
3. Масштабування — нормалізація числових параметрів для стабільної роботи алгоритмів машинного навчання.
4. Виділення трендів і сезонності — застосування декомпозиції часових рядів для точнішого прогнозування.

timestamp	house_id	power_kwh	temperature	humidity	day_of_week	hour	is_weekend
01-01 00:00:00	1006	1.07	24.0	59.9	0	0	False
01-01 00:15:00	1003	1.08	19.3	76.1	0	0	False
01-01 00:30:00	1007	1.66	4.1	34.4	0	0	False
01-01 00:45:00	1004	2.72	-2.1	39.8	0	0	False
01-01 01:00:00	1006	2.27	15.5	32.3	0	1	False
01-01 01:15:00	1009	1.6	8.2	46.3	0	1	False
01-01 01:30:00	1002	3.14	-1.3	49.4	0	1	False
01-01 01:45:00	1006	0.87	9.9	43.6	0	1	False
01-01 02:00:00	1007	1.6	-4.0	71.4	0	2	False
01-01 02:15:00	1004	1.96	22.3	47.8	0	2	False

Рис. 2. Фрагмент датасету PSML

3. Методика дослідження

У межах цього дослідження ставиться задача прогнозування навантаження в електричних мережах на основі розумного дому. Метою є передбачення значення енергоспоживання у майбутні часові періоди. Такий тип задач є критично важливим для планування генерації, балансування навантаження, оптимізації роботи енергетичних систем і запобігання перевантаженням.

Завданням моделі є знаходження залежності між часовими характеристиками (день тижня, година, місяць тощо) та фактичним навантаженням.

У дослідженні застосовано кілька класів алгоритмів машинного навчання для вирішення задач прогнозування, класифікації та кластеризації побутового електроспоживання на прикладі даних PSML. Підхід базується на поєднанні класичних методів обробки часових рядів та сучасних алгоритмів, здатних виявляти складні нелінійні залежності.

Для аналізу використано PSML — відкритий набір даних, що містить показники навантаження енергоспоживання з часовою роздільністю 1 година. Набір включає дані за декілька років, що дозволяє моделі враховувати сезонні й добові коливання.

Перед моделюванням дані було нормалізовано та закодовано категоріальні змінні (наприклад, день тижня — one-hot encoding). Перед моделюванням виконано:

- нормалізацію числових ознак (MinMax) для приведення значень у діапазон [0,1] та уникнення домінування одних змінних над іншими;
- one-hot encoding для категоріальних змінних (dayofweek), щоб уникнути хибного порядкування днів;
- циклічне кодування (sin, cos) для годин та місяців, щоб зберегти їхню періодичність;
- розділення вибірки за часом (навчання на минулих даних, тестування на нових), що відповідає реальним умовам прогнозування.

Для побудови моделей машинного навчання необхідно було розділити дані на незалежні підмножини, щоб оцінка якості моделі була об’єктивною, а прогнозна здатність — прийнятною для нових даних.

У даному дослідженні застосовано наступну схему:

- навчальна вибірка — 70% даних - використовувалась для тренування моделей та підбору параметрів;
- тестова вибірка — 30% даних - призначена для остаточного тестування якості моделей на «невиданих» прикладах.

Валідаційна вибірка не виділялась окремо, оскільки її роль виконує внутрішнє розбиття у процесі крос-валідації та підбору гіперпараметрів, наприклад, із використанням алгоритму GridSearchCV.

Розбиття виконувалося за допомогою функції train_test_split() з бібліотеки scikit-learn із фіксацією параметра random_state=42 для забезпечення відтворюваності результатів у повторних експериментах.

4. Прогнозування параметрів споживання (регресія)

Для вирішення задач прогнозування та аналізу структури даних використано наступні моделі:

1. Support Vector Regressor (SVR)

Метод опорних векторів дозволяє будувати нелінійні моделі за рахунок використання ядрових функцій. Було протестовано Gaussian RBF kernel. Модель показала позитивні результати на короткострокових прогнозах. Використані гіперпараметри:

kernel = 'rbf' — радіальна базисна функція
C = 1000 — штраф за помилку
epsilon = 0.1 — ширина "труби" точності
gamma = 'scale' — автоматичне визначення параметра

2. Random Forest Regressor (RFR),

Метод Випадкового Лісу як ансамблевий метод дозволив врахувати взаємозв'язки між температурою, годиною доби та обсягом споживання. RF був обраний завдяки високій стійкості до перенавчання і здатності працювати з некорельованими ознаками. Використані гіперпараметри:

n_estimators = 200 — кількість дерев
max_depth = 15 — максимальна глибина дерева
min_samples_split = 5 — мінімальна кількість прикладів для розбиття
random_state = 42 — для стабільності результатів.

3. XGBoost Regressor

Алгоритм градієнтного підсилення над деревами рішень, який показав високу ефективність у задачах регресії на складних вибірках. Завдяки регуляризації він зменшує ризик перенавчання. Використані гіперпараметри:

n_estimators = 300 — кількість ітерацій
learning_rate = 0.05 — швидкість навчання
max_depth = 10 — максимальна глибина дерев
subsample = 0.8 — частка даних для побудови кожного дерева
colsample_bytree = 0.8 — частка ознак для кожного дерева

У додатку 1 наведено фрагмент програмного коду на Python, який реалізує прогнозування енергоспоживання на основі метеорологічних ознак з використанням моделі Random Forest.

Для оцінки точності прогнозування використано три основні метрики регресії:

- MAE (Mean Absolute Error) — середня абсолютна похибка;
- RMSE (Root Mean Squared Error) — корінь середньоквадратичної помилки;
- R^2 (коефіцієнт детермінації) — частка поясненої дисперсії;
- MAPE (Mean Absolute Percentage Error) — середня абсолютна відносна похибка, яка показує похибку у відсотках від фактичного значення.

Метрики обчислювались як для тестової, так і для крос-валідаційної вибірки.

Моделі SVR, RFR та XGBoost було протестовано на одній і тій самій вибірці. Результати представлено у таблиці 1:

Таблиця1.

Похибка регресії у прогнозуванні спожитої електроенергії (параметр power_kwh)

Модель	MAE (кВт)	RMSE (кВт)	R^2 (коеф. детермінації)	MAPE (%)
SVR	0.112	0.157	0.930	6.8
RFR	0.098	0.141	0.944	5.9
XGBoost Regressor	0.091	0.136	0.951	5.2

Як видно з результатів (табл.1), найкращі показники продемонструвала модель XGBoost Regressor. Вона забезпечила найнижчі значення MAE та RMSE, а також найвищий коефіцієнт детермінації ($R^2 = 0.951$). Додатково важливим показником є MAPE, який склав лише 5.2%, що свідчить про високу відносну точність прогнозу навіть за умови коливань у структурі споживання. За цільовий параметр прогнозування обрано споживання електроенергії (power_kwh). Порівняно з іншими моделями:

- Support Vector Regressor (SVR) показав позитивні результати на короткострокових прогнозах, проте мав вищу відносну похибку (MAPE 6.8%) у порівнянні з ансамблевими методами.
- Random Forest Regressor (RFR) продемонстрував стабільність та кращу точність від SVR за всіма метриками, зокрема, зменшив середню похибку до MAE = 0.098 кВт та MAPE = 5.9%.
- XGBoost Regressor перевершив обидві моделі, забезпечивши найбільш збалансоване поєднання високої точності та узагальнювальної здатності.

Тож результати експериментів підтвердили доцільність використання більш складних ансамблевих методів для задач прогнозування енергоспоживання. Завдяки здатності враховувати нелінійні залежності та ефективно працювати з великими обсягами даних, XGBoost може бути рекомендований як базова модель для практичних систем прогнозування у сфері «розумних мереж» та управління енергоспоживанням.

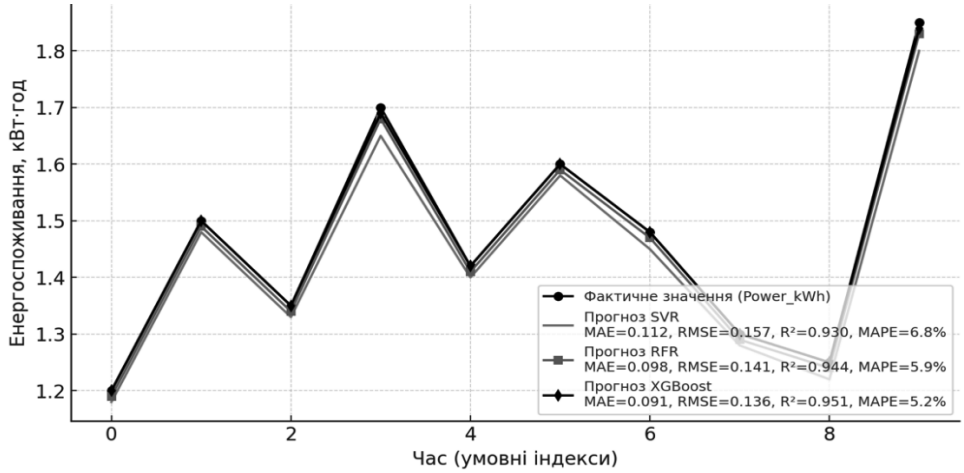


Рис. 3. Порівняння прогнозів моделей регресії (SVR, RFR, XGBoost)

5. Класифікація режимів роботи

У другому етапі дослідження вирішувалася задача класифікації режимів роботи енергоспоживання (device_mode), яка передбачала приналежності кожного відрізка часу до однієї з трьох категорій:

- низький режим (low) — відповідає періодам мінімального базового навантаження, зазвичай у нічний час або у випадку тривалої відсутності мешканців;
- середній режим (medium) — характеризує типовий денний рівень споживання у робочі дні, коли використовуються побутові прилади середньої потужності;
- піковий режим (peak) — відображає короткочасні стрибки навантаження, що виникають у вечірні години або у моменти одночасної роботи кількох енергоємних пристроїв (наприклад, кондиціонера, пральної машини та кухонної техніки).

Для розв'язання задачі було застосовано алгоритмічні підходи:

- Random Forest Classifier — ансамблевий метод на основі дерев рішень, стійкий до шуму та добре працює з некорельованими ознаками;

- SVM Classifier — метод опорних векторів із використанням RBF-ядра, здатний будувати нелінійні межі рішень для складних даних подальшого порівняння;
- XGBoost Classifier — алгоритм градієнтного підсилення над деревами рішень, що продемонстрував високу гнучкість у роботі з великими та частково зашумленими даними. Його застосування виявилося доцільним з огляду на складність розподілу навантаження та можливу наявність нелінійних залежностей у даних.

Навчання та тестування обох моделей здійснювалося на тих самих вибірках, що і в задачі регресії: 70% даних було використано для навчання, 30% — для тестування (параметр random_state=42 забезпечив відтворюваність результатів).

Якість класифікації оцінювалася за стандартними метриками: точність (Accuracy), точність передбачення (Precision), повнота (Recall) та F1-міра. Отримані результати подано у таблиці 2.

Таблиця 2.

Метрики класифікації режиму електроспоживання розумного дому

Модель	Accuracy	Precision	Recall	F1-score
Random Forest	0.894	0.901	0.889	0.894
SVM Classifier	0.881	0.887	0.873	0.879
XGBoost Classifier	0.912	0.918	0.905	0.911

Аналіз показує, що XGBoost Classifier знову перевершує інші моделі за всіма показниками. Водночас Random Forest показав кращі результати за метриками порівняно з SVM, що підтверджує його здатність ефективно працювати із зашумленими даними (Рис.4).

Зокрема, точність і повнота у XGBoost Classifier перевищили 91%, що підтверджує його придатність для впровадження у системах розумного дому. Це відкриває можливості для задач балансування енергоспоживання, попередження перевантажень електромережі та побудови адаптивних тарифних стратегій.

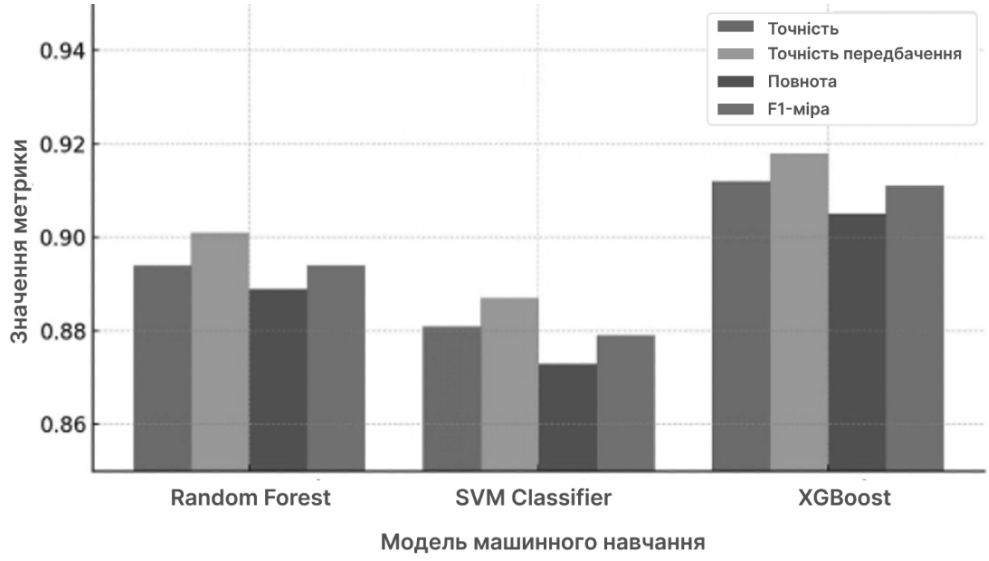


Рис. 4. Порівняння моделей класифікації режимів роботи

6. Кластеризація профілів енергоспоживання

На третьому етапі дослідження було проведено кластеризацію добових профілів енергоспоживання з метою виявлення груп днів із подібною структурою навантаження. Для цього вихідні дані PSML були агреговані до добового масштабу та нормалізовані, що дало можливість зосередитися

на формі профілю, а не на абсолютних величинах споживання.

Першим було застосовано алгоритм KMeans, який належить до класу методів розбиття даних на заздалегідь визначену кількість кластерів. Для вибору оптимального значення параметра k використовували метод «лікоть» та коефіцієнт силуету, що дозволило визначити три стійкі групи профілів. Перша група відповідала типовим будням із вираженими піками споживання у ранкові та вечірні години, друга – вихідним дням із більш рівномірним та згладженим навантаженням, а третя група відображала аномальні випадки, коли споживання електроенергії мало нетипово високі або низькі значення протягом доби. Таким чином KMeans дозволив сформувати структуровану типологію днів за їхніми характерними режимами роботи побутових пристроїв.

Паралельно було використано алгоритм DBSCAN, який, на відміну від KMeans, не потребує попереднього визначення кількості кластерів. Цей метод показав себе ефективним у пошуку щільних зон подібних профілів та автоматичного виявлення «шумових» точок, що не належали жодному кластеру. Як результат, DBSCAN виявив компактні групи днів із подібною поведінкою споживання, а також виділив окремі аномалії, які відображають нетипові сценарії використання електроенергії в системах розумного дому.

Порівняльний аналіз продемонстрував, що KMeans є більш придатним для опису повторюваних і структурованих шаблонів енергоспоживання, тоді як DBSCAN надає кращі результати у випадках пошуку рідкісних або відхилених від норми режимів. Відповідні результати візуалізовано на рисунку 5, де відображено типові кластери, отримані за допомогою обох методів. Для кожного кластера побудовано середній профіль споживання, який обчислюється як середнє значення добового споживання для всіх днів, що належать до кластера. Альтернативно, може використовуватися медіанний профіль, щоб зменшити вплив аномалій. Саме ці середні або медіанні графіки показані на рисунку 5, і вони ілюструють типову форму навантаження для кожного кластера.

Отримані кластери мають важливе прикладне значення. Вони можуть бути використані для персоналізації прогнозів залежно від типу дня, оптимізації графіків роботи енергоємних приладів, а також для розробки більш гнучких та справедливих тарифних стратегій, що враховують різні моделі споживання електроенергії. Таким чином, кластеризація дозволяє не лише виявити приховані закономірності у даних, а й забезпечує основу для практичного застосування в системах розумного дому та енергетичному менеджменті.

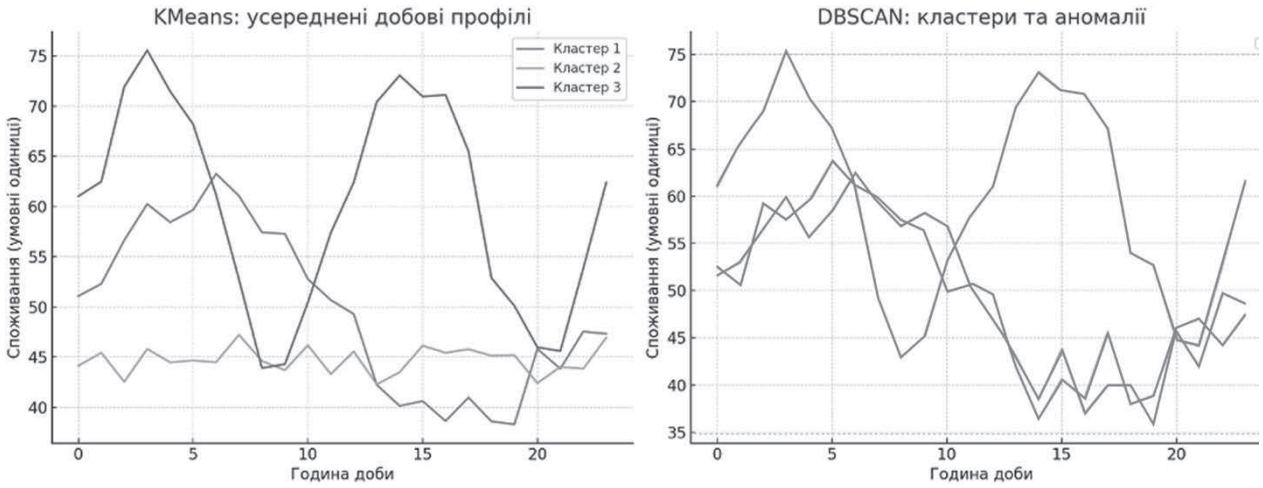


Рис. 5. Результати кластеризації профілів енергоспоживання (KMeans та DBSCAN)

Висновки

У статті досліджено можливості застосування методів машинного навчання для аналізу та прогнозування енергоспоживання в системах розумного дому на основі датасету PSML. Було реалізовано комплексний підхід, що включає три основні етапи: регресійне прогнозування, класифікацію режимів роботи та кластеризацію профілів споживання.

На етапі регресії порівнювались моделі SVR, RFR, XGBoost Regressor. Найкращі результати показав XGBoost Regressor, що забезпечив найнижчі похибки (MAE = 0.091, RMSE = 0.136, MAPE = 5.2%) та найвище значення коефіцієнта детермінації ($R^2 = 0.951$). Це свідчить про високу здатність моделі враховувати нелінійні залежності та узагальнювати дані, що робить її найбільш придатною для задач прогнозування енергоспоживання.

Етап класифікації дозволив виділити три основні режими роботи: низький, середній та піковий. Порівняння моделей показало, що XGBoost Classifier також перевершив Random Forest та SVM за всіма метриками, досягнувши точності понад 91%. Це підтверджує його ефективність для задач віднесення споживання до різних режимів (низький, середній, піковий), що є критично важливим для балансування навантаження та запобігання перевантаженням мережі.

Кластеризація добових профілів споживання із застосуванням KMeans та DBSCAN дозволила виділити типові будні, вихідні та аномальні дні. KMeans виявив повторювані шаблони, тоді як DBSCAN ефективно ідентифікував рідкісні або нетипові сценарії, що забезпечує можливість адаптивного планування роботи енергоємних пристроїв та розробки стратегій тарифікації.

Завдяки комплексному застосуванню методів регресії, класифікації та кластеризації досягнуто:

- точного прогнозування енергоспоживання;
- коректної ідентифікації режимів роботи пристроїв;
- виділення типових та аномальних сценаріїв споживання.

Узагальнюючи результати, можна зробити висновок, що поєднання різних підходів машинного навчання дозволяє не лише досягти високої точності прогнозування енергоспоживання, а й гнучко управляти режимами роботи пристроїв та виявляти приховані закономірності у даних. Найбільш ефективним алгоритмом у задачах як регресії, так і класифікації виявився XGBoost, що робить його перспективною базовою моделлю для впровадження у практичні системи розумного дому та розумних мереж.

References

1. Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and practice (2nd ed.). OTexts. https://otexts.com/fpp2/
2. Weron, R. (2014). Electricity price forecasting: A review of the state-of-the-art with a look into the future. International Journal of Forecasting, 30(4), 1030–1081.
3. Ryu, S., Noh, J., & Kim, H. (2017). Deep neural network based demand side short term load forecasting. Energies, 10(1).
4. Open source PSML dataset https://github.com/tamu-engineering-research/Open-source-power-dataset?tab=readme-ov-file
5. Sinitsyn, I. P., Shevchenko, V. L., Doroshenko, A. Yu., & Yatsenko, O. A. (2025). Research of software solutions for forecasting electricity production and consumption in Ukraine based on machine learning methods. Problems of programming and information technologies, 15(1), 45–58. https://pp.isofts.kiev.ua/ojs1/article/view/586
6. Python Software Foundation. (2024). Scikit-learn: Machine learning in Python. https://scikit-learn.org
7. Vyshnevskyy, O. V., & Zhuravchak, L. (2023). Machine Learning Methods to Increase the Energy Efficiency of Buildings. Computer Systems and Networks, 14, 189–209. https://www.researchgate.net/publication/378031111_Machine_Learning_Methods_to_Increase_the_Energy_Efficiency_of_Buildings
8. Li, W., & Wang, J. (2019). Electricity consumption forecasting using a hybrid model based on data preprocessing and long short-

term memory neural network. Energy, 169, 1099–1110.
9. Ahmed, R., & Khalid, M. (2020). A review on the selected applications of forecasting models in renewable energy systems. Renewable and Sustainable Energy Reviews, 124, 109792.
10. Khac, T. N., & Bui, T. V. (2021). Load forecasting using machine learning: A comparative study of neural networks and random forests. Energy Reports, 7, 394–403.

Додаток 1.

Фрагмент програмного коду для прогнозування енергоспоживання в системі розумний дім

```
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

# 1. Завантаження та попередня обробка даних
df = pd.read_csv("psml_data.csv", parse_dates=["datetime"])
df = df.dropna(subset=["load"])  # видалення рядків з відсутнім споживанням

# 2. Вибір ознак та цільової змінної

features = ["temperature", "humidity", "wind_speed"] # метео-параметри як ознаки
target = "load"  # споживання енергії

X = df[features]
y = df[target]

# 3. Розділення даних на тренувальні та тестові множини
X_train, X_test, y_train, y_test = train_test_split(
   X, y, test_size=0.2, random_state=42
)

# 4. Ініціалізація та навчання моделі Random Forest
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# 5. Прогнозування та оцінка точності моделі
y_pred = model.predict(X_test)
rmse = mean_squared_error(y_test, y_pred, squared=False)
print(f"RMSE на тестових даних: {rmse:.2f} кВт·год")
```

Одержано: 26.09.2025
Внутрішня рецензія отримана: 13.10.2025
Зовнішня рецензія отримана: 04.10.2025

***Про авторів*:**

Гайдукевич Владислав Олегович,
аспірант
https://orcid.org/0000-0002-0614-6778

Дорошенко Анатолій Юхимович,
доктор фізико-математичних наук,
професор,
завідувач відділу теорії комп'ютерних обчислень ІПС НАН України,
професор кафедри інформаційних систем та технологій КПІ імені Ігоря Сікорського.
http://orcid.org/0000-0002-8435-1451,

Місце роботи авторів:

Інститут програмних систем НАН України, 03187, м. Київ-187, проспект Академіка Глушкова, 40.
e-mail: doroshenkoanatoliy2@gmail.com, gaidukevichvlad@gmail.com

УДК 004.4 https://doi.org/10.15407/pp2025.03.039

В.І. Шинкаренко, О.В. Макаров

КОНСТРУКТИВНО-ПРОДУКЦІЙНЕ МОДЕЛЮВАННЯ ХРОМОСОМ ГЕНЕТИЧНОГО АЛГОРИТМУ ІЗ ЗАКОДОВАНИМИ АЛГОРИТМАМИ СОРТУВАННЯ

У разі структурної адаптації алгоритмів із використанням генетичного алгоритму однією із проблем є кодування структури алгоритмів у хромосому. У статті розглядається підхід до структурної адаптації та формування ефективних алгоритмів сортування на основі парадигми конструктивно-продукційного моделювання. Запропоновано метод представлення хромосом у вигляді закодованих структур, що відображають різні варіанти алгоритмів сортування. Такий підхід дозволяє формувати простір пошуку рішень не лише у вигляді набору числових параметрів, а й у вигляді цілісних програмних конструкцій. Описано операції підстановки, часткового виведення та композиції, які забезпечують синтез нових алгоритмів сортування. Для формування і відбору функціонально еквівалентних алгоритмів застосовано генетичний алгоритм. Наведено приклади побудови хромосом-дерев, що кодують сортувальні процедури різної складності. Реалізовано програму для генерації та еволюційного вдосконалення хромосом із закодованими алгоритмами сортування. продемонстровано застосування конструктивно-продукційного моделювання для вирішення задачі кодування структурно різних, але функціонально еквівалентних алгоритмів у хромосомах. Результати експериментів підтвердили, що застосування конструктивно-продукційного моделювання забезпечує підвищення різноманітності популяцій та прискорює знаходження ефективних рішень у порівнянні з класичними генетичними алгоритмами. Запропонована методика може бути використана для автоматизованого конструювання алгоритмів, оптимізації їхньої структури та адаптації до специфічних умов використання.

Ключові слова: конструктивно-продукційне моделювання, алгоритми сортування, часова ефективність, програмне забезпечення, інформаційні технології, генетичний алгоритм.

V.I. Shinkarenko, O.V. Makarov

CONSTRUCTIVE-SYNTHESIZING MODELING OF THE GENETIC ALGORITHM CHROMOSOMES WITH ENCODED SORTING ALGORITHMS

In the structural adaptation of algorithms using a genetic algorithm, one of the challenges is encoding the structure of algorithms into a chromosome. This article explores an approach to structural adaptation and the development of efficient sorting algorithms based on the paradigm of constructive-synthesizing modeling. A method is proposed for representing chromosomes as encoded structures corresponding to various variants of sorting algorithms. This approach allows the solution search space to be formed not only as a set of numerical parameters but also as complete software. Operations of substitution, partial inference, and composition are described, which enable the synthesis of new sorting algorithms. A genetic algorithm is applied to generate and select functionally equivalent algorithms. Examples are provided of constructing chromosome trees that encode sorting procedures of varying complexity. A program has been implemented for generating and evolutionarily improving chromosomes with encoded sorting algorithms. The application of constructive-synthesizing modeling to the problem of encoding structurally different but functionally equivalent algorithms into chromosomes is demonstrated. Experimental results confirmed that usage of constructive-synthesizing modeling increases population diversity and accelerates the discovery of efficient solutions compared to classical genetic algorithms. The proposed methodology can be used for automated algorithm construction, optimization of their structure, and adaptation to specific usage conditions.

Key words: constructive-synthesizing modeling, sorting algorithm, time efficiency, software, information technology, genetic algorithm.

Вступ

Структурна адаптація алгоритмів за показниками часової ефективності передбачає зміну складових цих алгоритмів. Для формування і відбору функціонально еквівалентних алгоритмів застосовується генетичний алгоритм. Тут постає проблема кодування алгоритмів, що підлягають адаптації, у вигляді хромосоми. Це ускладнюється ще й можливістю розподілу обчислень у різних частинах даних. Еталонними алгоритмами у програмуванні є алгоритми сортування. Тому на прикладі алгоритмів сортування у цій роботі продемонстровано застосування конструктивно-продукційного моделювання для вирішення задачі кодування структурно різних, але функціонально еквівалентних алгоритмів у хромосомах.

Незважаючи на наявність великої кількості фундаментальних алгоритмів сортування, таких як Insertion Sort, Quick Sort [1] або Merge Sort [2], які широко використовуються завдяки своїй простоті реалізації та зрозумілій логіці, проблема підвищення їхньої ефективності залишається актуальною. Зростання обсягів даних, розвиток багатоядерних і паралельних архітектур, а також потреба в оптимізації під кеш-пам'ять і специфічні структури даних зумовлюють появу нових підходів. Саме тому активно ведуться наукові дослідження, спрямовані як на удосконалення класичних методів (Insertion Sort, Merge Sort, Quicksort), так і на створення нових алгоритмів, здатних поєднувати простоту, стабільність і високу продуктивність у сучасних умовах.

До найбільш відомих прикладів побудови алгоритмів на основі кількох класичних належить Timsort [3] – гібридний алгоритм, що поєднує переваги Merge Sort та Insertion Sort і використовується в стандартних бібліотеках Python та Java, а також Introsort [4], який адаптивно перемикається між Quick Sort, Heap Sort і Insertion Sort залежно від глибини рекурсії, забезпечуючи гарантовану ефективність у найгіршому випадку.

Попри наявність ефективних рішень, розробка нових алгоритмів сортування продовжується. Одним із сучасних підходів до вдосконалення алгоритмів сортування є розробка паралельних методів, що враховують особливості розподілу даних та ефективність операцій ранжування. Наприклад, у роботі [5] запропоновано модель паралельного алгоритму сортування з формуванням рангу, яка демонструє переваги у швидкодії та масштабованості для великих обсягів даних.

Сучасні дослідження пропонують вдосконалення класичних алгоритмів сортування. Наприклад, Multi-Pivot Quicksort [6] – використання декількох опорних елементів (k-pivot), що покращує ефективність роботи із кешем та знижує кількість порівнянь. Однією із можливих реалізацій є варіант із трьома опорними елементами, який продемонстрував приріст продуктивності у 7-8% порівняно із класичним алгоритмом з 1 опорним елементом.

На відміну від класичного сортування вставками, у [7] масив має дві відсортовані послідовності на початку і в кінці масиву, а невідсортована частина знаходиться між ними. Також важливою перевагою є можливість вставляти більше одного елемента на правильні позиції за одну ітерацію. Експериментально алгоритм показав менший середній час виконання, ніж Quick Sort для відносно невеликих масивів обсягом до 1500 елементів.

Adaptive ShiversSort [8] – алгоритм сортування, заснований на TimSort. Особливо ефективний для сортування частково впорядкованих даних. Основна ідея полягає в тому, що масив спочатку розбивається на вже відсортовані підпослідовності, які виявляються під час одного проходу. Adaptive ShiversSort є 3-aware алгоритмом – тобто, вибираючи операції злиття, він аналізує лише три верхні елементи стека послідовностей, що знижує складність керування процесом. Алгоритм додатково приймає цілочислений параметр, який змінює логіку злиття послідовностей.

Magnetic Bubble Sort [9]: Іноваційний метод, який замість поодиноких перестановок сусідніх елементів, оперує «блоком» елементів. Такі блоки поводяться

як "магнітики", притягуючись або відштовхуючись залежно від значень. Такий підхід дозволяє «переносити» відразу кілька елементів, що економить кількість проходів масивом у випадках, коли в масиві багато повторів. У деяких тестах час сортування зменшився на 20 % порівняно з bubble sort.

One-By-One (OBO): A Fast Sorting Algorithm [10]: Новий in-place алгоритм сортування. Ідея полягає у поступовому "вивільненні" кожного елемента на правильне місце по черзі (one-by-one), комбінуючи стратегічне локальне впорядкування і глобальну оптимізацію. Такий підхід особливо ефективний у разі невеликих масивів і масивів, що частково або майже впорядковані, оскільки він зменшує непотрібні переміщення шляхом прямого позиціонування. OBO демонструє кращий середній час виконання в порівнянні з класичними квадратичними алгоритмами, наприклад Selection Sort і Bubble Sort.

Для відбору кращого за часовими показниками алгоритму сортування доцільне використання генетичного алгоритму [11] (ГА), що є поширеним інструментом для розв'язання складних оптимізаційних задач, які виникають у різних галузях науки і техніки. Він базується на принципах природного відбору та генетики і використовує популяцію можливих розв'язків, які еволюціонують з покоління в покоління [12]. Одним із ключових аспектів генетичного алгоритму є формування хромосом, які представляють можливі розв'язки задачі, у даному випадку конкретні алгоритми сортування.

Для формування хромосом у цій роботі використано конструктивно-продукційне моделювання (КПМ) – підхід, який дозволяє створювати складні структури шляхом застосування продукційних правил [13]. Цей метод широко використовується в різних галузях, таких як комп'ютерна графіка, моделювання біологічних систем та автоматизоване проєктування. Однією з основних переваг використання КПМ у генетичних алгоритмах є можливість створення структур, які відповідають певним обмеженням та вимогам задачі. Це дозволяє зменшити розмір пошукового простору та підвищити ефективність алгоритму. Крім того, КПМ дозволяє легко інтегрувати доменні знання у процес генерації хромосом, що може значно покращити якість розв'язків.

У цій статті буде детально розглянуто процес формування хромосом за допомогою КПМ, включаючи опис правил продукції.

Конструктивно-продукційна модель формування хромосоми

Узагальненим конструктором [13] називається трійка

$$C = \langle M, \Sigma, \Lambda \rangle, \qquad (1)$$

де M – неоднорідний розширюваний носій; Σ – сигнатура операцій та відношень; Λ – інформаційне забезпечення конструювання (ІЗК): онтологія, мета, правила та обмеження, початкова форма та умови закінчення.

Визначимо спеціалізацію конструктору C_{SA} – моделі формування хромосоми, що кодує алгоритм сортування. Виконаємо операцію уточнюючого перетворення узагальненого конструктора C.

$$C = \langle M, \Sigma, \Lambda \rangle \quad {}_{s}\!\longmapsto C_{SA} = \langle M_{SA}, \Sigma_{SA}, \Lambda_{SA} \rangle, \qquad (2)$$

де M_{SA} – носій, що включає, зокрема, термінальний (T_1) і нетермінальний(N_1) алфавіт, а також множину правил продукції Ψ з правилами $\psi_i = \langle s_i, g_i \rangle \in \Psi$, де i – номер правила, s_i – послідовність відношень підстановки, g_i – послідовність операцій над атрибутами. Σ_{SA} – операції та відносини на елементах M_{SA}. ІЗК $\Lambda_{SA} \supset \Lambda$.

Для формування конструкцій у вигляді хромосоми у Σ_{SA} передбачені операції підстановки, часткового та повного виводу. Вивід виконується від початкового нетерміналу, який є початковою формою ітераційним виконанням операцій часткового виводу. Операція часткового виводу обирає одне з правил із множини Ψ та виконує заміну лівої частини правила на праву у поточній формі та виконує операції над атрибутами. Вона використовує операцію підстановки: саме зміна поточної форми і є частиною операції повного виводу, починаючи з початкового нетерміналу і закінчуючи готовою конструкцією.

У цій роботі $g_i = \langle g_{i,1}, g_{i,2} \rangle \in \Psi$. Операції виконуються у наступній послідовності: спочатку виконуються операції над атрибутами $g_{i,1}$ потім операції підстановки s_i і в кінці – $g_{i,2}$.

ІЗК Λ_{SA} містить визначення, доповнення та обмеження, які уточнюють алфавіт, атрибути носія, відносини підстановки задають особливості виконання операцій підстановки та виведення.

Нетермінальний алфавіт $N_1 = \{{}_{\phi}\alpha_i\}$ складається з допоміжних символів з атрибутами, що позначаються грецькими літерами.

Термінальним алфавітом T_1 є множина генів, які утворюють хромосому, відповідну алгоритму сортування. Гени відповідають частинам відомих алгоритмів сортування, алгоритмам передобробки [14,15] даних і допоміжним функціям. Наведемо список усіх терміналів – генів, відповідно до:

- BSF (Bubble sort forward) – одного проходу алгоритму сортування бульбашкою;
- BSB (Bubble sort backward) – одного проходу алгоритму сортування бульбашкою у зворотному порядку;
- FSB (Find swap biggest element) – пошуку найбільшого елементу у невідсортованій частині масиву із перестановкою на поточне місце;
- FSS (Find swap smallest element) – пошуку найменшого елементу у невідсортованій частині масиву із перестановкою на поточне місце;
- SEEI (Single element end insertion) – операції вставки поточного елемента у відсортовану частину в кінці масиву;
- SEI (Single element insertion) – операції вставки поточного елемента у відсортовану частину на початку масиву;
- SIS (Split in subarrays) – операції розділення невідсортованої частини масиву на дві рівні частини для подальшого сортування кожної з них окремо. Також зберігання покажчиків та розмірів масивів для подальшого виконання операції злиття;
- P (Partition) – встановлення елемента під індексом і на правильне місце у відсортованому масиві і перекидання менших елементів ліворуч, а більших – праворуч;
- FS (Final sort) – одного із загальновідомих алгоритмів сортування (quicksort, mergesort, heapsort);
- PUA (Pop unsorted array) – дістання покажчика і розміру наступної невідсортованої частини для сортування;
- ESS (End subarray sort) – допоміжної операції, яка закриває фігурні дужки відкриті попередніми операціями для перевірки індикатора відсортованості;
- ML (Merge left) – збереження покажчиків на відсортовану частину на початку та решту масиву для подальшого злиття в один відсортований масив;
- MR (Merge right) – збереження покажчиків на відсортовану частину в кінці та решту масиву для подальшого злиття в один відсортований масив;
- QP (Quick Preprocess) – передбачення позиції елемента у відсортованому масиві й перестановка;
- PM (Preprocess with Memory) – те саме, що і QP, але передбачені індекси запам'ятовуються і перестановка виконується на наступне раніше не передбачувану позицію;
- SR (Sort Ranges) – розвертання усіх послідовностей елементів, що йдуть у зворотному порядку;
- BLQP (Block Local Quick Preprocessing) – розбивка сортованих даних на блоки певного обсягу і виконання швидкої перестановки у межах блоку;
- BLPM (Block Local Preprocessing with Memory) – розбивка сортованих даних на блоки певного обсягу і виконання перестановки із пам'яттю у межах блока;
- BGQP (Block Global Quick Preprocessing) – визначення позиції елемента (як у швидкій передобробці) і виконання перестановки тільки

якщо передбачена позиція перебуває у межах блоку;

- BGPM (Block Global Preprocessing with Memory) – визначення позиції елемента (як у передобробці з пам'яттю) і виконання перестановки, тільки якщо передбачена позиція є у межах блоку.

У процесі формування хромосоми [16] операції підстановки (на основі відпо-

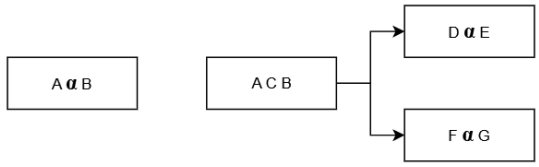


Рис 1. Початкова форма (зліва) і форма після виконання операції підстановки (справа)

відних відношень підстановки) реалізують додавання терміналів (генів) до вузлів дерева і нових вузлів до дерева. Розглянемо відношення підстановки.

Відношення $\beta \ _{\tau}\!\rightarrow \gamma$ дає можливість виконати заміну β на γ у збудованій частині хромосоми, якщо в ній є β і атрибут доступності τ дорівнює true.

$\beta \ _{\tau}\!\rightarrow \gamma \begin{matrix}\nearrow \rho \\ \searrow \delta\end{matrix}$, де β, γ, δ і ρ –деякі послідовності терміналів і нетерміналів. Відношення дає можливість виконати заміну β на γ у збудованій частині хромосоми, а конкретно у поточному вузлі хромосоми-дерева, якщо в ній є β і атрибут доступності τ дорівнює true. Відношення «↗» і «↘» дають можливість додати лівий і правий вузли нащадки до поточного вузла дерева і додати ρ і δ до відповідних вузлів.

Розглянемо наступний приклад правила підстановки.

$$\alpha \ _{\tau}\!\rightarrow C \begin{matrix}\nearrow D \cdot \alpha \cdot E \\ \searrow F \cdot \alpha \cdot G\end{matrix}, \qquad (3)$$

Поточна послідовність $A \cdot \alpha \cdot B$, де α – нетермінал і A, B – термінали. У ході виконання операції підстановки на основі відношення підстановки (3) буде додано ген C до початкової послідовності генів поточного вузла (Node level 1), два вузли-нащадки до поточного вузла. Також додано ген D і нетермінал α до початкової послідовності та ген E до кінцевої послідовності лівого вузла нащадка. Ген F і нетермінал α буде додано до початкової послідовності правого вузла нащадка, а ген G – відповідно до кінцевої. Результат підстановки представлено на рис. 1. Далі підстановки будуть продовжені відповідно для нетерміналів лівого та правого вузлів.

Визначені такі операції над атрибутами: :=(a, b) – присвоєння значення b змінній a; +(a, b, c) – додавання аргументів b і c, результат зберігається у a; ==(a, b, c) – якщо b дорівнює c, зберігає true у a, інакше – false; ∧(a, b, c) – виконання операції логічне «і» над агрументами b і c, результат зберігається у a; >(a, b, c) – порівнює аргументи b і c. Якщо b більше за c, присвоює a значення true, інакше – false; <(a, b, c) – порівнює аргументи b і c. Якщо b менше за c, присвоює a значення true, інакше – false; >=(a, b, c) – порівнює аргументи b і c. Якщо b більше або рівне c, присвоює a значення true, інакше – false; <=(a, b, c) – порівнює аргументи b і c. Якщо b менше або рівне c, присвоює a значення true, інакше – false.

Атрибути терміналів і нетерміналів поточної форми:

- b_1/b_2 – ознаки того, що елементи у відсортованій частині на початку/кінці масиву менші/більші за усі інші;
- cD – поточна глибина дерева-хромосоми;
- nG – кількість генів у поточному вузлі дерева-хромосоми;
- pGU – ознака того, що попередній використаний ген є будь-яким геном передобробки;
- MD – значення максимальної глибини дерева-хромосоми;
- MGBS – значення мінімальної кількості генів у вузлі, після досягнення якого стають доступними гени розбиття.

Початкові умови конструювання: α – нетермінал, з якого починається виведення.

Умови завершення конструювання: Хромосома повністю сконструйована (поточна форма не містить нетерміналів).

Правила підстановки конструктора формування хромосом

У підході, що розглядається, після додавання кожного нового вузла до дерева-хромосоми, одразу додаються ген PUA до початкової послідовності генів і ген ESS – до кінцевої.

Перша група правил містить відношення підстановки генів, що відповідають частинам існуючих алгоритмів сортування. Від значення атрибута залежить кількість доданих терміналів і значення атрибутів, які змінюються після виконання підстановки.

Детально розглянемо правило $\psi_1 = \langle s_1, \langle g_{1,1}, g_{1,2} \rangle\rangle$.

До операцій підстановки виконуються операції над атрибутами $g_{1,1}$. Встановлюються флаги τ_1 і τ_2, що роблять доступними відношення s_1.

Виконання операції підстановки за вибраним відношенням підстановки $\alpha \ {}_{\tau_1}\!\!\rightarrow FSS \cdot \alpha$ забезпечить додавання терміналу FSS до початкової послідовності генів поточного вузла і нетермінала α. Якщо значення атрибута τ_2 буде дорівнювати true, то доступним стає відношення $\alpha \ {}_{\tau_2}\!\!\rightarrow ML \cdot FSS \cdot \alpha$ і замість одного буде додано два термінали: ML і FSS.

Після операцій підстановки виконуються операції над атрибутами $g_{1,2}$. Поточні значення деяких атрибутів можуть змінювати цю поведінку. Якщо значення атрибута b_1 дорівнює true, то значення атрибута nG збільшиться на 1, і на 2 якщо b_1 дорівнює false. Далі втановлюються значення атрибутів pGU = false і b_1 = true.

$$s_1 = \langle {}^{\alpha\ \tau_1 \rightarrow FSS\cdot\alpha,}_{\alpha\ \tau_2 \rightarrow ML\cdot FSS\cdot\alpha.} \rangle;$$
$$g_{1,1} = \langle {}^{==(\tau_1, b_1, \text{true}),}_{==(\tau_2, b_1, \text{false}).} \rangle;$$
$$g_{1,2} = \langle {}^{\text{if}\left(b_1, +(nG, nG, 1), +(nG, nG, 2)\right),}_{:=(pGU, \text{false}),} {}_{:=(b_1, \text{true}).} \rangle \quad (4)$$

Аналогічно до правила ψ_1 у правилі $\psi_2 = \langle s_2, \langle g_{2,1}, g_{2,2} \rangle\rangle$ до початкової послідовності генів додається один (FSB) або два ($MR \cdot FSB$) термінали залежно від значення атрибутів τ_1 і τ_2.

$$s_2 = \langle {}^{\alpha\ \tau_1 \rightarrow FSB\cdot\alpha,}_{\alpha\ \tau_2 \rightarrow MR\cdot FSB\cdot\alpha.} \rangle;$$
$$g_{2,1} = \langle {}^{==(\tau_1, b_2, \text{true}),}_{==(\tau_2, b_2, \text{false}).} \rangle;$$
$$g_{2,2} = \langle {}^{\text{if}\left(b_2, +(nG, nG, 1), +(nG, nG, 2)\right),}_{:=(pGU, \text{false}),} {}_{:=(b_1, \text{true}).} \rangle \quad (5)$$

Наступні правила містять відношення підстановки генів SEI і SEEI:

$$s_3 = \langle \alpha \rightarrow SEI \cdot \alpha \rangle;\ g_{3,1} = \langle \varepsilon \rangle;$$
$$g_{3,2} = \langle {}^{+(nG, nG, 1),}_{:=(pGU, \text{false}),} {}_{:=(b_1, \text{false}).} \rangle \quad (6)$$

$$s_4 = \langle \alpha \rightarrow SEEI \cdot \alpha \rangle;\ g_{4,1} = \langle \varepsilon \rangle;$$
$$g_{4,2} = \langle {}^{+(nG, nG, 1),}_{:=(pGU, \text{false}),} {}_{:=(b_2, \text{false}).} \rangle \quad (7)$$

Символ ε – порожньо, означає відсутність операцій над атрибутами до операцій підстановки.

Після виконання підстановки значення атрибута nG збільшується на 1, pGU стає рівним false. Відповідно для першого і другого правила виконується b1 = false або b2 = false.

Правило $\psi_5 = \langle s_5, \langle g_{5,1}, g_{5,2} \rangle\rangle$ містить відношення підстановки гена BSF:

Якщо значення атрибута τ_1 істинне, доступне відношення ${}_{\tau_1}\!\!\rightarrow$, і до поточного вузла буде додано ген BSF і нетермінал α, для якого будуть продовжені підстановки. Якщо істинне значення τ_2, то ${}_{\tau_2}\!\!\rightarrow$ доступне. Буде додано два гени MR і BSF та нетермінал α. Правило підстановки буде виконуватись, якщо хоча б одне відношення доступне.

$$s_5 = \langle {}^{\alpha\ \tau_1 \rightarrow BSF\cdot\alpha,}_{\alpha\ \tau_2 \rightarrow MR\cdot BSF\cdot\alpha.} \rangle;\ g_{5,1} = g_{2,1};$$
$$g_{5,2} = g_{2,2} \quad (8)$$

Правило $\psi_6 = \langle s_6, \langle g_{6,1}, g_{6,2} \rangle\rangle$ містить відношення підстановки для додавання (під час реалізації конструктора) гена BSB у частково сформовану хромосому:

$$s_6 = \langle {}^{\alpha\ \tau_1 \rightarrow BSB\cdot\alpha,}_{\alpha\ \tau_2 \rightarrow ML\cdot BSB\cdot\alpha.} \rangle;\ g_{6,1} = g_{1,1};$$
$$g_{6,2} = g_{1,2} \quad (9)$$

Наступні два правила містять відношення підстановки генів розбиття P і SIS – у процесі виконання підстановки, за

якими до поточного вузла додаються два вузли-нащадки. Виконання підстановок завершується для поточного вузла і буде продовжено для вузлів-нащадків.

У результаті підстановки s_7 до поточного вузла додається ген P і два вузли-нащадки (лівий і правий). Далі паралельно виконуються 2 підстановки. Для доданих нетерміналів будуть продовжені підстановки відповідно у лівий і правий вузли. Гени, що стоять зліва від нетермінала α, будуть додані у початкову послідовність генів вузла, справа – у кінцеву послідовність.

$$\begin{gathered} s_7 = \langle \alpha \ {}_{\tau_1}\!\rightarrow P^{\nearrow PUA\cdot\alpha\cdot ESS}_{\searrow PUA\cdot\alpha\cdot ESS}\rangle; \\ g_{7,1} = \langle {}^{<(f_1,\mathrm{cD},\mathrm{MD}),}_{>=(f_2,\mathrm{nG},\mathrm{MGBS}),}{}^{\wedge(\tau_1,f_1,f_2).}\rangle; \\ g_{7,2} = \langle {}^{+(\mathrm{cD},\mathrm{cD},1),}_{:=(\mathrm{pGU},\mathrm{false}),}{}^{:=(\mathrm{nG},0).}\rangle \end{gathered} \quad (10)$$

Правило $\psi_8 = \langle s_8, \langle g_{8,1}, g_{8,2}\rangle\rangle$ містить відношення підстановки гена SIS:

$$\begin{gathered} s_8 = \langle \alpha \ {}_{\tau_1}\!\rightarrow SIS^{\nearrow PUA\cdot\alpha\cdot ESS}_{\searrow PUA\cdot\alpha\cdot ESS}\rangle, \\ g_{8,1} = g_{7,1};\ g_{8,2} = g_{7,2} \end{gathered} \quad (11)$$

Правило $\psi_9 = \langle s_9, \langle g_{9,1}, g_{9,2}\rangle\rangle$ містить відношення підстановки гена фінального сортування FS. Даний ген використовується для гарантованого відсортування поточної частини масиву сортованих даних, тому в результаті підстановки до послідовності не додаються нетермінали і після не виконуються операції над атрибутами.

Наступне правило $\psi_{10} = \langle s_{10}, \langle g_{10,1}, g_{10,2}\rangle\rangle$ містить відношення підстановки генів передобробки. До операції підстановки виконується операція над атрибутом і встановлюється значення τ_1. У процесі виконання операції підстановки до послідовності терміналів додається ген передобробки і нетермінал α. Символ "|" позначає відношення «або»: можливість виконання лише однієї із зазначених підстановок. Потім значення атрибута nG збільшується на 1, pGU стає рівним false.

$$\begin{gathered} s_{10} = \langle \alpha \ {}_{\tau_1}\!\rightarrow QP\cdot\alpha \mid PM\cdot \\ \alpha \mid SR\cdot\alpha \mid BLQP\cdot\alpha \mid BLPM\cdot \\ \alpha \mid BGQP\cdot\alpha \mid BGPM\cdot\alpha\rangle; \\ g_{10,1} = \langle == (\tau_1, \mathrm{pGU}, \mathrm{false})\rangle; \\ g_{10,2} = \langle {}^{+(\mathrm{nG},\mathrm{nG},1),}_{:=(\mathrm{pGU},\mathrm{true}).}\rangle \end{gathered} \quad (12)$$

Розширений приклад формування хромосоми

Розглянемо приклад покрокового процесу формування хромосоми-дерева. На початку встановлюються наступні початкові значення параметрів конструювання і атрибутів: MD = 3, MGBS = 1, b1 = true, b2 = true, cD = 1, nG = 0, pGU = false.

Розглянемо послідовність підстановок для кореневого вузла хромосоми рівня 1 під номером 1. Першим обирається правило підстановки ψ_{10}, додається ген швидкої передобробки QP, встановлюються значення атрибутів $nG = 1; pGU = true;$. Наступними обираються два однакових правила ψ_5. Додаються гени BSF, встановлюються атрибути $nG = 3; pGU = false;$. Останнім обирається правило ψ_8, що завершує вибір правил для поточного вузла, додає ген SIS і встановлює значення атрибутів cD=1; pGU=false; nG=0;. Також додає два вузли - нащадки і відповідні термінали і нетермінали у їхній пос-

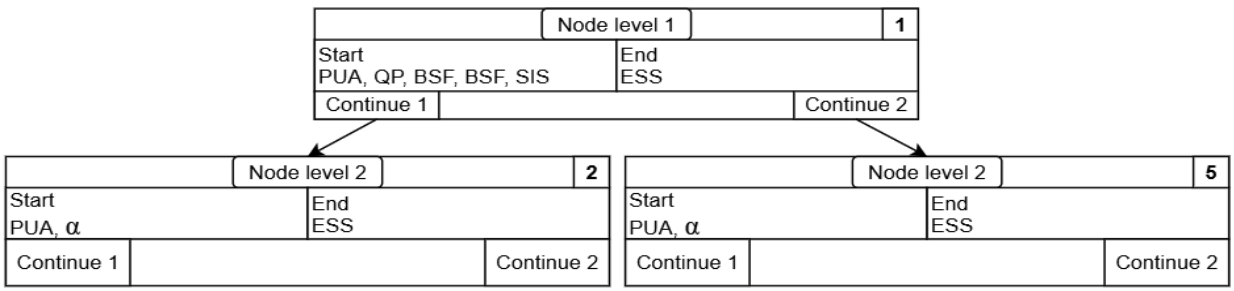


Рис. 2. Змінена форма після застосування правил $\boldsymbol{\psi_{10}}, \boldsymbol{\psi_5}, \boldsymbol{\psi_5}, \boldsymbol{\psi_8}$

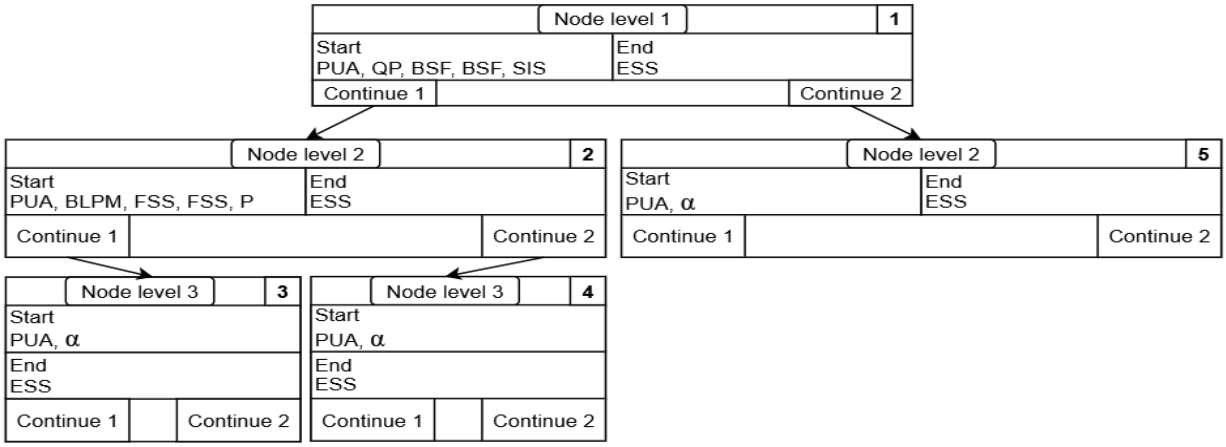


Рис. 3. Змінена форма після застосування правил $\boldsymbol{\psi_{14}}$, $\boldsymbol{\psi_1}$, $\boldsymbol{\psi_1}$, $\boldsymbol{\psi_7}$ щодо форми на рис. 2

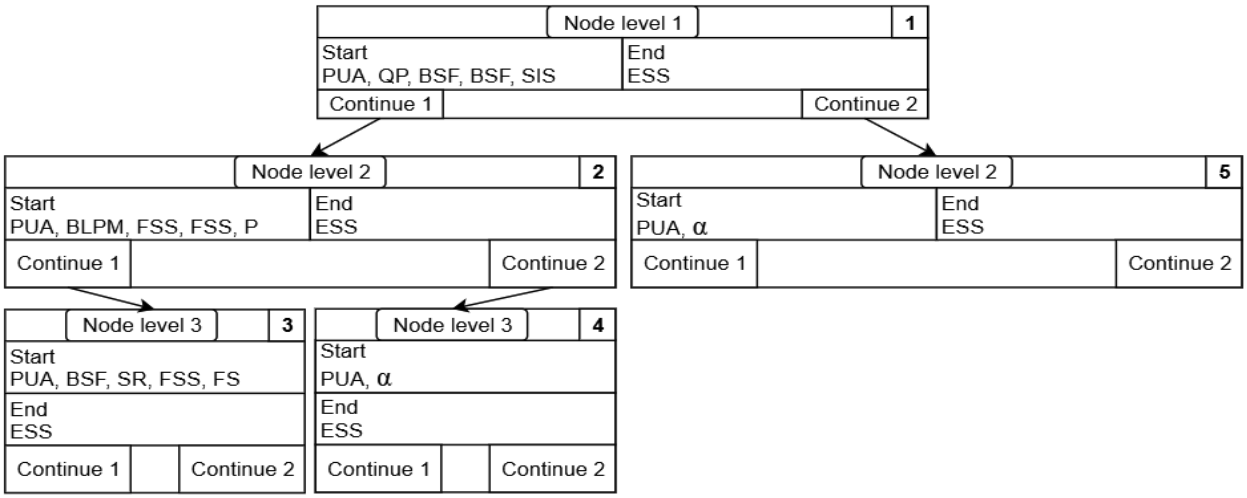


Рис. 4. Змінена форма після застосування правил $\boldsymbol{\psi_5}$, $\boldsymbol{\psi_{12}}$, $\boldsymbol{\psi_1}$, $\boldsymbol{\psi_9}$ щодо форми на рис 3

лідовності генів (рис. 2). Нерозкриті нетермінали лишились у блоках 2 та 5.

Наступними виконуються підстановки для вузла 2 (рис. 3). Обирається правило підстановки ψ_{14}. Додається ген BLPM та встановлюються значення атрибутів $nG = 1; pGU = true;$. Далі обираються два правила ψ_1. Додаються гени FSS і встановлюються значення атрибутів $nG = 3; pGU = false;$.

Обирається останнє правило ψ_7. Додається ген і встановлюються значення атрибутів cD=2; pGU=false; nG=0;. Додаються два вузли-нащадки під номерами 3 і 4 до поточного вузла, також гени і нетермінали.

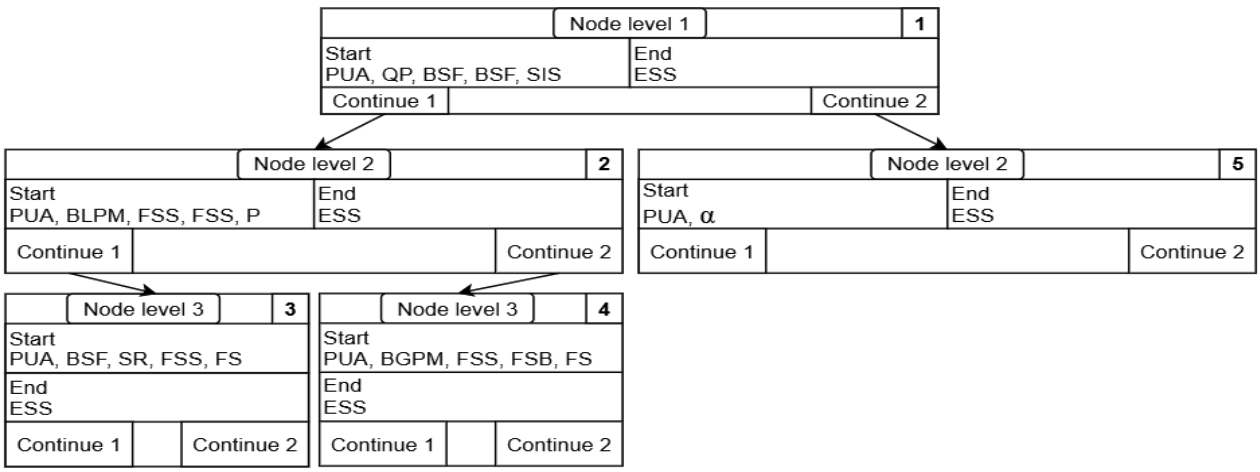


Рис. 5. Змінена форма після застосування правил $\boldsymbol{\psi_{14}}$, $\boldsymbol{\psi_1}$, $\boldsymbol{\psi_2}$, $\boldsymbol{\psi_9}$ щодо форми на рис. 4

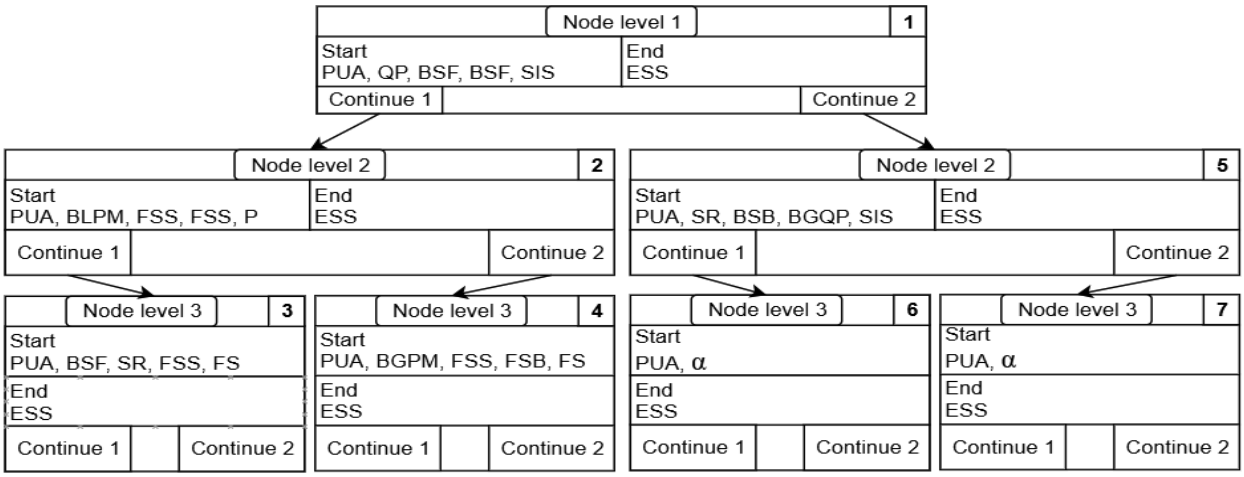


Рис. 6. Змінена форма після застосування правил $\boldsymbol{\psi_{12}}$, $\boldsymbol{\psi_6}$, $\boldsymbol{\psi_{15}}$, $\boldsymbol{\psi_8}$ щодо форми на рис. 5

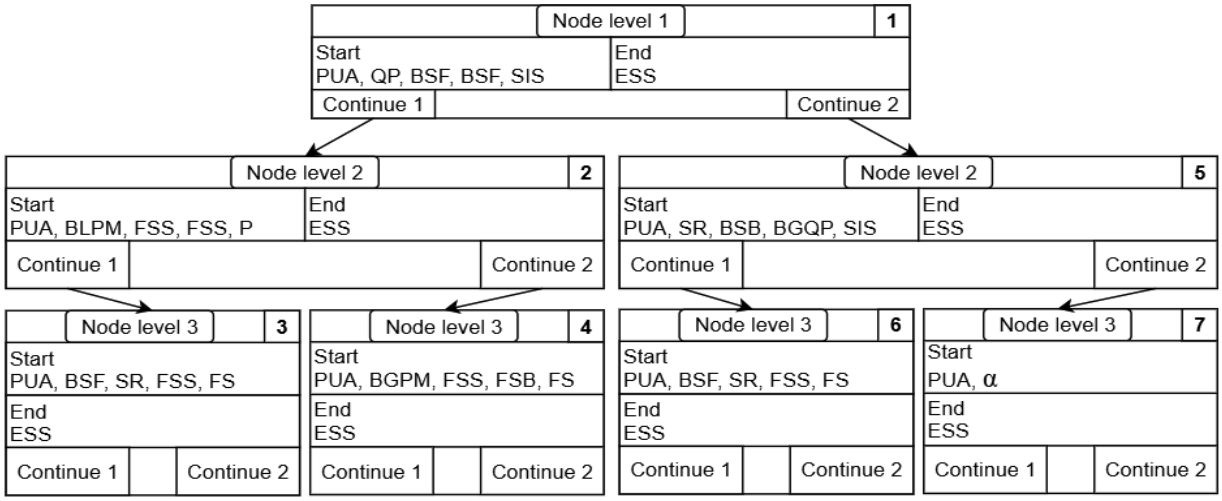


Рис. 7. Змінена форма після застосування правил $\boldsymbol{\psi_5}$, $\boldsymbol{\psi_{12}}$, $\boldsymbol{\psi_1}$, $\boldsymbol{\psi_5}$ щодо форми на рис. 6

Наступним виконуються підстановки для вузла 3 (рис. 4). Обирається правило підстановки ψ_5. Додається ген BSF, встановлюються атрибути $nG = 1$; $pGU = false$;. Під час виконання підстановки за правилом ψ_{12}, до послідоності генів додається ген SR, встановлюються значення атрибутів $nG = 2; pGU = true$;. Далі обирається правило ψ_1, додається ген FSS, встановлюються значення атрибутів $nG = 3; pGU = false$;. Оскільки досягнута максимальна глибина хромосоми-дерева, гени розбиття стають недоступними і обирається правило фінального сортування ψ_9. Нові вузли не створюються і формування поточного вузла завершується.

Наступними виконуються підстановки для вузла 4 (рис. 5). Обираються правила підстановки ψ_{14}, ψ_1, ψ_2. Оскільки досягнута максимальна глибина хромосоми-дерева, гени розбиття стають недоступні і обирається правило фінального сортування ψ_9. Нові вузли не створюються і формування поточного вузла завершується.

Виконуються підстановки для вузла 5 (рис. 6). Обираються правила підстановки ψ_{12}, ψ_6, ψ_{15}, ψ_8. Додаються два вузли-нащадки під номерами 6 і 7 до поточного вузла, додаються гени і нетермінали.

Виконуються підстановки для вузла 6 (рис. 7). Обираються правила підстановки ψ_5, ψ_{12}, ψ_1. Оскільки досягнута максимальна глибина хромосоми-дерева, гени розбиття стають недоступними і обирається правило фінального сортування ψ_9. Нові вузли не створюються і формування поточного вузла завершується.

Виконуються підстановки для вузла 7 (рис. 8). Обираються правила підстановки ψ_{15}, ψ_2, ψ_6. Оскільки досягнута максимальна глибина хромосоми-дерева, гени розбиття стають недоступні і обирається правило фінального сортування ψ_9. Нові

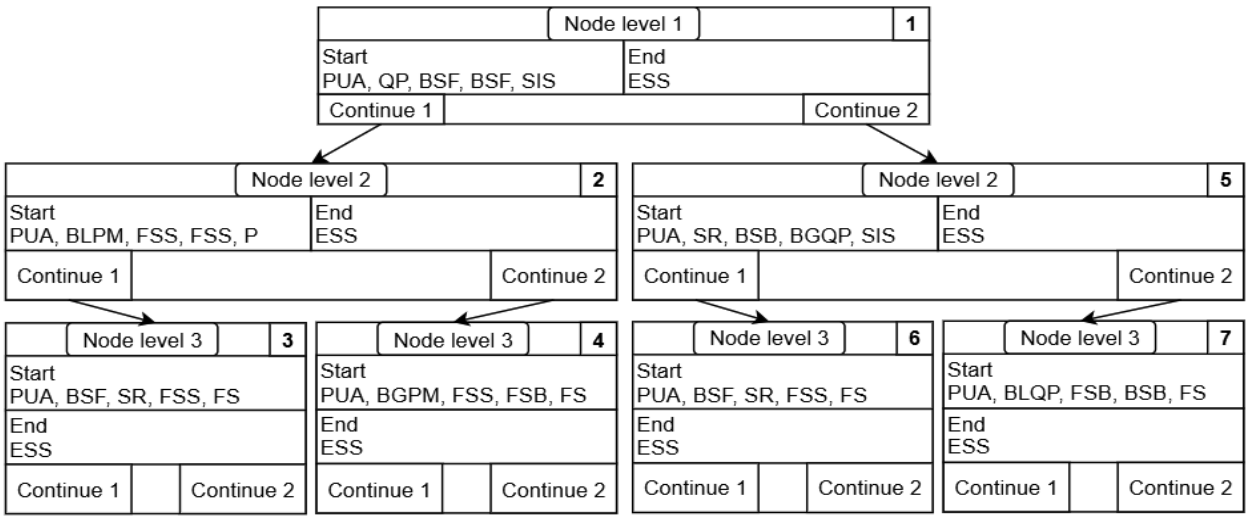


Рис. 8. Змінена форма після застосування правил $\boldsymbol{\psi_{15}}, \boldsymbol{\psi_{2}}, \boldsymbol{\psi_{6}}, \boldsymbol{\psi_{15}}$ щодо форми на рис. 7

вузли не створюються і формування поточного вузла завершується.

Використання хромосом генетичним алгоритмом

Для застосування генетичного алгоритму необхідно сформувати початкову популяцію хромосом. Потрібне перетворення хромосоми у текст програми і виконання сортування експериментальних даних кожним алгоритмом-індивідом популяції. Здійснюється пошук певної кількості кращих алгоритмів. Кращим вважається алгоритм, що виконує сортування за менший проміжок часу. Потрібен відбір певної кількості кращих хромосом і додавання їх до наступної популяції. Далі - формування нових хромосом із застосуванням механізмів схрещення і мутацій, а також додавання нових.

Далі представлений скелет програми.

[Крок 1. Налаштування параметрів експерименту]
[Крок 2. Генерація популяції]
[Крок 3. Генерація вхідних даних]
[Крок 4. Оцінка хромосоми]
[Крок 4.1. Декодування хромосоми в код алгоритму сортування]
[Крок 4.1.1. Для кожного гена]
[Крок 4.1.1.1. Отримати код, що відповідає переданому гену]
[Крок 4.1.1.2. Додати код гена до коду алгоритму сортування]
[Крок 4.2. Побудова програми для запуску згенерованого алгоритму сортування]
[Крок 4.3. Запуск програми, яка сортує вхідні дані та вимірює час виконання]
[Крок 5. Отримання результатів оцінки]
[Крок 6. Генерація нової популяції або завершення експерименту]

Приклад коду формування тексту програми за геном QP у кроці 4.1.1.1.

```
std::tie(min, max) = Utilities::Find-
MinMax(arr + s, e - s + 1);
if (min == max)
{
 isSorted = true;
}
if (!isSorted)
{
 auto range = max - min;
 auto maxIndex = e - s;
 auto maxStepsQP = std::min(std::max((e -
s + 1) / 1000, 10), 100);
 for (int i = s; i <= e; ++i)
 {
  int j = 0;
  while (j++ < maxStepsQP)
  {
   int predictedPlace = double(arr[i] - min)
/ range * maxIndex + s;
   if (i == predictedPlace || arr[i] ==
arr[predictedPlace])
   {
    break;
   }
  std::swap(arr[i], arr[predictedPlace]);
  }
}
```

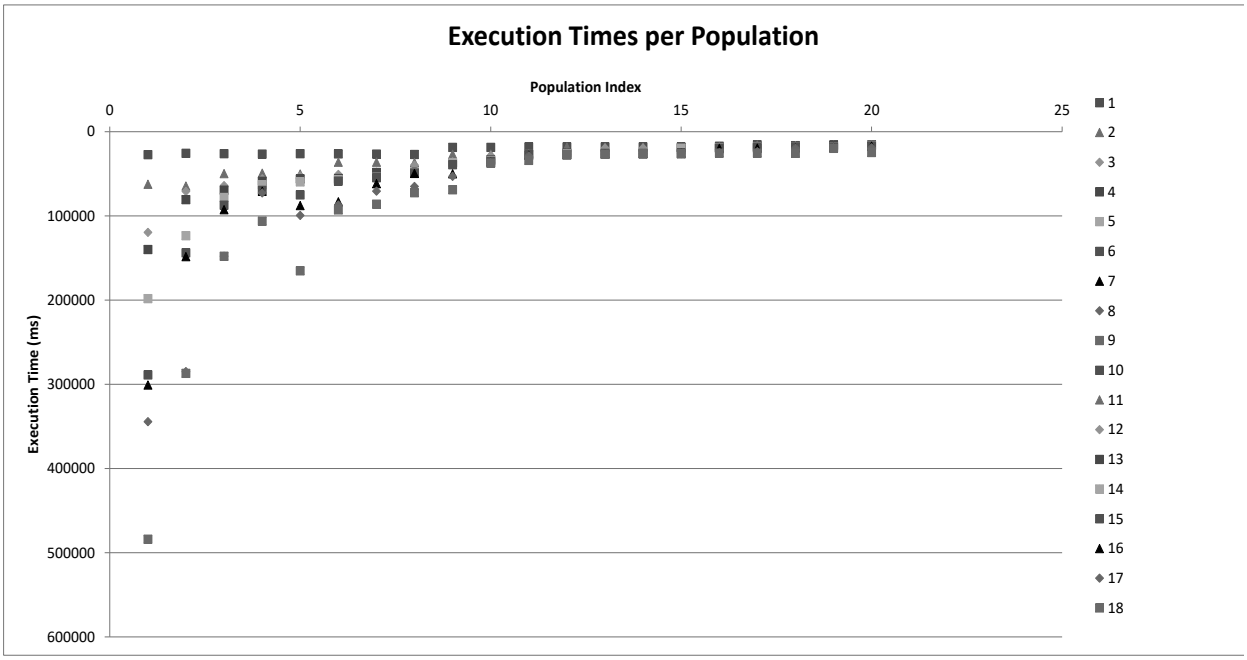


Рис. 9. Зміна часу сортування у різних поколіннях хромосом

У тексті програми: *s* – індекс першого елемента сортованого підмасива; *e* – індекс останнього елемента сортованого підмасива; *maxStepsQP* – максимальна кількість передбачень для кожного елемента; *min*, *max* – значення найменшого і найбільшого елементів сортованого підмасиву; *arr* – сортований масив.

Експериментальні дослідження

У ході роботи було проведено багато експериментів із різними типами і обсягами даних за різних початкових умов і параметрів завершення. Далі представлено один із експериментів еволюційного вибору кращого алгоритму сортування спецефічних даних.

Специфікація полягає у наступному. Експеримент проводився на даних обсягом один мільйон елементів. Масив заповнювався випадковими числами у діапазоні від нуля до одного мільйона, потім виконувалось повне сортування даних. Далі у повністю відсортованому масиві вибирались N ділянок з сумарною довжиною L. У межах кожної ділянки випадковим чином вибирались 2 індекси, і елементи за обраними індексами мінялись місцями, процес повторювався кількість разів відповідну довжині ділянки. Наведені експерименти виконувались на ноутбуці із технічними характеристиками, представленими у табл.1.

Таблиця 1.

Технічні засоби експерименту

Характеристики центрального процесору	
Поле	Значення
Тип	Intel Core i7-12650H (12th Gen)
Кількість ядер	10 cores (6 Performance + 4 Efficiency)
Кеш L1	864 KB
Кеш L2	9.5 MB
Кеш L3	24 MB
Характеристики оперативної пам’яті	
Поле	Значення
Тип	DDR5-4800
Розмір модулю	8 ГБ
Кількість модулів	2 x 8 ГБ

На рис. 9 представлені результати експерименту. До останньої популяції додано 4 загальновідомі алгоритми для порівняння: швидке, злиттям, вставками і сортування вибором. Кращий алгоритм знаходиться праворуч вгорі, нижче йдуть усі інші алгоритми у порядку погіршення часової ефективності.

Далі представлені п’ять кращих хромосом і усі загальновідомі алгоритми, позначені на графіку числами відповідно:

1. SR_SIS_PUA_P_PUA_BSB_BSF_FS_ESS_PUA_FSS_BSB_BSB_FS_ESS_ESS_PUA_P_PUA_FSB_FS_ESS_PUA_FS_ESS_ESS_ESS
2. SR_SIS_PUA_P_PUA_BSB_FSS_FS_ESS_PUA_FSB_BSB_BSB_FS_ESS_ESS_PUA_P_PUA_FSB_FS_ESS_PUA_FS_ESS_ESS_ESS
3. SR_SIS_PUA_P_PUA_BSB_FSS_FS_ESS_PUA_FSS_BSB_BSB_FS_ESS_ESS_PUA_P_PUA_FSB_FS_ESS_PUA_FS_ESS_ESS_ESS
4. SR_SIS_PUA_P_PUA_BSB_FSS_FS_ESS_PUA_FSS_BSB_BSB_FS_ESS_ESS_PUA_P_PUA_FSB_FS_ESS_PUA_FS_ESS_ESS_ESS
5. QuickSort

   …
14. HeapSort

   …
18. MergeSort

   …
20. InsertionSort

На графіку надано час сортування різних алгоритмів. Перші чотири алгоритми (позиції 1–4) сформовані генетичним алгоритмом на основі підходу, представленого в даній роботі. Час сортування порівнюється з класичними алгоритмами сортування, які розмістились на позиціях 5, 14, 18 та 20.

Результати демонструють, що конструйовані алгоритми, представлені хромосомами, здатні перевершити традиційні за швидкістю виконання на тестових наборах даних. Підтверджують доцільність використання нових алгоритмів у задачах, де критичним є час обробки великих масивів інформації.

Висновки

У разі структурної адаптації алгоритмів досягається можливість гнучко перебудовувати їхню внутрішню структуру залежно від властивостей вхідних даних, цільових вимог чи обмежень обчислювального середовища. Такий підхід дозволяє поєднувати фундаментальні алгоритмічні фрагменти у нові структури, що зберігають коректність і водночас покращують часові та ресурсні характеристики.

Розроблено підхід до формування хромосом, які відповідають алгоритмам сортування, сформованим із частин класичних алгоритмів сортування (таких як Bubble Sort, Quick Sort, Merge Sort) і деяких передобробок у вигляді структур, придатних для генетичних та еволюційних обчислень.

Конструктивно-продукційне моделювання продемонструвало ефективність у формуванні хромосом, що забезпечує можливість структурної адаптації алгоритмів для специфічних вхідних даних.

Запропонована модель забезпечує гнучкість у генерації хромосом з урахуванням різних критеріїв оптимізації, таких як час виконання. Структура кодованого дерева забезпечує уніфікований підхід до процесу генерації послідовності генів для кожного вузла за допомогою рекурсії. Таким чином уможливлюється генерування хромосом будь-якої довжини.

Проведені експерименти підтвердили, що структуровані хромосоми, сформовані за допомогою продукційних правил, демонструють вищу ефективність у задачах еволюційного пошуку порівняно з випадковою ініціалізацією.

Отримані результати можуть бути використані для автоматизованого синтезу алгоритмів, оптимізації програмного коду та побудови інтелектуальних систем, що навчаються.

Література

1. C. A. R. Hoare. Quicksort. The Computer Journal. – Vol. 5, No. 1. – pp. 10–16. – 1962.
2. D. E. Knuth. Sorting by Exchanging. The Art of Computer Programming. – Vol. 3: Sorting and Searching. – 1st ed. – Addison-Wesley. – pp. 110–111. – 1973.
3. N. Auger, V. Jugé, C. Nicaud, C. Pivoteau. On the worst-case complexity of Timsort. 26th Annual European Symposium on Algorithms (ESA 2018). – LIPIcs, Vol. 112. – pp. 4:1–13. – 2018. – Available: https://arxiv.org/abs/1805.08612
4. D. R. Musser. Introspective Sorting and Selection Algorithms. Software: Practice and Experience. – Vol. 27, No. 8. – pp. 983–993. – 1997.
5. T. B. Martyniuk, B. I. Krukivskyi. Peculiarities of the parallel sorting algorithm with rank

formation. Cybernetics and Systems Analysis. – Vol. 58, No. 1. – pp. 24–28. – 2022. – DOI: https://doi.org/10.1007/s10559-022-00431-8.

6. M. Dietzfelbinger. How Good Is Multi-Pivot Quicksort?. ArXiv, Cornell University. – 2015.
7. A. S. Mohammed, Ş. E. Amrahov, F. V. Çelebi. Bidirectional Conditional Insertion Sort algorithm: An efficient progress on the classical insertion sort. Future Generation Computer Systems. – Vol. 71. – pp. 102–112. – June 2017.
8. V. Jugé. Adaptive shivers sort: an alternative sorting algorithm. ACM Transactions on Algorithms. – Vol. 20, No. 4. – pp. 1–55. – August 2024.
9. O. Appiah, E. M. Martey. Magnetic Bubble Sort Algorithm. International Journal of Computer Applications. – Vol. 122, No. 21. – pp. 24–28. – 2015. – DOI: 10.5120/21850-5168
10. A. Alotaibi, A. Almutairi, H. Kurdi. Onebyone (obo): A fast sorting algorithm. Procedia Computer Science. – Vol. 175. – pp. 270–277. – 2020.
11. F. M. Isa, W. N. M. Ariffin, M. S. Jusoh, E. P. Putri. A Review of Genetic Algorithm: Operations and Applications. Journal of Advanced Research in Applied Sciences and Engineering Technology. – 2020. – DOI: 10.37934/araset.40.1.134
12. Z. Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs. – 3rd ed. – Springer. – 1996.
13. В.І. Шинкаренко, В.М. Ільман. Конструктивно-продукційні структури та їх граматичні інтерпретації. І. Узагальнена формальна конструктивно-продукційна структура . Кібернетика і системний аналіз. – 2014. – Т. 50, № 5. – С. 8–16.
14. В.І. Шинкаренко, А.Ю. Дорошенко, О.А. Яценко, В.В. Разносілін, К.К. Галанін, Двокомпонентні алгоритми сортування, Проблеми програмування, 2022, № 3-4. С. 32-41. doi: 10.15407/pp2022.03-04.032.
15. В.І. Шинкаренко, О.В. Макаров, Дослідження ефективності деяких детермінованих методів передобробки сортування даних, Проблеми програмування, 2023, № 4, С. 3-14. doi: 10.15407/pp2023.04.003.
16. V. I. Shynkarenko, O. V. Makarov. Structural Adaptation of Sorting Algorithms Based on Constructive Fragments. CEUR Workshop Proceedings. – Vol. 3806. – pp. 16–29. – 2024.

References

1. C. A. R. Hoare. Quicksort. The Computer Journal. – Vol. 5, No. 1. – pp. 10–16. – 1962.
2. D. E. Knuth. Sorting by Exchanging. The Art of Computer Programming. – Vol. 3: Sorting and Searching. – 1st ed. – Addison-Wesley. – pp. 110–111. – 1973.
3. N. Auger, V. Jugé, C. Nicaud, C. Pivoteau. On the worst-case complexity of Timsort. 26th Annual European Symposium on Algorithms (ESA 2018). – LIPIcs, Vol. 112. – pp. 4:1–13. – 2018. – Available: https://arxiv.org/abs/1805.08612
4. D. R. Musser. Introspective Sorting and Selection Algorithms. Software: Practice and Experience. – Vol. 27, No. 8. – pp. 983–993. – 1997.
5. T. B. Martyniuk, B. I. Krukivskyi. Peculiarities of the parallel sorting algorithm with rank formation. Cybernetics and Systems Analysis. – Vol. 58, No. 1. – pp. 24–28. – 2022. – DOI: https://doi.org/10.1007/s10559-022-00431-8
6. M. Dietzfelbinger. How Good Is Multi-Pivot Quicksort?. ArXiv, Cornell University. – 2015.
7. A. S. Mohammed, Ş. E. Amrahov, F. V. Çelebi. Bidirectional Conditional Insertion Sort algorithm: An efficient progress on the classical insertion sort. Future Generation Computer Systems. – Vol. 71. – pp. 102–112. – June 2017.
8. V. Jugé. Adaptive shivers sort: an alternative sorting algorithm. ACM Transactions on Algorithms. – Vol. 20, No. 4. – pp. 1–55. – August 2024.
9. O. Appiah, E. M. Martey. Magnetic Bubble Sort Algorithm. International Journal of Computer Applications. – Vol. 122, No. 21. – pp. 24–28. – 2015. – DOI: 10.5120/21850-5168
10. A. Alotaibi, A. Almutairi, H. Kurdi. Onebyone (obo): A fast sorting algorithm. Procedia Computer Science. – Vol. 175. – pp. 270–277. – 2020.
11. F. M. Isa, W. N. M. Ariffin, M. S. Jusoh, E. P. Putri. A Review of Genetic Algorithm: Operations and Applications. Journal of Advanced Research in Applied Sciences and Engineering Technology. – 2020. – DOI: 10.37934/araset.40.1.134
12. Z. Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs. – 3rd ed. – Springer. – 1996.
13. V. I. Shynkarenko, V. M. Ilman. Constructive-Synthesizing Structures and Their Grammatical Interpretations. I. Generalized Formal Constructive-Synthesizing Structure.

Cybernetics and Systems Analysis. – Vol. 50, No. 5. – pp. 8–16.
14. V. I. Shinkarenko, A. Yu. Doroshenko, O. A. Yatsenko, V. V. Raznosilin, K. K. Galanin. Bicomponent sorting algorithms. Problems of Programming. – 2022. – No. 3–4. – pp. 32–41. – DOI: 10.15407/pp2022.03-04.032
15. V. I. Shinkarenko, O. V. Makarov. Study of the effectiveness of some deterministic preprocessing methods of data sorting. Problems of Programming. – 2023. – No. 4. – pp. 3–14. – DOI: 10.15407/pp2023.04.003
16. V. I. Shynkarenko, O. V. Makarov. Structural Adaptation of Sorting Algorithms Based on Constructive Fragments. CEUR Workshop Proceedings. – Vol. 3806. – pp. 16–29. – 2024.

Одержано: 12.10.2025
Внутрішня рецензія отримана: 18.10.2025
Зовнішня рецензія отримана: 19.10.2025

Про авторів:

Шинкаренко Віктор Іванович,
доктор технічних наук, професор,
https://orcid.org/0000-0001-8738-7225
E-mail: shinkarenko_vi@ua.fm

Макаров Олексій Вікторович,
аспірант,
https://orcid.org/0009-0003-0921-155X
E-mail: makarovov@hotmail.com

Місце роботи авторів:

Український державний університет
науки і технологій,
49010, Україна, Дніпро,
вул. Академіка Лазаряна, 2.
E-mail: office@ust.edu.ua

УДК 004.89:519.872 http://doi.org/10.15407/pp2025.03.053

Я. В. Іванчук, О. О. Борисюк

СИНТЕЗ ЕВОЛЮЦІЙНИХ МЕХАНІЗМІВ В РОЗРОБЦІ АДАПТИВНОГО АЛГОРИТМУ ОПТИМІЗАЦІЇ

У статті розроблений ефективний метод розв'язання оптимізаційних задач на основі синтезу еволюційних механізмів розвитку природи, в основу яких закладені принципи генетичного пошуку найкращих рішень. Проаналізовано еволюційні концепції розвитку біологічних видів таких відомих вчених, як Ч. Дарвін, Ж. Ламарк, Хуго де Фріз, К. Попер, М. Кімура і виділено ключові механізми появи нових індивідів із кращим пристосуванням (найкращі рішення). На основі відомих відповідних концепцій визначено основні положення теорії еволюції і побудовані відповідні моделі, що у свою чергу стали обчислювальними аналогіями для розробки еволюційних методів розв'язання оптимізаційних задач. Розроблено метод оптимізації, який базується на генетичному алгоритмі, в основу якого покладені базові оператори еволюції: репродукція (селекція), схрещування та мутація. Особливістю реалізації запропонованого підходу стала гібридизація класичного генетичного алгоритму з адаптивними механізмами налаштування параметрів та локального покращення рішень. У генетичному алгоритмі було використано оператори репродукції (турнірний та рулетковий відбір), одно- і двоточкового схрещування та мутації. Це дало змогу підвищити ефективність глобального пошуку за показниками збіжності (кількість обчислювальних ітерацій) і точності рішення (середня абсолютна похибка розв'язку у разі ста запусків), а також уникати зупинки обчислювального процесу в локальних екстремумах. Розроблено генетичний алгоритм, який містить усі механізми еволюційного обчислення. На базі генетичного алгоритму виконано розрахунок у середовищі Python модельних задач багатокритеріальної оптимізації в бінарному кодуванні оптимального рішення (OneMax, LeadingOnes) і в кодуванні рішення за допомогою дійсних чисел (двоекстремальна функція). У відповідних тестових задачах було зафіксоване стабільне досягнення глобального екстремуму цільової функції і стійкість алгоритму. Це дозволяє зробити висновок про ефективність використання запропонованого методу оптимізації багатокритеріальних функцій на основі генетичного методу.

Ключові слова: цільова функція, генетичний алгоритм, мутація, рекомбінація, пристосованість, хромосома, покоління.

Y. V. Ivanchuk, O. O. Borysuk

SYNTHESIS OF EVOLUTIONARY MECHANISMS IN THE DEVELOPMENT OF AN ADAPTIVE OPTIMISATION ALGORITHM

The article develops an effective method for solving optimization problems based on the synthesis of evolutionary mechanisms of nature's development, which are based on the principles of genetic search for the best solutions. It analyzes evolutionary concepts of biological species development by such well-known scientists as Ch. Darwin, J. Lamarck, Hugo de Vries, K. Popper, and M. Kimura. The key mechanisms for the emergence of new individuals with better adaptation (the best solutions) are identified. Based on the known relevant concepts, the main provisions of the theory of evolution are developed. Corresponding models are constructed, which in turn become computational analogies for the development of evolutionary methods for solving optimization problems. An optimization method has been developed based on a genetic algorithm, which is based on the basic operators of evolution: reproduction (selection), crossover, and mutation. A distinctive feature of the proposed approach is the hybridization of the classical genetic algorithm with adaptive mechanisms for parameter tuning and local improvement of solutions. The genetic algorithm used reproduction operators (tournament and roulette wheel selection), single- and double-point crossover, and mutation. This allows for an increase in the efficiency of global search in terms of convergence (number of computational iterations) and solution accuracy (average absolute error of the solution at 100 runs), as well as avoiding the stopping of the computational process at local extrema. Based on the developed optimization method, a genetic algorithm has been created that incorporates all the mechanisms of evolutionary computation. A genetic algorithm has been developed that contains all the mechanisms of evolutionary computation. Based on the genetic algorithm, model problems of multi-criteria optimisation were calculated in Python in binary coding of the optimal solution (OneMax, LeadingOnes) and in coding

of the solution using real numbers (two-extreme function). In the corresponding test problems, the stable achievement of the global extremum of the objective function and the stability of the algorithm were recorded. This allows us to conclude that the proposed method of optimizing multi-criteria functions based on genetic algorithms is effective.
Keywords: objective function, genetic algorithm, mutation, recombination, fitness, chromosome, generation.

Вступ

Однією із сучасних науково-технічних проблем є розробка методів і моделей для ефективного ухвалення рішень у складних інформаційних системах [1]. У зв'язку із відсутністю формального математичного апарату для побудови адекватних моделей виникає необхідність у пошуку, розробці та застосуванні нових перспективних інтелектуальних технологій моделювання та синтезу [2]. Водночас одним із перспективних напрямків стало моделювання процесів, що виникають у живій природі і прикладне застосування його в оптимізації, ухваленні рішень і побудові штучних інтелектуальних систем. Процеси і перетворення, що здійснюються в природних системах, відбуваються аналогічно підходам реалізації рішення оптимізаційних задач (ОЗ). У разі математичної формалізації рішення ОЗ методами, натхненними природними системами, використовується відомий принцип генерування початкової популяції агентів таким чином, щоб використовувалась уся сфера пошуку [3]. На наступному етапі будується цільова функція вирішуваної ОЗ, після чого виконуються специфічні для кожного із застосовуваних методів, натхненних природними системами, оператори пошуку. Даний підхід дозволяє досліджувати повний спектр альтернативних рішень для отримання ефективних результатів пошуку [4]. Перевагою цих методів є їхня модульна структура, що дозволяє використовувати їхні переваги та недоліки для проєктування нових варіацій алгоритмів, а також проводити їх комбінування, інтеграцію та гібридизацію. На основі проведених досліджень, відповідні методи демонструють ефективну роботу у обробці великих масивів даних [5, 6]. Тому актуальним є дослідження, спрямоване на розробку та моделювання природозумовлених методів оптимізації, в основу яких покладені еволюційні механізми пошуку і аналізу найкращих рішень (індивідів), що дозволить реалізовувати ефективне практичне впровадження в задачах високої обчислювальної складності.

Розробка інтелектуальних інформаційних систем (ІІС) є сучасним напрямком, що найбільш інтенсивно розвивається. Його основною функціональною частиною є методи комп'ютерного моделювання процесів, що протікають у живій природі. У науковій праці [7] показано, що нова ІІС проходить процес самоорганізації на основі шляхів еволюції. Тобто її побудова зводиться до відтворення собі подібних систем за сприятливих умов зовнішнього середовища, які мають необхідні властивості, котрих спочатку не було. Вона характеризується багаторівневістю, єдністю та несуперечливістю.

В еволюційній багатозадачній оптимізації взаємозв'язок між завданнями в основному визначається динамічним процесом через еволюцію популяції. Проте залишається маловивченим питання, яким чином передавати дані між завданнями відповідно до різних ступенів взаємозв'язку, і в міру просування пошуку. Для вирішення цієї проблеми та подальшого підвищення ефективності оптимізації відповідних задач запропонована стратегія гібридного перенесення знань [8]. Зокрема, на основі розподілу популяції здійснюється оцінка взаємозв'язку завдань, після чого застосовується механізм багатознакового перенесення значень для реалізації декількох стратегій перенесення знань.

Для розробки і вдосконолення конфігурації програмних продуктів зазвичай спільно працюють кілька команд фахівців розробників. Дуже часто наявність декількох підпроєктів призводить до параметричних невідповідностей, які заважають і затримують випуск кінцевого програмного

продукту. У статті [9] запропонувано, застосувати еволюційні обчислення (ЕО), зокрема, генетичні алгоритми для вирішення невідповідностей у маштабних програмних продуктах. Еволюційний підхід пошуку і відбору оптимальних рішень на сьогоднішні служить відомою й ефективною методологією для розв'язання різних типів ОЗ. Тому розробка ефективних адаптивних алгоритмів як інструменту оптимізаційних методів, що можуть реалізовуватись як послідовними, так і паралельними перетвореннями є актуальною задачею.

Еталонний аналіз є основою аналізу та розробки обчислювальних методів, особливо в галузі ЕО, де теоретичний аналіз алгоритмів є практично неможливим. У статті [10] показано, що деякі з часто використовуваних еталонних функцій мають свої відповідні оптимуми в центрі множини можливих рішень, а це створює серйозну проблему для аналізу методів ЕО. На основі аналізу семи еволюційних методів було визначено специфічний оператор центрування, який дозволяє ефективно знаходити оптимум у центрі набору еталонних функцій.

Генетичні алгоритми (ГА) є досить ефективним інструментом пошуку розв'язків задач оптимізації. Дія ГА базується на випадковому процесі формування популяції можливих розв'язків, серед яких відбираються найкращі. У статті [11] розроблена операторна модель ГА, яка застосовується до навчання нейронної мережі. Операторний ГА базується на застосуванні інволютивних операторів, що діють у декартовому добутку двох екземплярів *n*-вимірного евклідового простору і здійснюють операції кросоверу та мутації.

Використання стандартного ГА стикається з проблемою, пов'язаною з різною кількістю елементарних дій із сортування, що призводить до використання хромосом різної довжини. Для рішення цієї проблеми запропоновано представлення хромосоми у формі бінарного дерева [12]. Для формування алгоритму, який гарантовано буде сортувати масив, усі кінцеві вузли включають ген фінального сортування у кінець початкової послідовності генів. Для декодування і формування відповідного алгоритму сортування була виконана лінеріалізація: формування текстового представлення за алгоритмом обходу дерева у глибину.

Огляд сучасних наукових джерел свідчить, що залишаються недостатньо дослідженими окремі аспекти, критично важливі для створення ефективних методів розв'язання ОЗ. Зокрема, досі не розроблено узагальнених підходів до адаптивної передачі даних між завданнями з урахуванням їхнього взаємозв'язку та динаміки пошуку. А існуючі гібридні стратегії перенесення знань не забезпечують стійкої роботи алгоритмів у змінних умовах середовища. Недостатньо дослідженими залишаються також методи автоматичного налаштування параметрів ЕА, що впливають на їхню здатність уникати передчасної збіжності до локальних екстремумів. Крім того, наявні операторні моделі та структури хромосом є вузькоспеціалізованими й не гарантують універсальності пошуку.

Метою роботи було розроблення ефективного методу розв'язання ОЗ на основі синтезу еволюційних механізмів розвитку природи, в основу яких покладено принципи генетичного пошуку найкращих рішень, що дало змогу уникнути зупинки обчислювального процесу в локальних екстремумах.

Для досягнення поставленої мети вирішувалися наступні **задачі:** проведення аналізу відомих моделей еволюції і визначення основних механізмів, що можуть впливати на збіжність рішення в ОЗ; на основі відомих еволюційних моделей розроблено метод оптимізаційного розв'язання ОЗ за допомогою ГА; за допомогою розробленого еволюційного методу розв'язання ОЗ на базі ГА виконано розрахунок модельних задач багатокритеріальної оптимізації і здійснити порівняльний аналіз отриманих результатів.

Матеріали та методи дослідження. У дослідженні використано сукупність теоретичних і прикладних методів, що базуються на концепціях ЕО, зокрема ГА, а також сучасних підходах до побудови адаптивних оптимізаційних процедур. Основу розробленого методу становить ГА, який реалізує пошук оптимального рішення за допомогою базових операторів еволюції.

Для валідації запропонованого методу були відібрані типові оптимізаційні задачі із бінарним кодуванням рішення та із кодуванням у вигляді дійсних чисел. Реалізація ГА і проведення обчислювальних експериментів здійснювалися в середовищі Python. Основні показники, що використовувалися для оцінювання ефективності алгоритму: якість отриманого рішення (значення цільової функції); швидкість збіжності (кількість ітерацій до досягнення критерію зупину); стійкість алгоритму (відхилення результатів за умови багатократного запуску).

Результати дослідження. Основою сучасного розвитку штучних інтелектуальних систем є результатом синтезу різних концепцій теоретичних основ еволюції та генної інженерії [13]. Базова модель еволюції Ч. Дарвіна може бути описана як поступовий багатоступеневий процес відтворення нащадків, що реалізує механізм «виживання найсильніших» (Рис. 1).

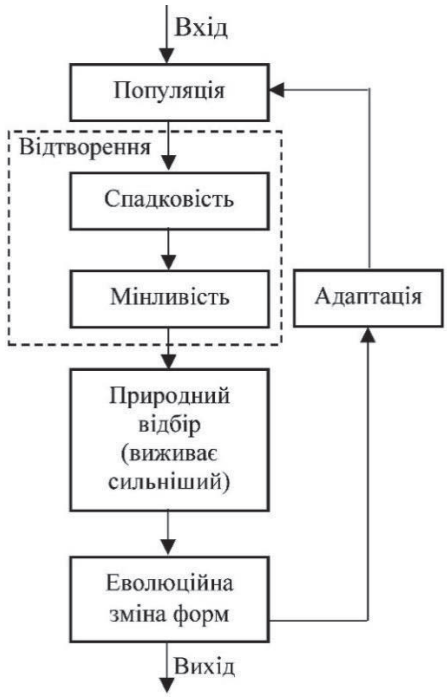


Рис. 1. Схема моделі еволюції Ч. Дарвіна

У процесі опису представленої схеми може бути сформований алгоритм вирішення ОЗ, а саме: у першому блоці міститься популяція (на першому етапі створюється популяція альтернативних рішень вихідної ОЗ і задається кількість поколінь еволюції), потім реалізується блок відтворення на основі невизначеної спадкової мінливості з урахуванням впливу зовнішнього середовища (на другому етапі реалізується процес відтворення нащадків (нових альтернативних рішень поставленої задачі). Далі реалізується блок природного відбору, заснований на механізмі «виживають найсильніші» (на третьому етапі відбираються кращі альтернативні рішення згідно з принципом Ч. Дарвіна «виживають найсильніші»). І в останньому блоці відбувається еволюційна зміна форм, тобто кращі рішення, які виживають, стають домінуючими в даній популяції («кращі» рішення замінюють «гірші» на основі заданого критерію оптимізації для створення нової популяції і продовження пошуку кращих рішень).

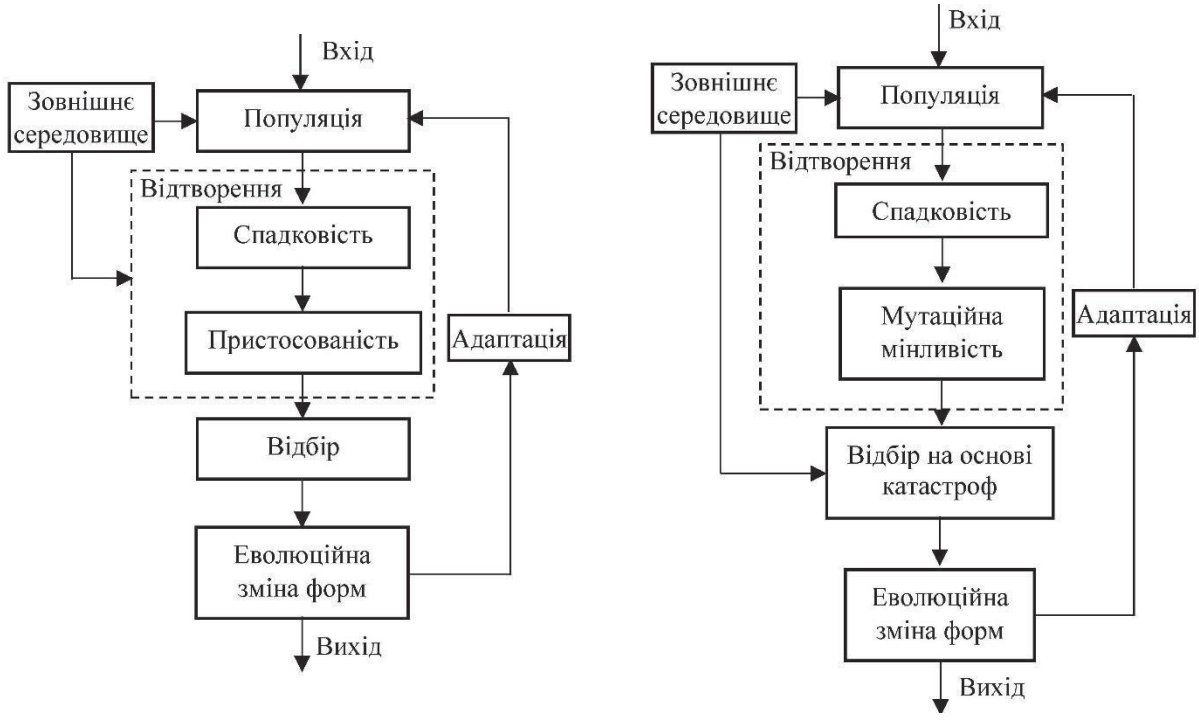
1886 року голландським ученим Хуго де Фрізом була запропонована модель еволюції у вигляді стрибкоподібного процесу за рахунок мутаційної мінливості організмів [4]. Дана модель підтверджує палеонтологічну теорію соціальних і географічних катастроф, що призводять до різкої зміни видів й популяцій і проявляються усього кілька разів за мільйони поколінь. У даній моделі нові види в еволюції виникають не через поступове накопичення безперервних випадкових змін [14], а шляхом раптової появи різких стрибків у розвитку, названих мутацією, що приводило до зміни ознак і перетворювало один вид на інший (рис. 2, б).

На основі представленої схеми може бути сформований наступний алгоритм вирішення ОЗ, а саме: у першому блоці, як і в попередніх моделях, міститься популяція (створюється популяція альтернативних рішень вихідної ОЗ і задається кількість поколінь еволюції); на відміну від інших моделей, тут реалізується блок відтворення з урахуванням мутаційної мінливості (відтворення нових альтернативних рішень поставленої задачі з урахуванням застосування мутаційних перетворень). Далі реалізується відбір організмів, після наслідків прояву катастроф під впливом зовнішнього середовища, що призводить до скорочення популяції (з урахуванням зміни технічного за-

вдання (катастрофа), тобто під впливом зовнішнього середовища здійснюється відбір на основі заданого критерію оптимізації для створення нової популяції, зміни параметрів і режимів пошуку). І в останньому блоці відбувається еволюційна зміна форм, де рішення, що залишилися, складуть нову популяцію з новими якостями.

Австрійський філософ і соціолог К. Р. Поппер 1972 року запропонував еволюційну модель [15], яка описує процес, що реалізує ієрархічну систему гнучких механізмів управління, в яких мутація інтерпретується як метод випадкових проб і помилок, а відбір є одним із способів управління для усунення помилок у взаємодії із зовнішнім середовищем (рис. 3, а). На базі цієї схеми може бути сформований наступний алгоритм вирішення ОЗ, а саме [16]: у першому блоці формується популяція пробних рішень (створюється пробна популяція альтернативних рішень вихідної ОЗ, задається кількість поколінь еволюції та параметрів пошуку). На наступному етапі реалізується блок відтворення на основі спадковості та мутаційної мінливості з урахуванням прямого і зворотного впливу зовнішнього середовища (реалізується процес відтворення нащадків (нових альтернативних рішень поставленої задачі) на основі схрещування і мутаційних перетворень). Далі реалізується блок відбору під управлінням зовнішнього середовища (відбираються альтернативні рішення згідно з критерієм оптимізації). І в останньому блоці відбувається еволюційна зміна форм, тобто створюється нова проблема (популяція), яка і бере участь у подальшому еволюційному процесі (створюється нова популяція для продовження пошуку).

1968 року японський біолог Мотоо Кімура запропонував модель еволюції [17] в якій переважна більшість мутацій на молекулярному рівні має нейтральний характер відносно природного відбору. Тому мінливість організмів (особливо в малих популяціях) пояснюється не дією відбору, а випадковими мутаціями, які є нейтральними або майже нейтральними. Опис представленої схеми (рис. 3, б) може слугувати у формуванні наступного алгоритму вирішення ОЗ, а саме [17, 18]: у першому блоці формується початкова популяція (створюється популяція альтернативних рішень ви-



а) б)

Рис. 2. Схеми моделі еволюції:
а – модель Ж. Ламарка; б – модель Хуго де Фріза

хідної ОЗ і задається кількість поколінь еволюції та параметрів пошуку).

На наступному етапі реалізується блок відтворення на основі спадковості і мутаційної мінливості, а далі реалізується блок нейтрального відбору, в якому відбирається рівно половина особин з кращими і гіршими або середніми ознаками (реалізується процес відтворення нащадків (нових альтернативних рішень поставленої задачі) на основі схрещування і мутаційних перетворень). І в останньому блоці, як завжди, відбувається еволюційна зміна форм, тобто створюється нова популяція, яка і бере участь у подальшому еволюційному процесі (з відібраних індивідів створюється нова популяція альтернативних рішень для продовження пошуку).

Аналізуючи представлені еволюційні моделі еволюції (див. рис. 1–3) можна представити основні положення теорії еволюції й підходу відповідного методу обчислення в ОЗ:

1) Основною елементарною одиницею еволюції вважається локальна популяція (рішення);

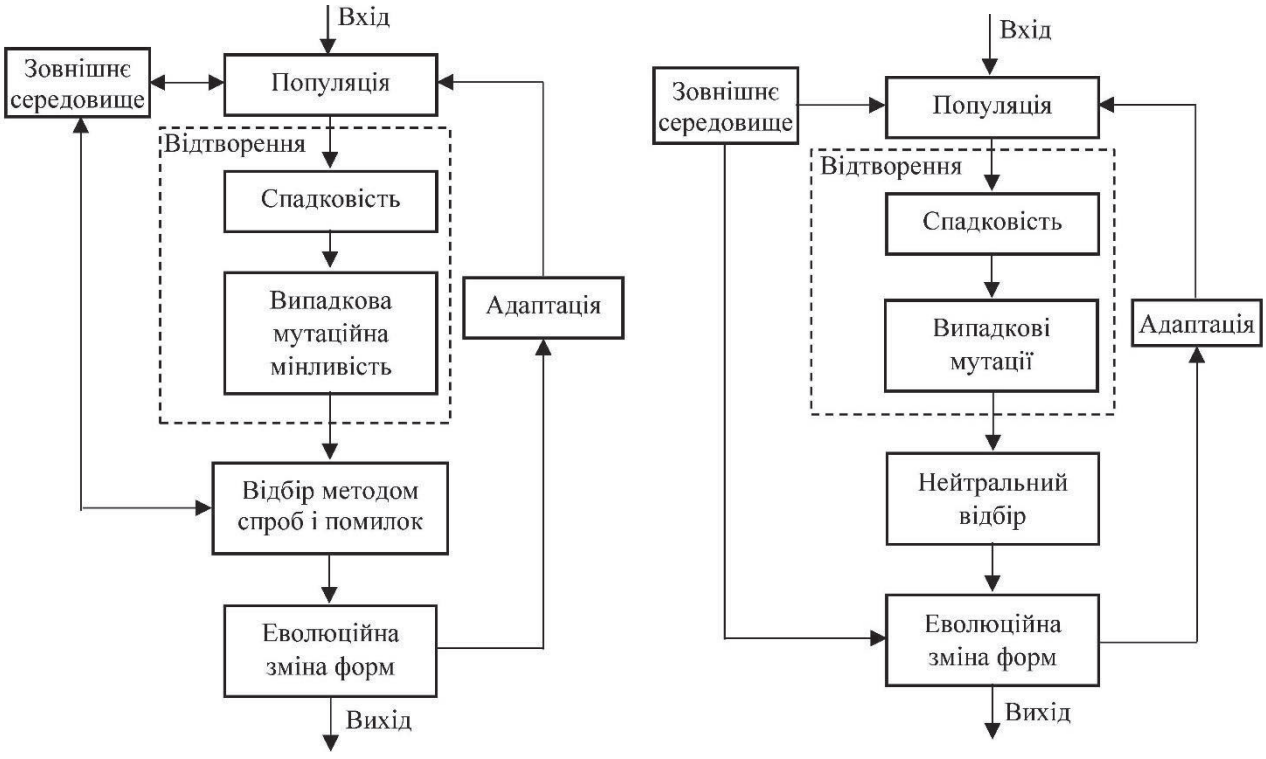
2) Основним матеріалом для еволюції слугує мутаційна та рекомбінаційна мінливість (генетичні обчислювальні оператори);

3) Природний відбір є основною причиною виникнення та розвитку адаптацій, а також створює складні генетичні системи (визначення рішення із максимальним (мінімальним) значенням цільової функції);

4) Випадкові мутаційні та генетичні зміни, що відбуваються в невеликій популяції, є причинами формування нейтральних ознак (утворення нових рішень, відмінних від початкових значень цільової функції);

5) Поява нових видів здійснюється переважно в умовах географічної ізоляції (локалізація рішень у визначеній розрахунковій області допустимих значень).

Під час розв'язання різних ОЗ ефективно використовують стратегії, концепції, методи, механізми еволюційного моделювання [18]. Одним з основних методів еволюційного моделювання є ГА [19, 20] метою якого є пошук оптимального розв'язку деякої задачі. Оскільки ГА імітує в своїй



а) б)

Рис. 3. Модифіковані схеми моделі еволюції:

а – модель К. Попера; б – модель М. Кімури

роботі природні способи оптимізації генетичного успадкування та природного відбору, то під час опису ГА використовуються визначення, запозичені із генетики.

Цільова функція $f(x)$ еквівалентна природному поняттю пристосованості живого організму. Вектор параметрів $x=(x_1, x_2, \ldots, x_n)$ цільової функції називається фенотипом, а окремі його параметри x_i ознаками, $i=1, \ldots, n$.

Кожна координата x_i вектора $x=(x_1, x_2, \ldots, x_n)\in D$ представляється в певній формі S_i, що найкраще підходить для використання в ГА залежно від типу ОЗ, і називається геном. Для цього необхідно перетворити, у загальному випадку на взаємно однозначне, вектор параметрів $x=(x_1, x_2, \ldots, x_n)\in D$ в структуру $S=(s_1, s_2, \ldots, s_n)\in S$, що називається хромосомою (генотипом, індивід):

$$D \xrightarrow{e^{-1}} S, \quad (1)$$

де e^{-1} – функція декодування.

Окремі гени хромосоми являють собою унікальні незалежні зміні при рішенні різного типу ОЗ. Параметри можуть бути групою бітів, змінними із плаваючою точкою або простими двозначними числами у двійковому коді.

У просторі представлень S вводиться функція пристосованості $\mu(s)$:

$$S \xrightarrow{\mu} R, \quad (2)$$

де R – множина дійсних чисел, аналогічна цільовій функції $f(x)$ на множині D. Функцією $\mu(s)$ може бути будь-яка функція, що задовольняє наступну умову:

$$\forall x^1, x^2 \in D : s^1 = e(x^1), s^2 = e(x^2), s^1 \neq s^2, \quad (3)$$

якщо $f(x^1) > f(x^2)$, то $\mu(s^1) > \mu(s^2)$.

Розв'язок вихідної ОЗ $f(x^*) = \max(\min)\ f(x)$, зводиться до пошуку рішення s^* іншої задачі оптимізації: $\mu(s^*) = \max(\min)\ \mu(s)$.

В процесі розв'язку використовуються кінцеві набори можливих рішень, які називаються популяціями:

$$I_q = \left\{ s^k = \left(s_1^k, s_2^k, \ldots, s_n^k\right), k = 1, 2, \ldots, N \right\} \in S, \quad (4)$$

де s^k – хромосома із номером k, N – розмір популяції, s_i^k – ген із номером i, q – значення номера покоління.

Після чого здійснюється зворотне перетворення:

$$x^* = e^{-1}(s^*). \quad (5)$$

Існують різні способи задання функції пристосованості:

1) як правило покладають $\mu(s)=f(e^{-1}(s))$;

2) у деяких випадках необхідно, щоб функція $\mu(s)$ приймала тільки додатні значення. Тоді функція пристосованості може бути представлена, як:

$$\mu(s^k) = f(s^k) + \left| \min_{k=1,2,\ldots,m} f(s^k) \right| + 1 \quad (6)$$

або $\mu(s^k) = \left(f(s^k) - \left| \min_{k=1,2,\ldots,m} f(s^k) \right| + 1 \right)^2$.

На кожному етапі генерації ГА хромосоми є результатом застосування генетичних операторів: репродукція (селекція), схрещування і мутація [20]. На рисунку 4 представлена схема ГА, яка поєднала усі особливості розглянутих моделей еволюції (див. рис. 1-3) і дозволила розробити ефективний механізм евристичного розв'язання ОЗ.

Крок 1. Початок роботи програми ГА.

Крок 2. Введення цільової функції $f(x)$ як форми пристосованості індивіда; p_1, p_2 – ймовірності схрещування і мутації відповідно; N – розміру популяції; n – довжини хромосоми; Q – загальна кількість ітерацій (поколінь).

Крок 3. Ініціалізація лічильника значення номера поточного покоління q.

Крок 4. Створення початкової популяції з N хромосом (індивідів). Для цього випадковим чином генерується кінцевий набір пробних рішень:

$$I_q = \{ s^k = (s_1^k, s_2^k, \ldots, s_n^k), k = 1, 2, \ldots, N \} \in S, \quad (7)$$

де S – множина дійсних або бінарних чисел, $q=0$ – нульове покоління.

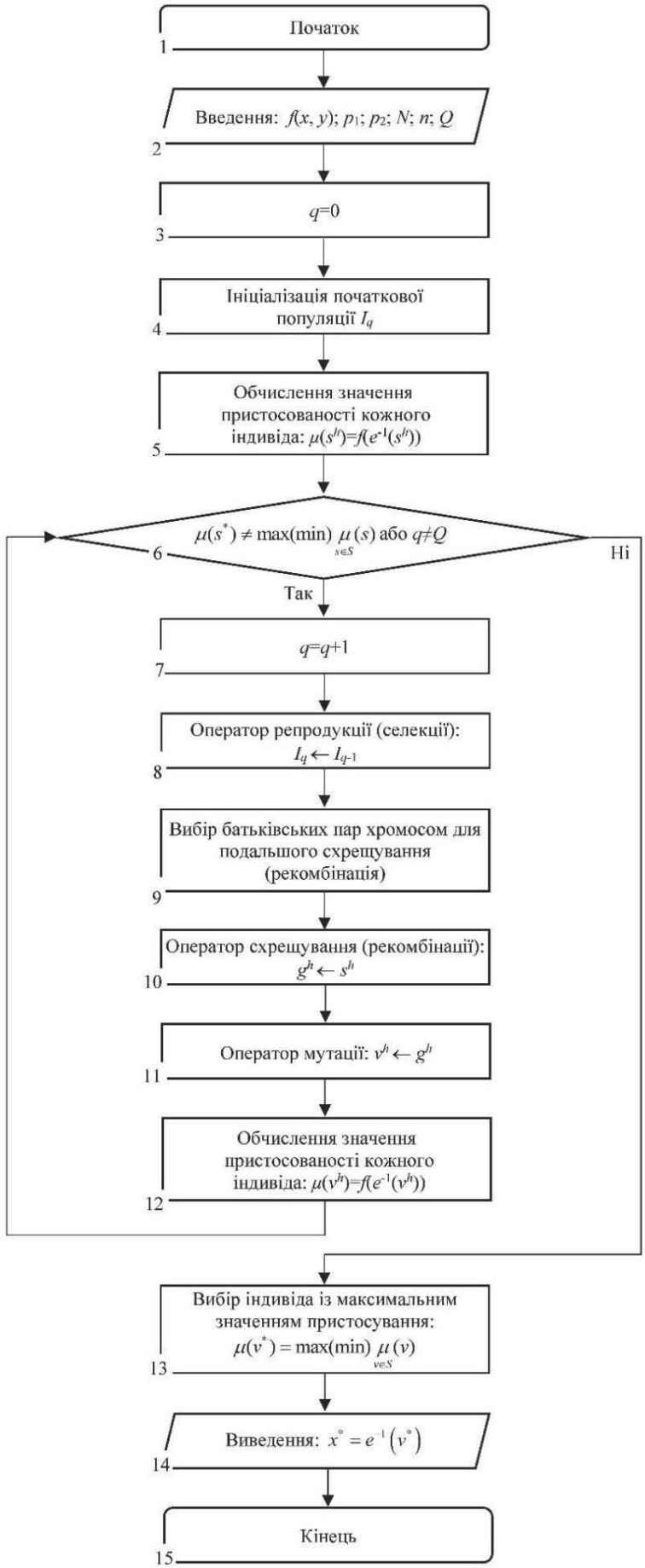


Рис. 4. Схема генетичного алгоритму

Крок 5. Оцінка ступіню пристосованості кожного індивіда – обчислення значення цільової функції $\mu(s^h)=f(e^{-1}(s^h))$.

Крок 6. Ініціалізація умови роботи циклу, а саме: виконання обчислень доти, поки значення пристосованості (цільової функції) індивіда s^* не буде задовольняти умову $\mu(s^*) = \max(\min)\ \mu(s)$ або значення $s \in S$ покоління $q=Q$, де Q –загальне число ітерацій обчислення ГА.

Крок 7. Збільшення поточного значення номера покоління, а саме: $q=q+1$.

Крок 8. Застосування оператора репродукції (селекції). Використовуючи методи рулетки, ранжування або турнірної селекції вибрати N батьківських хромосом із попередньої популяції індивідів: $I_q \leftarrow I_{q-1}$, де $I_{q,\,q-1} \in S$.

Крок 9. Виконання вибору батьківських пар хромосом для подальшого схрещування (рекомбінація), використовуючи оператори панміксії, інбридингу або аутбридингу.

Крок 10. Застосування оператора схрещування (рекомбінація). Використання таких методів рекомбінації, як кросингування або арифметичне і проміжне схрещування з імовірністю p_1, отримати популяцію нащадків індивідів на основі батьківських хромосом:

$$(g_1^h, g_2^h, \ldots, g_k^h) \leftarrow (s_1^h, s_2^h, \ldots, s_k^h). \quad (8)$$

Крок 11. Застосування оператору мутації. З імовірністю p_2 піддати мутації популяцію нащадків індивідів:

$$(v_1^h, v_2^h, \ldots, v_k^h) \leftarrow (g_1^h, g_2^h, \ldots, g_k^h). \quad (9)$$

Крок 12. Оцінка ступеню пристосованості кожного індивіда в новій популяції мутованих нащадків – обчислення відповідного значення цільової функції $\mu(v^h)=f(e^{-1}(v^h))$.

Крок 13. Якщо виконується умова зупинки на кроці 6, тоді необхідно здійснити вибір індивіда із максимальним значенням пристосування:

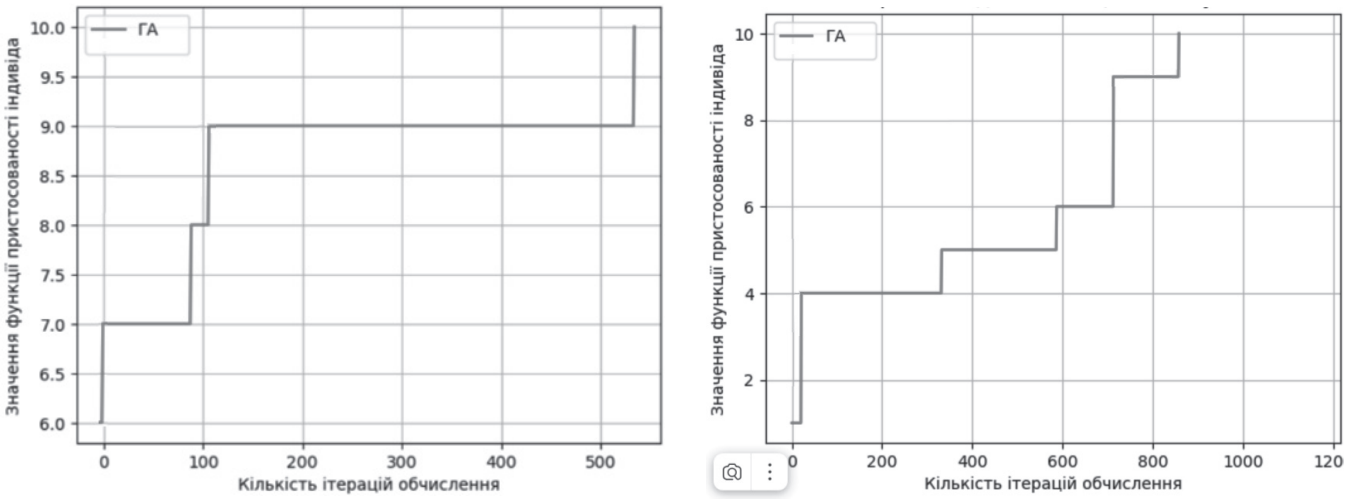
$$v^{h*} = \left(v_1^{h*}, v_2^{h*}, \ldots, v_k^{h*}\right), \quad (10)$$

де $\mu(v^*) = \max(\min)\ \mu(v)$.

Крок 14. Здійснення виведення результатів обчислень $x^* = e^{-1}(v^*)$.

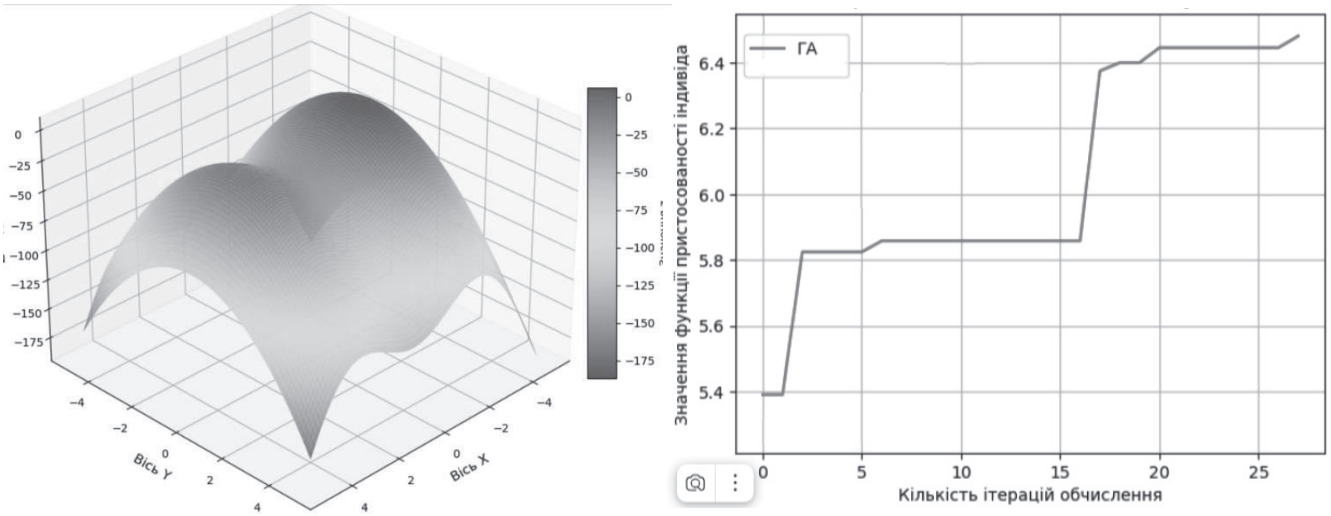
Крок 15. Кінець роботи програми ГА.

Оскільки ГА належить до числа евристичних методів, у нього відсутня точна умова збіжності розв'язку. Але використання стандартних класичних генетичних операторів дає змогу взяти до уваги всі наявні теоретичні обґрунтування та експери-



а) б)

Рис. 5. Діаграма процесу розв'язання ГА модельних задач:
а – OneMax; б – LeadingOnes



а) б)

Рис. 6. Діаграма двоекстремальної функції (а) та результату її розв'язання ГА (б):
а – тривимірний графік функції; б – діаграма зміни значення функції пристосованості

ментальні підтвердження ефективності ГА [13, 20].

Для проведення оцінки ефективності ГА у розв'язанні ОЗ, було обрано два типи модельних задач, а саме в бінарному представленні індивіда (рішення) і у формі дійсних чисел. Зокрема, для бінарного представлення множини можливих рішень, використано модельну задачу оптимізації OneMax і LengthMax [21, 22].

Під час аналізу результатів обчислення (рис. 5), було відмічено стабільне досягнення глобального оптимуму для популяції із чотирьох індивідів, зокрема: в задачі OneMax було виконано 558 ітерацій, а в задачі LengthMax - 859 ітерацій.

Для тестування ГА, де використовується популяція (рішення) із кодуванням у дійсній формі [10, 23], було обрано оптимізаційну задачу на основі двоекстремальної цільової функції (рис. 6, а). Зокрема, в оптимізаційній задачі на основі цільової двоекстремальної функції. Точкою глобального екстремуму відповідної цільової функції $f(x, y)= -3x^2-4y^2-23\cos(x-0,5)$ є точка $f(x^*, y^*)=6.4892$, де $x^*= -2,0709$ і $y^*=0$, а точкою локального максимуму $f(x^*, y^*)= -8.193$, де $x^*= 2,8164$ і $y^*=0$ (див. рис. 6, а). Аналізуючи результати обчислення (рис. 6, б) можна відмітити стабільне досягнення глобального оптимуму без зупинки у локальному екстремуму [24].

Аналізуючи результати обчислення на рисунку 6 можна відмітити стабільне досягнення глобального оптимуму без зупинки у локальному екстремумі для популяції із шести індивідів (було виконано 28 ітерацій).

Висновки

У статті проведено аналіз відомих еволюційних моделей і визначено основні концепції і підходи еволюційного обчислення ОЗ. Зокрема, розроблено метод оптимізації на основі ГА за допомогою базових операторів еволюції: репродукція (селекція), схрещування та мутації. Особливістю реалізації запропонованого підходу є гібридизація класичного ГА з адаптивними механізмами налаштування параметрів та локального покращення рішень, що дозволяє підвищити ефективність глобального пошуку. За допомогою розробленого еволюційного методу розв'язання ОЗ на базі ГА виконано розрахунок модельних задач багатокритеріальної оптимізації. Зокрема, в задачах OneMax і LeadingOnes в бінарному кодуванні оптимального рішення було визначено, що швидкість збіжності рішення у середньому склала 708 ітерацій, а у задачі пошуку глобального екстремуму двоекст-

ремальної функції склала 28 ітерацій. Більше того, у відповідних тестових задачах було зафіксоване стабільне досягнення глобального екстремуму цільової функції, а середня абсолютна похибка розв'язку у разі ста запусків склала у середньому $2{,}8 \cdot 10^{-3}$.

Література

1. Saif R., Nadeem S., Khaliq A., Zia S., Iftekhar A. Mathematical understanding of sequence alignment and phylogenetic algorithms: a comprehensive review of computation of different methods // *Advancements in Life Sciences*. – 2022. – Vol. 9, № 4. – P. 401–411.
2. Kudela J. A critical problem in benchmarking and analysis of evolutionary computation methods // *Nature Machine Intelligence*. – 2022. – Vol. 4, № 10. – P. 1238–1245. – DOI: 10.1038/s42256-022-00579-0.
3. Karim R. A. H., Vassányi I., Kósa I. After-meal blood glucose level prediction using an absorption model for neural network training // *Computers in Biology and Medicine*. – 2020. – Vol. 125. – Article no. 103956. – DOI: 10.1016/j.compbiomed.2020.103956.
4. Doerr B., Neumann F. A survey on recent progress in the theory of evolutionary algorithms for discrete optimization // *ACM Transactions on Evolutionary Learning and Optimization*. – 2021. – Vol. 1, № 4. – P. 16. – DOI: 10.1145/3472304.
5. Afzal U., Mahmood T., Usmani S. Evolutionary computing to solve product inconsistencies in software product lines // *Science of Computer Programming*. – 2022. – Vol. 224. – Article no. 102875. – DOI: 10.1016/j.scico.2022.102875.
6. Xu Y., Zhang H., Zeng X., Nojima Y. An adaptive convergence enhanced evolutionary algorithm for many-objective optimization problems // *Swarm and Evolutionary Computation*. – 2022. – Vol. 75. – Article no. 101180. – DOI: 10.1016/j.swevo.2022.101180.
7. Thang T. B., Dao T. C., Long N., Binh H. Parameter adaptation in multifactorial evolutionary algorithm for many-task optimization // *Memetic Computing*. – 2021. – Vol. 13, № 4. – DOI: 10.1007/s12293-021-00347-4.
8. Cai Y., Peng D., Liu P., Guo J.-M. Evolutionary multi-task optimization with hybrid knowledge transfer strategy // *Information Sciences*. – 2021. – Vol. 580. – P. 874–896. – DOI: 10.1016/j.ins.2021.09.021.
9. Agrawal S., Tiwari A., Naik P., Srivastava A. Improved differential evolution based on multi-armed bandit for multimodal optimization problems // *Applied Intelligence*. – 2021. – Vol. 51, № 10. – P. 7625–7646. – DOI: 10.1007/s10489-021-02261-1.
10. Ma X., Yang J., Sun H., Hu Z., Wei L. Feature information prediction algorithm for dynamic multi-objective optimization problems // *European Journal of Operational Research*. – 2021. – Vol. 295, № 3. – P. 965–981. – DOI: 10.1016/j.ejor.2021.01.028.
11. Олійник Л. О. Операторний генетичний алгоритм і навчання нейронної мережі // *Computer Science and Applied Mathematics*. – 2023. – № 2. – С. 52–58. – DOI: 10.26661/2786-6254-2023-2-07.
12. Shinkarenko V. I., Makarov O. V. Genetic algorithm for structural adaptation of sorting algorithms // *Problems in Programming*. – 2024. – № 2–3. – P. 11–18. – DOI: 10.15407/pp2024.02-03.011.
13. López-Ibáñez M., Branke J., Paquete L. Reproducibility in evolutionary computation // *ACM Transactions on Evolutionary Learning and Optimization*. – 2021. – Vol. 1, № 4. – Article no. 13. – DOI: 10.1145/3466624.
14. Li W., Liu K., Guo Q., Zhang Z., Ji Q., Wu Z. Genetic algorithm-based optimization of curved-tube nozzle parameters for rotating spinning // *Frontiers in Bioengineering and Biotechnology*. – 2021. – Vol. 9. – Article no. 781614. – DOI: 10.3389/fbioe.2021.781614.
15. Liang Y. Y., Shen J. C., Li W. Evolution of compressive mechanical properties of early hypertrophic scar during laser treatment // *Journal of Biomechanics*. – 2021. – Vol. 129. – Article no. 110783. – DOI: 10.1016/j.jbiomech.2021.110783.
16. Holmes A., Darbyshire T., Brennan M., McTierney S., Small A. The genome sequence of *Gari tellinella* (Lamarck, 1818), a sunset clam // *Wellcome Open Research*. – 2022. – Vol. 7. – Article no. 116. – DOI: 10.12688/wellcomeopenres.17805.1.
17. Khan A., Burmeister A. R., Wahl L. M. Evolution along the parasitism-mutualism continuum determines the genetic repertoire of prophages // *PLoS Computational Biology*. – 2020. – Vol. 16, № 12. – Article no. 1008482. – DOI: 10.1371/journal.pcbi.1008482.
18. Iskovych-Lototsky R. D., Ivanchuk Y. V., Veselovsky Y. P., Gromaszek K., Oralbekova A. Automatic system for modeling of working processes in pressure generators of hydraulic vibrating and vibro-impact machines // *Proceedings of SPIE 10808, Photonics Applications in Astronomy, Communications, Indus-*

try, and High-Energy Physics Experiments 2018. – 2018. – Article no. 1080850. – DOI: 10.1117/12.2501532.

19. Kvyetnyy R., Ivanchuk Y. *Computational Methods and Algorithms*: [Textbook]. – Vinnytsya: VNTU, 2024.
20. Alam M., Samad M. D., Vidyaratne L., Glandon A., Iftekharuddin K. M. Survey on deep neural networks in speech and vision systems // *Neurocomputing*. – 2020. – Vol. 417. – P. 302–321. – DOI: 10.1016/j.neucom.2020.07.053.
21. Kvyetnyy R., Ivanchuk Y., Yarovyi A., Horobets Y. Algorithm for increasing the stability level of cryptosystems // *Selected Papers of the VIII International Scientific Conference "Information Technology and Implementation (IT&I-2021)"*. – 2021. – Vol. 3179. – P. 293–301. – CEUR Workshop Proceedings. – URL: https://ceur-ws.org/Vol-3179/Short_2.pdf.
22. Sun X., Wang D., Kang H., Shen Y., Chen Q. A two-stage differential evolution algorithm with mutation strategy combination // *Symmetry*. – 2021. – Vol. 13, № 11. – Article no. 2163. – DOI: 10.3390/sym13112163.
23. Iskovych–Lototsky R. D., Ivanchuk Y. V., Veselovsky Y. P. Simulation of working processes in the pyrolysis plant for waste recycling // *Eastern–European Journal of Enterprise Technologies. Engineering Technological Systems*. – 2016. – Vol. 1, № 8(79). – P. 11–20. – DOI: 10.15587/1729-4061.2016.59419.
24. Czégel D., Giaffar H., Tenenbaum J. B., Szathmáry E. Bayes and Darwin: how replicator populations implement Bayesian computations // *BioEssays*. – 2022. – Vol. 44, № 4. – Article no. 2100255. – DOI: 10.1002/bies.202100255.

References

1. Saif, R., Nadeem, S., Khaliq, A., Zia, S. and Iftekhar, A. (2022) Mathematical Understanding of Sequence Alignment and Phylogenetic Algorithms: A Comprehensive Review of Computation of Different Methods. *Advancements in Life Sciences*, 9(4), pp. 401–411.
2. Kudela, J. (2022) A critical problem in benchmarking and analysis of evolutionary computation methods. *Nature Machine Intelligence*, 4(10), pp. 1238–1245. doi: 10.1038/s42256-022-00579-0.
3. Karim, R.A.H., Vassányi, I. and Kósa, I. (2020) After-meal blood glucose level prediction using an absorption model for neural network training. *Computers in Biology and Medicine*, 125, Article 103956. doi: 10.1016/j.compbiomed.2020.103956.
4. Doerr, B. and Neumann, F. (2021) A Survey on Recent Progress in the Theory of Evolutionary Algorithms for Discrete Optimization. *ACM Transactions on Evolutionary Learning and Optimization*, 1(4), Article 16. doi: 10.1145/3472304.
5. Afzal, U., Mahmood, T. and Usmani, S. (2022) Evolutionary Computing to solve product inconsistencies in Software Product Lines. *Science of Computer Programming*, 224, Article 102875. doi: 10.1016/j.scico.2022.102875.
6. Xu, Y., Zhang, H., Zeng, X. and Nojima, Y. (2022) An adaptive convergence enhanced evolutionary algorithm for many-objective optimization problems. *Swarm and Evolutionary Computation*, 75, Article 101180. doi: 10.1016/j.swevo.2022.101180.
7. Thang, T.B., Dao, T.C., Long, N. and Binh, H. (2021) Parameter adaptation in multifactorial evolutionary algorithm for many-task optimization. *Memetic Computing*, 13(4). doi: 10.1007/s12293-021-00347-4.
8. Cai, Y., Peng, D., Liu, P. and Guo, J.-M. (2021) Evolutionary multi-task optimization with hybrid knowledge transfer strategy. *Information Sciences*, 580, pp. 874–896. doi: 10.1016/j.ins.2021.09.021.
9. Agrawal, S., Tiwari, A., Naik, P. and Srivastava, A. (2021) Improved differential evolution based on multi-armed bandit for multimodal optimization problems. *Applied Intelligence*, 51(10), pp. 7625–7646. doi: 10.1007/s10489-021-02261-1.
10. Ma, X., Yang, J., Sun, H., Hu, Z. and Wei, L. (2021) Feature information prediction algorithm for dynamic multi-objective optimization problems. *European Journal of Operational Research*, 295(3), pp. 965–981. doi: 10.1016/j.ejor.2021.01.028.
11. Oliynyk, L.O. (2023) Operator Genetic Algorithm and Neural Network Training. *Computer Science and Applied Mathematics*, (2), pp. 52–58. doi: 10.26661/2786-6254-2023-2-07.
12. Shinkarenko, V.I. and Makarov, O.V. (2024) Genetic algorithm for structural adaptation of sorting algorithms. *Problems in Programming*, 2–3, pp. 11–18. doi: 10.15407/pp2024.02-03.011.
13. López-Ibáñez, M., Branke, J. and Paquete, L. (2021) Reproducibility in Evolutionary Computation. *ACM Transactions on Evolutionary Learning and Optimization*, 1(4), Article 13. doi: 10.1145/3466624.

14. Li, W., Liu, K., Guo, Q., Zhang, Z., Ji, Q. and Wu, Z. (2021) Genetic Algorithm-Based Optimization of Curved-Tube Nozzle Parameters for Rotating Spinning. *Frontiers in Bioengineering and Biotechnology*, 9, Article 781614. doi: 10.3389/fbioe.2021.781614.
15. Liang, Y.Y., Shen, J.C. and Li, W. (2021) Evolution of compressive mechanical properties of early hypertrophic scar during laser treatment. *Journal of Biomechanics*, 129, Article 110783. doi: 10.1016/j.jbiomech.2021.110783.
16. Holmes, A., Darbyshire, T., Brennan, M., McTierney, S. and Small, A. (2022) The genome sequence of Gari tellinella (Lamarck, 1818), a sunset clam. *Wellcome Open Research*, 7, Article 116. doi: 10.12688/wellcomeopenres.17805.1.
17. Khan, A., Burmeister, A.R. and Wahl, L.M. (2020) Evolution along the parasitism-mutualism continuum determines the genetic repertoire of prophages. *PLoS Computational Biology*, 16(12), Article e1008482. doi: 10.1371/journal.pcbi.1008482.
18. Iskovych-Lototsky, R.D., Ivanchuk, Y.V., Veselovsky, Y.P., and Gromaszek, K. (2018) Automatic system for modeling of working processes in pressure generators of hydraulic vibrating and vibro-impact machines. *Proc. SPIE 10808, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2018*, 1080850. doi: 10.1117/12.2501532.
19. Kvyetnyy, R. and Ivanchuk, Y. (2024) Computational Methods and Algorithms. Vinnytsya: VNTU.
20. Alam, M., Samad, M.D., Vidyaratne, L., Glandon, A. and Iftekharuddin, K.M. (2020) Survey on Deep Neural Networks in Speech and Vision Systems. *Neurocomputing*, 417, pp. 302–321. doi: 10.1016/j.neucom.2020.07.053.
21. Kvyetnyy, R., Ivanchuk, Y., Yarovyi, A. and Horobets, Y. (2021) Algorithm for Increasing the Stability Level of Cryptosystems. *Proceedings of the VIII International Scientific Conference "Information Technology and Implementation (IT&I-2021)*", 3179, pp. 293–301. Available at: https://ceur-ws.org/Vol-3179/Short_2.pdf.
22. Sun, X., Wang, D., Kang, H., Shen, Y. and Chen, Q. (2021) A two-stage differential evolution algorithm with mutation strategy combination. *Symmetry*, 13(11), Article 2163. doi: 10.3390/sym13112163.
23. Iskovych–Lototsky, R.D., Ivanchuk, Y.V. and Veselovsky, Y.P. (2016) Simulation of working processes in the pyrolysis plant for waste recycling. *Eastern-European Journal of Enterprise Technologies: Engineering Technological Systems*, 1(8), pp. 11–20. doi: 10.15587/1729-4061.2016.59419.
24. Czégel, D., Giaffar, H., Tenenbaum, J.B. and Szathmáry, E. (2022) Bayes and Darwin: How replicator populations implement Bayesian computations. *BioEssays*, 44(4), Article 2100255. doi: 10.1002/bies.202100255.

Одержано: 07.10.2025
Внутрішня рецензія отримана: 13.10.2025
Зовнішня рецензія отримана: 17.10.2025

Про авторів:

[1]*Іванчук Ярослав Володимирович*,
доктор технічних наук, професор,
https://orcid.org/0000-0002-4775-6505.

[1]*Олександр Олегович Борисюк*,
аспірант,
https://orcid.org/0009-0004-0244-7700.

Місце роботи авторів:

[1]Вінницький національний технічний університет
тел. +380 (0432) 65-19-03
21021, Україна, м. Вінниця,
вул. Хмельницьке шосе, 95
E-mail: ivanchuck@ukr.net
https://vntu.edu.ua/

УДК 004.8 https://doi.org/10.15407/pp2025.03.066

М. Г. Шевченко, М.В. Андрощук

МУЛЬТИМОДАЛЬНИЙ RAG З ВИКОРИСТАННЯМ ТЕКСТОВИХ ТА ВІЗУАЛЬНИХ ДАНИХ

Стаття присвячена розробці та дослідженню мультимодальної системи генерації, доповненої пошуком (Retrieval-Augmented Generation), призначеної для аналізу та інтерпретації медичних зображень. Об'єктом дослідження є рентгенівські знімки грудної клітки та відповідні їм радіологічні звіти. Основна мета роботи полягала у створенні системи, здатної виконувати два ключові завдання: генерувати детальний радіологічний звіт для вхідного зображення та надавати точні відповіді на конкретні запитання щодо нього. Додатковою ціллю було демонстрування того, що застосування мультимодального підходу з пошуком релевантної інформації суттєво покращує якість генерації порівняно з використанням великих мультимодальних моделей без компонента пошуку. Для реалізації системи було використано комбінацію сучасних моделей глибокого навчання. За основу для створення векторних представлень текстових та візуальних даних було взято модель BiomedCLIP, яку було додатково навчено на цільовому наборі даних. Функції генератора виконувала велика мовна модель LLaVA-Med 1.5, адаптована для медичної галузі та квантизована для роботи в умовах обмежених обчислювальних ресурсів. Архітектура системи також включає допоміжні класифікатори на основі DenseNet121 для визначення проєкції знімка та ідентифікації наявних клінічних ознак, що дозволило підвищити точність пошуку. У процесі експериментального дослідження було протестовано шість різних конфігурацій розробленої системи. Оцінювання проводилося з використанням низки метрик, зокрема, точності та F1 для задачі відповіді на питання, а також BLEU, ROUGE, F1-CheXbert та F1-RadGraph для оцінки якості згенерованих звітів. Результати тестування продемонстрували значну перевагу всіх конфігурацій системи над базовою моделлю-генератором. Найкращі результати показала конфігурація, що використовує класифікатори проєкції та клінічних ознак із вимогою точного збігу знайдених патологій. Дослідження підтвердило, що інтеграція механізму пошуку релевантних даних значно підвищує структурну та змістовну якість генерованих текстових описів для медичних зображень.

Ключові слова: генерація доповнена пошуком, мультимодальність, медичні зображення, генерація звітів, глибинне навчання, великі мовні моделі.

M.H. Shevchenko, M.V. Androshchuk

MULTIMODAL RAG USING TEXT AND VISUAL DATA

This paper presents the development and investigation of a multimodal Retrieval-Augmented Generation system designed for the analysis and interpretation of medical images. The research focuses on chest X-ray images and their corresponding radiology reports. The primary goal was to create a system capable of performing two key tasks: generating a detailed radiology report for an input image and providing accurate answers to specific questions about it. A secondary goal was to demonstrate that employing a multimodal retrieval-augmented approach significantly improves generation quality compared to using large multimodal models without a retrieval component. The system's implementation utilizes a combination of state-of-the-art deep learning models. The BiomedCLIP model, fine-tuned on the target dataset, was used to generate vector embeddings for both text and visual data. The generator component is based on the large language model LLaVA-Med 1.5, which is adapted for the medical domain and quantized to operate under limited computational resources. The system architecture also includes auxiliary classifiers based on DenseNet121 to determine the image projection and identify clinical findings, thereby enhancing retrieval accuracy. The experimental evaluation involved testing six different configurations of the developed system. The evaluation was conducted using a range of metrics, including accuracy and F1-score for the question-answering task, as well as BLEU, ROUGE, F1-CheXbert, and F1-RadGraph for assessing the quality of the generated reports. The test results demonstrated a significant advantage of all system configurations over the baseline generator model. The best results were achieved by the configuration that utilizes projection and clinical finding classifiers with an exact match requirement for the identified pathologies. The study confirmed that integrating a relevant data retrieval mechanism significantly enhances both the structural and semantic quality of the generated textual descriptions for medical images.

Keywords: Retrieval-Augmented Generation, multimodality, medical imaging, report generation, deep learning, large language models.

Вступ

Мультимодальний RAG — це RAG, який використовує не один, а кілька типів даних для ретривера(англ. retriever) й генератора одночасно, наприклад: текст, зображення, відео, аудіо тощо. Застосування кількох типів даних часто надає генератору більше інформації, завдяки чому генеруються вичерпніші та релавантніші відповіді на запит користувача [34].

Мультимодальний RAG із використанням текстових та візуальних даних — це найпоширеніший вид мультимодального RAG. Він застосовує тексти, зображення та відео для генерації відповіді. Найчастіше використовують саме зображення, а не відео, оскільки робота з відео є значно складнішою через великий обсяг, необхідність обробки послідовностей кадрів, а також складність виокремлення релевантної інформації з часових даних. Застосування мультимодального RAG із використанням текстових та візуальних даних часто підвищує коректність і релевантність відповідей генераторів [3], оскільки велика кількість інформації зберігається саме у візуальному вигляді й не має текстового відповідника. Наприклад, така інформація, як просторові відношення між об'єктами, емоційний контекст, специфічні деталі зображень (текстури, кольори, форми), а також складні візуальні структури (графіки, діаграми, медичні знімки), часто не мають точного текстового відповідника або потребують значного спрощення у разі опису словами.

В роботі для реалізації мультимодального RAG було обрано медичну предметну сферу, зокрема, аналізу та інтерпретації рентгенівських знімків грудної клітки та їхніх звітів. Медична предметна сфера є доволі популярним напрямком створення RAG. Вибір такої предметної сфери надає широкі можливості для створення різних RAG систем та дозволить оцінити як використання мультимодального RAG покращує роботу існуючих систем. Іншим фактором вибору такої предметної сфери є достатня кількість доступних наборів даних у вільному доступі, що не завжди властиво іншим предметним сферам. Найчастіше такі набори даних включають певну кількість рентгенівських знімків грудних кліток та звітів у текстовому форматі для кожного знімку.

Мультимодальна RAG-система, створена в роботі, підтримує два сценарії використання. Перший сценарій використання — це генерація звіту на вхідний рентгенівський знімок, щодо наявності паталогій, хвороб тощо. Інший сценарій використання — це генерація відповіді на вхідне питання щодо певного рентгенівського знімку. Ці сценарії є типовими для мультимодального RAG. Також такі сценарії використання є стандартними для використання в медичній сфері. Для реалізації даних сценаріїв використання, ретривер буде знаходити релевантні звіти до вхідного зображення. Потім знайдені звіти будуть разом із вхідним зображенням та питанням надаватися генератору для генерації звіту або відповіді на вхідне питання.

Під час розробки мультимодальної RAG-системи з використанням текстових та візуальних даних для роботи з рентгенівськими знімками грудної клітки та звітами до них були використанні набори даних: MIMIC-CXR [24], MIMIC-CXR-JPG [14], CheXpert [16] та IU-Xray [9].

Методологія дослідження

Рентгенівські знімки грудей бувають двох типів: передні і бічні. Передні — зроблені спереду грудної клітки. Бічні — зроблені збоку грудної клітки. Передні і бічні знімки значно відрізняються один від одного, бо на передніх можна бачити паталогії і клінічні ознаки, яких не видно на бічних і навпаки. Через це було вирішено реалізувати дві моделі, які створюють індекси (далі — моделі індексації): одна модель буде працювати з передніми зображеннями і їхніми звітами, а інша з бічними зображеннями і звітами. Для того, щоб ретривер міг ідентифікувати, яке зображення — переднє, а яке — бічне, було створено додаткову модель для ідентифікації сторони (далі — класифікатор сторін). Також для покращення ретривера, щоб він знаходив

найбільш релевантні звіти, було розроблено модель для створення метаданих, а саме інформацію про наявність певної клінічної ознаки на кожному зображенні (далі — класифікатор хвороб).

Як модель для створення індексів була вибрана модель BiomedClip [33]. BiomedClip — це модель, створена на основі CLIP [25], для задач біомедичного бачення. BiomedClip був натренований на наборі даних PMC-15M, створений дослідниками BiomedClip. PMC-15M складається із 15 мільйонів пар підпис-картинок з різних медичних галузей: офтальмологія, стоматологія, рентгенографія та інші. Також дослідники BiomedClip створили модель PubMedBERT та свій власний Vision Transformer, які використовуються як кодувальник тексту та кодувальник зображень відповідно до архітектури CLIP.

Хоча BiomedClip натренований на медичних даних, він не навчений безпосередньо на тих даних, які планується використовувати в ретривері для релевантного пошуку. Також варто враховувати, що BiomedClip натренований на широкому спектрі медичних галузей, а нам потрібно лише рентгенівські знімки грудей. Враховуючи попередні зауваження, зрозуміло, що просте використання BiomedClip як моделі для створення індексів, може часто призводити до нерелевантних результатів [32].

Для покращення роботи BiomedCLIP застосовують передавальне навчання. У межах цієї роботи BiomedCLIP було донавчено на даних MIMIC-CXR, оскільки саме ці дані надалі використовуються ретривером для релевантного пошуку.

Для донавчання був створений програмний застосунок із використанням бібліотеки PyTorch. Як функція втрат [2] використовується перехрестна втрата ентропії (англ. cross entropy loss) [21] та AdamW [20] як оптимізатор [29].

В результаті було донавчено три окремі моделі BiomedClip.

Перша — для індексації передніх рентгенівських знімків і їхніх звітів (далі — модель індексації передніх знімків). Вона була донавчена на 100000 випадково вибраних пар передніх рентгенгенівських знімків та їхніх звітів з MIMIC-CXR. Модель донавчалася протягом 100 епох [5] на відеокарті Nvidia GeForce RTX 3090 24 ГБ. Для донавчання використовувалися такі гіперпараметри [7]: розмір партії (англ. batch size) [4] — 64, темп навчання (англ. learning rate) [1] — $0.00005\sqrt{8}$. Значення втрат на останній епосі склало: 0.0336.

Друга — для індексації бічних рентгенівських знімків і їхніх звітів (далі — модель індексації бічних знімків). Вона була донавчена на 50000 випадково вибраних пар бічних рентгенгенівських знімків та їхніх звітів з MIMIC-CXR. Модель донавчалася протягом 100 епох на відеокарті Nvidia GeForce RTX 2060 6 ГБ. Для донавчання використовувалися такі гіперпараметри: розмір партії — 8, темп навчання — 0.00005. Значення втрат на останній епосі склало: 0.0306.

Третя — для індексації передніх та бічних рентгенівських знімків і їхніх звітів (далі — загальна модель індексації). Вона була донавчена на 150000 випадково вибраних пар передніх та бічних рентгенгенівських знімків та їхніх звітів з MIMIC-CXR. Модель донавчалася протягом 95 епох на відеокарті Nvidia GeForce RTX 3090 24 ГБ. Для донавчання використовувалися такі гіперпараметри: розмір партії — 64, темп навчання — $0.00005\sqrt{8}$. Значення втрат на останній епосі склало: 0.0337.

Для того, щоб ретривер міг правильно вибирати модель для класифікації передніх чи бічних знімків, він повинен вміти класифікувати зображення на передні і бічні. Це класична задача класифікації зображень [11]. Для такої задачі найчастіше використовуються нейронні мережі. Тренування класифікатора сторін відбувалося на наборі даних CheXpert, бо він містить інформацію про сторону кожного рентгенівського знімка. За нейронну мережу для дотренування була обрана модель DenseNet121, яка також використовувалася в дослідженні набору даних CheXpert і показала кращі результати, ніж інші моделі в дослідженні. DenseNet121 — це згорткова нейронна мережа [13], яка складається з 121 шару, кожен пов'язаний один з одним.

Для тренування використовувалася передавальне навчання. Для цього бралася модель DenseNet121, яка була попередньо натренована на наборі даних ImageNet [10]. Як функція втрат використовувалася перехрестна втрата ентропії та стохастичний градієнтний спуск [27] як оптимізатор.

Тренування класифікатора сторін відбувалося протягом 10 епох на відеокарті Nvidia GeForce RTX 2060 ГБ. Для тренування використовувалося 20000 випадково вибраних рентенівських знімків з набору даних CheXpert. Тестування моделі після кожної епохи відбувалося на 2000 рентгенівських знімках. Уже після першої епохи тренування, точність визначення сторони зображення на тренувальних та тестових даних склала 100%, тому тренування було закінчене достроково на 5-ій епосі. Для донавчання використовувалися такі гіперпараметри: розмір партії — 32, темп навчання — 0.001.

Для покращення релевантності знайдених результатів ретривером було створено класифікатор хвороб. Класифікатор хвороб визначає, які клінічні ознаки із заданого списку наявні на зображенні, а які ні. Це задача класифікації за багатьма класами (англ. multi-label classification) [26], задача в якій один об’єкт може відповідати декільком класам одночасно. Ця задача схожа на класифікацію зображень, використану в класифікаторі сторін. Для її розв’язання також найчастіше використовують нейронні мережі.

Для створення класифікатора хвороб було використано модель DenseNet121, як в класифікаторі сторін. Для тренування було використано набір даних CheXpert, тому що він містить інформацію про наявність певної ознаки зі списку 14 клінічних ознак для кожного зображення. Кожна ознака для кожного зображення в наборі даних CheXpert може мати одне з 4-ох значень: «наявна», «відсутня», «не зазначена», «не зрозуміло». В тренуванні значення «не зазначена» і «не зрозуміло» будуть сприйматися як один клас, тому що важлива лише наявність або відсутність ознаки. Вихідним значенням тренувальної моделі DenseNet121 є вектор із 14 чисел, кожне з яких відповідає за наявність певної хвороби. Значення чисел вектора знаходяться в межах від нуля включно до одиниці включно, числа більше 0.7 будуть сприйматися як наявність хвороби, числа менше 0.3 будуть сприйматися як її відсутність. Усе, що між 0.3 та 0.7, буде відповідати значенням «не зазначена» або «не відомо». Для тренування, як в класифікаторі сторін, було використане передавальне навчання та модель DenseNet121, яка була попередньо натренована на наборі даних ImageNet. Як функція втрат використовувалася перехрестна втрата ентропії та Adam [17] як оптимізатор.

Тренування класифікатора хвороб відбувалося протягом 50 епох на відеокарті Nvidia GeForce RTX 2060 6 ГБ. Для тренування використовувалося 156553 випадково вибраних рентенівських знімків з набору даних CheXpert. Тестування моделі після кожної епохи відбувалося на 67095 рентгенівських знімках. Після останньої епохи точність визначення наявості та відсутності хвороб на тестових даних склала 86,71%, що майже відповідає результатам у досліджені CheXpert. Для донавчання використовувалися такі гіперпараметри: розмір партії — 32, темп навчання — 0.0001.

Для збереження індексів, звітів, метаданих та виконання пошуку за індексами, як векторна база даних була взята ChromaDb, котра повністю відповідає вимогам для створення системи. В ChromaDb дані зберігаються в колекціях аналогічно до таблиць в реляційних базах даних. Кожен запис в колекції повинен мати індекс, за яким буде відбуватися пошук, та якісь дані. Також запис може містити додаткові дані — метадані, за якими може відбуватися додатковий пошук. Індекси всіх записів в колекції повинні бути однієї розмірності.

У межах реалізації системи було створено три окремі колекції в ChromaDb, кожна з яких відповідала певному типу рентгенівських знімків і використовувала відповідну модель індексації. Для створення колекцій використовувся набір даних MIMIC-CXR. Перша колекція була створена за допомогою моделі індексації передніх знімків на 100000 випадково вибраних пар передніх ретгенівських знімків та їхніх звітів. Друга була створена за допомогою

моделі індексації бічних знімків на 50000 випадково вибраних пар бічних ретгенівських знімків та їхніх звітів. Третя була створена за допомогою загальної моделі індексації на 150000 випадково вибраних пар передніх і бічних ретгенівських знімків та їхніх звітів.

Генератором була обрана велика мовна модель LLaVA-Med 1.5 [18]. Ця модель є дотренованою версією LLaVA 1.5 на медичних даних, розробленою дослідниками з Microsoft [18]. У роботі використовувалася LLaVA-Med, яка містить 7 мільярдів параметрів. Оскільки для параметрів в нейронних мережах використовується 32-бітні числа з рухомою комою, то для запуску LLaVA-Med 1.5 на відеокарті треба понад 26 ГБ відеопам'яті. Під час виконання роботи не було доступу до відеокарт з такою кількістю відеопам'яті. Для того, щоб запускати нейронні мережі на відеокартах з меншою кількісю відеопам'яті, використовують техніку квантинізації [22]. Квантинізація дозволяє зменшити кількість пам'яті, яку займає модель, за допомогою представлення параметрів моделі 16-ти, 8-ми або 4-ма бітними числами з рухомою комою. Звісно, використання квантинізації може призвести до деградації моделі, але за дослідженням це або не відбувається взагалі, або зміни не суттєві. Для запуску LLaVA-Med 1.5 на Nvidia GeForce RTX 2060 6 ГБ була використана техніка квантинізації у режимі 4-біт. В результаті модель займає приблизно 5 ГБ відеопам'яті.

Для демонстрації роботи системи у процесі генерування звіту на вхідне зображення було використано випадковий рентгенівський знімок з набору даних MIMIC-CXR. В результаті мультимодальний RAG згенерував такий звіт: «The chest X-ray image shows a large right pleural effusion, which is an abnormal accumulation of fluid in the pleural space surrounding the lungs. This finding is concerning for pneumonia, which is an infection that causes inflammation in the air sacs of the lungs. It is important to consult a healthcare professional for a thorough evaluation and proper diagnosis of the underlying cause of these findings.». Еталонний звіт із набору даних для цього ж самого зображення: «impression: Increased right pleural loculated effusion with chest tube in place. Increasing consolidation in the right lung is concerning for pneumonia. Findings: PA and lateral views of the chest provided. Port-A-Cath is unchanged in position with its tip positioned in the expected location of the mid SVC. A right pleural drain is in place with increased opacity in the right lung and probable increase in size of the loculated right pleural effusion. Findings are concerning for a superimposed consolidation/pneumonia. The left lung remains essentially clear. The heart is difficult to assess given the effacement of the right heart border. The prominence of the mediastinum may reflect in part adjacent loculated pleural fluid. No pneumothorax is seen.».

Для демонстрації роботи системи як оцінювача конкретного питання за вхідне зображення було використано випадковий рентгенівський знімок з набору даних MIMIC-CXR Вхідним питанням до системи було: «Is the cardiomediastinal silhouette within normal limits?». В результаті система згенерувала таку відповідь: «Yes, the cardiomediastinal silhouette appears to be within normal limits in the image.». Еталонна відповідь на це питання до цього ж самого зображення з набору даних: «Yes».

Тестування системи

Тестування створеної системи проводилося на шістьох конфігураціях мультимодального RAG, кожна з яких була налаштована на знаходження 5-ти релевантних звітів до кожного запиту. Як зразок порівняння конфігурацій також була протестована LLaVA-Med 1.5 без використання ретриверу.

В рамках створення конфігурацій системи використовуються поняття «точне співпадіння хвороб» і «неточне співпадіння хвороб», позначаючи точне співпадіння хвороб у сховищі відносно вхідного зображення, чи щоб було хоча б одне співпадіння хвороб відповідно.

Ретривер першої конфігурації мультимодальної RAG-системи (рис.1) включає класифікатор сторін, модель індексації передніх знімків, модель індексації бічних знімків і класифікатор хвороб. Ретривер використовує неточне співпадіння хвороб.

Рис. 1. Конфігурація 1 (з класифікатором сторін, з класифікатором хвороб, неточне співпадіння хвороб)

Ретривер другої кофігурації мультимодальної RAG-системи (Рис.2) майже такий самий, як у конфігурації 1, але використовує точне співпадіння хвороб.

Рис. 2. Конфігурація 2 (з класифікатором сторін, з класифікатором хвороб, точне співпадіння хвороб)

Ретривер третьої кофігурації мультимодальної RAG-системи (рис.2) схожий на ретривер конфігурації 1 і конфігурації 2, але не використовує класифікатор хвороб.

Рис. 3. Конфігурація 3 (з класифікатором сторін, без класифікатора хвороб)

Ретривер четвертої конфігурації мультимодальної RAG-системи (рис.4) включає загальну модель індексації та класифікатор хвороб. Ретривер використовує неточне співпадіння хвороб.

Рис. 4. Конфігурація 4 (без класифікатора сторін, з класифікатором хвороб, неточне співпадіння хвороб)

Ретривер п’ятої кофігурації мультимодальної RAG-системи (рис.5) майже такий самий, як у конфігурації 4, але використовує точне співпадіння хвороб.

Рис. 5. Конфігурація 5 (без класифікатора сторін, з класифікатором хвороб, точне співпадіння хвороб)

Ретривер шостої кофігурації мультимодальної RAG-системи (рис.6) схожий на ретривер конфігурації 4 і конфігурації 5, але не використовує класифікатор хвороб.

Рис. 6. Конфігурація 6 (без класифікатора сторін, без класифікатора хвороб)

Для тестування мультимодальної RAG-системи для відповіді на вхідне запитання і зображення до нього були використані дві вибірки тестових даних, створені дослідниками мультимодального RAG [32]. Кожна вибірка даних містить приблизно 2500 запитань, відповідей і зображень. Одна вибірка містить зображення з набору даних MIMIC-CXR, а інша з набору даних IU-Xray. Дослідники, які створили вибірки даних, використовували велику мовну модель ChatGPT-4 для створення запитань і відповідей до рентгенівських знімків. Потім дослідники власноруч відфільтрували і перевірили кожне питання і відповідь. Усі запитання створені таким чином, щоб вони мали тільки два варіанта відповіді: «так» або «ні».

Для тестування системи відповідати на вхідне запитання і зображення до нього було використано дві метрики: метрика точності (англ. accuracy) [12] і метрика F1 (англ. F1-score) [30].

Метрика точності — одна з найпростіших і найпоширеніших оцінок якості

класифікаційних моделей. Вона показує, яка частка всіх передбачень моделі виявилася правильною. В нашому випадку вона покаже яка частка відповідей була правильно визначена. Точність набуває значень від нуля включно до одиниці включно, де нуль означає, що ніяка відповідь не була визначена правильно, а одиниця означає, що всі відповіді були визначені правильно.

Для обчислення метрики F1 треба поділити всі відповіді на позитивні і негативні. Позитивні відповіді — це відповіді, які ми оцінюємо, а негативні — всі інші. F1 показує, наскільки добре модель одночасно уникає хибних спрацьовувань (помилкових позитивних відповідей) та пропусків (хибних негативних відповідей). Вона обчислюється як гармонійне середнє [6] між влучністю [31] (англ. precision, доля правильних позитивних передбачень серед усіх позитивних передбачень) і повнотою [31] (англ. recall, доля правильних позитивних передбачень серед усіх позитивних відповідей), тому низьке значення однієї з цих метрик суттєво знижує F1. F1, як і точність, набуває значень від нуля включно до одиниці включно. Нуль означає, що модель не змогла правильно передбачити жодного позитивного випадку, а одиниця означає ідеальну відповідність. Для обчислення метрики F1 на вибірках даних буде використовуватися зважена F1 (англ. F1-weighted). Зважена F1 обчислюється окремо для кожного типу відповіді, а потім об'єднується, використовуючи кількісне значення кожного типу відповіді.

Тестування відбувалося на 1000 випадково вибраних запитань з кожної вибірки даних. Результати тестування наведені у таблиці 1.

Таблиця 1.

Результати тестування системи відповідати на питання за зображенням

Система	MIMIC-CXR		IU-Xray	
	Точність	Зважена F1	Точність	Зважена F1
LLaVA-Med 1.5	0.717	0.668	0.411	0.418
Конфігурація 1	**0.840**	**0.838**	0.921	0.924
Конфігурація 2	0.844	0.842	**0.909**	**0.913**
Конфігурація 3	0.831	0.830	0.929	0.931
Конфігурація 4	0.766	0.766	0.917	0.919
Конфігурація 5	0.786	0.784	0.918	0.921
Конфігурація 6	0.773	0.773	0.921	0.923

Для тестування можливості мультимодального RAG генерувати звіт до вхідного зображення були використані дві вибірки тестових даних, створені дослідниками мультимодального RAG [32]. Перша вибірка містить 700 зображень та звітів з набору даних MIMIC-CXR, а друга — 1180 зображень та звітів з набору даних IU-Xray. Кожна вибірка сформована так, аби вона містила якнайрізноманітніші звіти. Тестування системи за різними метриками проводилось на 700 зображень та звітів з набору даних MIMIC-CXR (з першої вибірки) та 1000 зображень та звітів з набору даних IU-Xray (з другої вибірки).

Для тестування системи генерувати звіт до вхідного зображення було використано всього чотири метрики: BLEU [23],

ROUGE [19], F1-CheXbert [28], F1-RadGraph [8].

BLEU (Bilingual Evaluation Understudy) — це метрика для порівняння згенерованого чи перекладеного тексту з одним або кількома еталонними варіантами. Для оцінки за метрикою BLEU підраховують кількість однакових послідовностей із *n* слів (*n*-грам), що зустрічаються як у згенерованому, так і в еталонному тексті. Потім обчислюється влучність як відношення числа збігів до загальної кількості n-грам у згенерованому тексті. Зазвичай влучність обраховують для *n*-грам від одного до чотирьох слів. Далі ці влучності об'єднують за допомогою геометричного середнього. Оскільки короткий (порівняно з еталонним) згенерований текст часто має вищу частку збігів, під час розрахунку BLEU застосовують штраф за короткість (англ. brevity penalty). BLEU набуває значень від нуля включно до одиниці включно. Чим більше значення BLEU, тим більше згенерований текст схожий на еталонний. Варто зазначити, що ця метрика оцінює лише збіги слів, а не зміст чи стиль тексту.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) — це набір метрик для порівняння згенерованого чи перекладеного тексту з одним або кількома еталонними варіантами. Набір ROUGE складається з трьох метрик: ROUGE-1, ROUGE-2 і ROUGE-L. Для оцінки за ROUGE-1 підраховують збіги окремих слів між згенерованим і еталонним текстом. Потім обчислюється значення метрики F1 на основі кількості таких збігів. ROUGE-2 обчислюється аналогічно до ROUGE-1, але замість окремих слів рахують двослівні послідовності. Ця метрика дозволяє оцінити, наскільки модель правильно відтворює не лише окремі слова, а й стійкі словосполучення, порядок слів і короткі фрази. Для оцінки за ROUGE-L обраховують довжину найдовшої спільної підпослідовності між згенерованим і еталонним текстом. Ця метрика відображає не лише точні збіги слів, а й загальну послідовність, у якій вони з'являються. Усі метрики ROUGE набувають значень від нуля включно до одиниці включно. Чим більше значення метрики, тим більше згенерований текст схожий на еталонний. Як і BLEU, набір ROUGE не оцінює зміст чи стиль тексту, а лише лексичну подібність. Результати наведені у таблиці 2 і таблиці 3.

Таблиця 2.

Результати тестування системи генерувати звіт на вхідне зображення за допомогою метрики BLEU

Система	MIMIC-CXR	IU-Xray
LLaVA-Med 1.5	0.006204	0.014087
Конфігурація 1	0.025932	0.035426
Конфігурація 2	**0.042499**	0.036582
Конфігурація 3	0.027740	0.034322
Конфігурація 4	0.018719	0.035311
Конфігурація 5	0.022738	**0.037315**
Конфігурація 6	0.018848	0.035051

Таблиця 3.

Результати тестування системи генерувати звіт на вхідне зображення за допомогою метрик ROUGE-1, ROUGE-2, ROUGE-L

Система	MIMIC-CXR			IU-Xray		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
LLaVA-Med 1.5	0.177572	0.017352	0.111342	0.178954	0.024311	0.122759
Конфігурація 1	0.255128	0.063660	0.164584	0.258481	0.054633	0.175839
Конфігурація 2	**0.269676**	**0.079149**	**0.178946**	0.257788	0.054402	0.176659
Конфігурація 3	0.256807	0.067752	0.165909	0.258677	0.053155	0.176821
Конфігурація 4	0.233677	0.052150	0.147039	0.267725	**0.059491**	0.174326
Конфігурація 5	0.244741	0.055802	0.157708	**0.275352**	0.058866	**0.182624**
Конфігурація 6	0.234838	0.053292	0.148318	0.262273	0.055589	0.171794

F1-CheXbert — це метрика, яка оцінює наскільки точно модель відтворює набір клінічних ознак з набору даних CheXpert [14] у радіологічних звітах порівняно з еталонними звітами. F1-CheXbert використовує спеціальну модель CheXbert, яка визначає наявність кожної клінічної ознаки в конкретному звіті. Ця модель натренована на наборі даних CheXpert. Після цього CheXbert застосовують для виявлення клінічних ознак як у згенерованих, так і в еталонних звітах. Для кожної ознаки обчислюється значення метрики F1. Значення F1-CheXbert — це середнє арифметичне значення F1 для всіх ознак, щоб рідкісні, але важливі ознаки мали такий самий вплив на фінальний результат, як і найпоширеніші. F1-CheXbert набуває значень від нуля включно до одиниці включно. Чим більше значення F1-CheXbert, тим більше клінічних ознак із еталонного звіту модель правильно відтворила в згенерованому тексті.

F1-RadGraph — це метрика, яка оцінює, наскільки вдало згенерований радіологічний звіт відтворює структуровану інформацію про клінічні ознаки порівняно з еталонним. F1-RadGraph використовує спеціальну модель RadGraph [15], яка виявляє у звіті клінічні ознаки та зв'язки між ними, формуючи граф. RadGraph застосовують для виявлення клінічних ознак та зв'язків між ними, як у згенерованих, так і в еталонних звітах. Значення F1-RadGraph — це значення F1, а саме F1-мікро (англ. F1-micro), що обчислюється для всіх сутностей і зв'язків разом. F1-RadGraph набуває значень від нуля включно до одиниці включно. Чим більше значення F1-RadGraph, тим більше клінічних ознак і зв'язків між ними з еталонного звіту модель правильно відтворила в згенерованому тексті. Результати наведені у таблиці 4.

Таблиця 4.

Результати тестування системи генерувати звіт на вхідне зображення за допомогою метрики F1-RadGraph і F1-CheXpert

Система	MIMIC-CXR		IU-Xray	
	F1-RadGraph	F1-CheXpert	F1-RadGraph	F1-CheXpert
LLaVA-Med 1.5	0.020411	0.011978	0.059309	0.120694
Конфігурація 1	0.095536	0.266570	0.120658	0.217418
Конфігурація 2	**0.101452**	**0.289483**	0.119638	0.219341
Конфігурація 3	0.097419	0.263669	0.122248	**0.226831**
Конфігурація 4	0.076431	0.221617	0.118745	0.182259
Конфігурація 5	0.079204	0.235491	**0.122624**	0.180509
Конфігурація 6	0.078109	0.214062	0.112216	0.179557

За результатами тестування відповідати на запитання до вхідного зображення конфігурація 2 (з класифікатором сторін, з класифікатором хвороб, точне співпадіння хвороб) найкраще впоралася на наборі даних MIMIC-CXR, а конфігурація 3 (з класифікатором сторін, без класифікатора хвороб) найкраще впоралася на наборі даних IU-Xray. На наборі даних MIMIC-CXR система показала приблизно на 26% кращий результат, ніж LLaVA-Med 1.5 без ретриверу відповідно за зваженою F1 метрикою, а на наборі даних IU-Xray приблизно на 123% краще. Такий великий розрив у результатах можна пояснити тим, що розробники LLaVA-Med 1.5 використовували набір даних MIMIC-CXR для тренування, тому результати LLaVA-Med 1.5 на MIMIC-CXR вищі, ніж на IU-Xray. Низькі результати LLaVA-Med на IU-Xray можна пояснити її використанням у режимі 4-біт. Водночас розроблена система показала найкращі результати саме на IU-Xray.

Тестування розробленої системи генерувати звіт за допомогою метрик, які оцінюють збіги слів, а саме BLEU, ROUGE-1, ROUGE-L показало, що конфігурація 2 (з класифікатором сторін, з класифікатором хвороб, точне співпадіння хвороб) найкраще впоралася на наборі даних MIMIC-CXR, а конфігурація 5 (без класифікатора сторін, з класифікатором хвороб, точне співпадіння хвороб) найкраще впоралася на наборі даних IU-Xray. Результати метрики ROUGE-2 відрізняються тим, що конфігурація 4 (без класифікатора сторін, з класифікатором хвороб, неточне співпадіння хвороб) найкраще управилася на наборі даних IU-Xray. З результатів цих метрик зрозуміло, що система генерує більш схожі звіти на еталонні, ніж просто LLaVA-Med 1.5. Ймовірно, це зумовлено тим, що система надає релевантні звіти генератору, і він намагається створити звіти схожої структури, тоді як використання LLaVA-Med 1.5 без генератора генерує доволі короткі і недеталізовані звіти. Конфігурації без використання класифікатора сторін, а з використанням загальної моделі індексації, показали кращі результати на наборі даних IU-Xray ймовірніше за все через те, що в

цьому наборі даних використовується однаковий звіт до передніх і бічних знімків на відміну від MIMIC-CXR.

Тестування розробленої системи генерувати звіт за допомогою метрики F1-CheXbert, яка оцінює зміст тексту, показало, що конфігурація 2 (з класифікатором сторін, з класифікатором хвороб, точне співпадіння хвороб) найкраще впоралася на наборі даних MIMIC-CXR, а конфігурація 3 (з класифікатором сторін, без класифікатора хвороб) найкраще впоралася на наборі даних IU-Xray. Конфігурація 2 показала більше, ніж вдвічі кращий результат, аніж LLaVA-Med 1.5 без ретриверу на наборі даних MIMIC-CXR, а конфігурація 3 більше, ніж в 1.8 рази кращий результат на наборі даних IU-Xray.

Тестування розробленої системи щодо генерування звіту за допомогою метрики F1-RadGraph, яка також оцінює зміст тексту, показало, що конфігурація 2 (з класифікатором сторін, з класифікатором хвороб, точне співпадіння хвороб) найкраще впоралася на наборі даних MIMIC-CXR, а конфігурація 5 (без класифікатора сторін, з класифікатором хвороб, точне співпадіння хвороб) найкраще управилася на наборі даних IU-Xray. Конфігурація 2 показала приблизно в 5 разів кращий результат, аніж LlaVA-Med без ретриверу на наборі даних MIMIC-CXR, а конфігурація 5 більше ніж вдвічі кращий результат на наборі даних IU-Xray.

Висновки

Загалом результати тестування демонструють, що використання техніки мультимодального RAG суттєво покращує генерацію відповідей і звітів LLaVA-Med 1.5, навіть попри обмеження 4-бітного режиму, за якого без ретривера генерує короткі і недеталізовані відповіді. Релевантні звіти, отримані через ретривер, дозволяють генератору створювати кращі за структурою й змістом відповіді. В більшості випадків кофігурації з використанням класифікатора хвороб, а саме точним співпадінням хвороб, показали найкращі результати. Випадки, коли такі конфігурації давали гірші результати, ймовірно пов'язані з неідеальною точністю класифікатора хвороб. Класифікатор сторін також у більшості сценаріїв покращував результати системи порівняно з конфігураціями без нього. Враховуючи вище сказане, можна зробити висновок, що конфігурація 2 (з класифікатором сторін, з класифікатором хвороб, точне співпадіння хвороб) є найкращою серед інших протестованих.

Під час роботи було створено мультимодальну RAG-систему для аналізу та інтерпретації рентгенівських знімків грудної клітки та їхніх звітів. Система використовує новітні технології та програмні засоби, а саме: PyTorch, LLaVA-Med 1.5, BioMedCLIP, DenseNet121, ChromaDB. Розроблена система здатна відповідати на запитання до рентгенівського знімку та генерувати радіологічний звіт до знімку. Архітектура системи складається з декількох підсистем: різні моделі індексації, класифікатор сторін, класифікатор хвороб, генератор, сховище даних. Було запропоновано декілька конфігурацій системи для їхнього тестування.

Також проведене тестування шести конфігурацій створеної мультимодальної RAG-системи. Для порівняння була протестована LLaVA-Med 1.5 без ретриверу. Генерацію відповідей на запитання до зображення оцінювали за метриками точності та F1, а генерацію звітів — за метриками BLEU, ROUGE, F1-CheXbert, F1-RadGraph. Результати тестування були детально проаналізовані та визначено найкращу конфігурацію системи. За результатами встановлено, що використання техніки мультимодального RAG значно покращує можливості LLaVA-Med 1.5 у роботі з рентгенівськими знімками і їхніми звітами, навіть за умов обмежених ресурсів, зокрема, у використанні моделі в 4-бітному режимі.

Майбутні перспективи розвитку можливі в напрямку використання не тільки тексту та зображень як даних, а й інших типів даних, наприклад: аудіо, відео тощо. Перспективним є також створення мультимодальних RAG-систем для інших предметних сфер та автоматизація процесу їхньої розробки.

References

1. Belcic, I. and Stryker, C. (n.d.) *What is Learning Rate in Machine Learning?*. IBM. [online] Available at: https://www.ibm.com/think/topics/learning-rate [Accessed 25 May 2025].
2. Bergmann, D. and Stryker, C. (n.d.) *What is Loss Function?*. IBM. [online] Available at: https://www.ibm.com/think/topics/loss-function [Accessed 25 May 2025].
3. Chen, W. et al. (2022) *MuRAG: Multimodal Retrieval-Augmented Generator for Open Question Answering over Images and Text*. [online] Available at: https://arxiv.org/abs/2210.02928 [Accessed 10 October 2025].
4. Coursera (n.d.) *What Does Batch Size Mean in Deep Learning? An In-Depth Guide*. [online] Available at: https://www.coursera.org/articles/what-does-batch-size-mean-in-deep-learning [Accessed 25 May 2025].
5. Coursera (n.d.) *What Is an Epoch in Machine Learning?*. [online] Available at: https://www.coursera.org/articles/epoch-in-machine-learning [Accessed 25 May 2025].
6. DeepAI (n.d.) *Harmonic Mean*. [online] Available at: https://deepai.org/machine-learning-glossary-and-terms/harmonic-mean [Accessed 25 May 2025].
7. DeepAI (n.d.) *Hyperparameter*. [online] Available at: https://deepai.org/machine-learning-glossary-and-terms/hyperparameter [Accessed 25 May 2025].
8. Delbrouck, J.-B. et al. (2022) *Improving the Factual Correctness of Radiology Report Generation with Semantic Rewards*. [online] Available at: https://arxiv.org/abs/2210.12186 [Accessed 10 October 2025].
9. Demner-Fushman, D. et al. (2015) 'Preparing a collection of radiology examinations for distribution and retrieval', *Journal of the American Medical Informatics Association*, 23(2), pp. 304–310. doi: 10.1093/jamia/ocv080.
10. Deng, J. et al. (2009) 'ImageNet: A large-scale hierarchical image database', in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Miami, Florida, 20-25 June. IEEE, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.
11. Hugging Face (n.d.) *What is Image Classification?*. [online] Available at: https://huggingface.co/tasks/image-classification [Accessed 25 May 2025].
12. IBM (n.d.) *IBM Watson Studio and Knowledge Catalog*. [online] Available at: https://www.ibm.com/docs/en/ws-and-kc?topic=metrics-accuracy [Accessed 25 May 2025].
13. IBM (n.d.) *What are Convolutional Neural Networks?*. [online] Available at: https://www.ibm.com/think/topics/convolutional-neural-networks [Accessed 25 May 2025].
14. Irvin, J. et al. (2019) *CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison*. [online] Available at: https://arxiv.org/abs/1901.07031 [Accessed 10 October 2025].
15. Jain, S. et al. (2021) *RadGraph: Extracting Clinical Entities and Relations from Radiology Reports*. [online] Available at: https://arxiv.org/abs/2106.14463 [Accessed 10 October 2025].
16. Johnson, A. E. W. et al. (2019) *MIMIC-CXR-JPG, a large publicly available database of labeled chest radiographs*. [online] Available at: https://arxiv.org/abs/1901.07042 [Accessed 10 October 2025].
17. Kingma, D. P. and Ba, J. (2014) *Adam: A Method for Stochastic Optimization*. [online] Available at: https://arxiv.org/abs/1412.6980 [Accessed 10 October 2025].
18. Li, C. et al. (2023) *LLaVA-Med: Training a Large Language-and-Vision Assistant for Biomedicine in One Day*. [online] Available at: https://arxiv.org/abs/2306.00890 [Accessed 10 October 2025].
19. Lin, C.-Y. (2004) 'ROUGE: A Package for Automatic Evaluation of Summaries', in *Text Summarization Branches Out*. Barcelona, Spain, 25-26 July. ACL, pp. 74–81. Available at: https://aclanthology.org/W04-1013/ [Accessed 10 October 2025].
20. Loshchilov, I. and Hutter, F. (2017) *Decoupled Weight Decay Regularization*. [online] Available at: https://arxiv.org/abs/1711.05101 [Accessed 10 October 2025].
21. Mao, A., Mohri, M. and Zhong, Y. (2023) *Cross-Entropy Loss Functions: Theoretical Analysis and Applications*. [online] Available at: https://arxiv.org/abs/2304.07288 [Accessed 10 October 2025].
22. Nagel, M. et al. (2021) *A White Paper on Neural Network Quantization*. [online] Available

at: https://arxiv.org/abs/2106.08295 [Accessed 10 October 2025].
23. Papineni, K. et al. (2002) 'Bleu: a Method for Automatic Evaluation of Machine Translation', in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, 7-12 July. ACL, pp. 311–318. doi: 10.3115/1073083.1073135.
24. PhysioNet (2023) *MIMIC-CXR Database*. [online] Available at: https://physionet.org/content/mimic-cxr/2.1.0/ [Accessed 25 May 2025].
25. Radford, A. et al. (2021) *Learning Transferable Visual Models From Natural Language Supervision*. [online] Available at: https://arxiv.org/abs/2103.00020 [Accessed 10 October 2025].
26. Read, J. and Perez-Cruz, F. (2015) *Deep Learning for Multi-label Classification*. [online] Available at: https://arxiv.org/abs/1502.05988 [Accessed 10 October 2025].
27. Ruder, S. (2016) *An overview of gradient descent optimization algorithms*. [online] Available at: https://arxiv.org/abs/1609.04747 [Accessed 10 October 2025].
28. Smit, A. et al. (2020) *CheXbert: Combining Automatic Labelers and Expert Annotations for Accurate Radiology Report Labeling Using BERT*. [online] Available at: https://arxiv.org/abs/2004.09167 [Accessed 10 October 2025].
29. Sun, S. et al. (2019) *A Survey of Optimization Methods from a Machine Learning Perspective*. [online] Available at: https://arxiv.org/abs/1906.06821 [Accessed 10 October 2025].
30. Wood, T. (n.d.) *F-Score*. DeepAI. [online] Available at: https://deepai.org/machine-learning-glossary-and-terms/f-score [Accessed 25 May 2025].
31. Wood, T. (n.d.) *Precision and Recall*. DeepAI. [online] Available at: https://deepai.org/machine-learning-glossary-and-terms/precision-and-recall [Accessed 25 May 2025].
32. Xia, P. et al. (2024) *MMed-RAG: Versatile Multimodal RAG System for Medical Vision Language Models*. [online] Available at: https://arxiv.org/abs/2410.13085 [Accessed 10 October 2025].
33. Zhang, S. et al. (2023) *BiomedCLIP: a multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs*. [online] Available at: https://arxiv.org/abs/2303.00915 [Accessed 10 October 2025].
34. Zhao, R. et al. (2023) *Retrieving Multimodal Information for Augmented Generation: A Survey*. [online] Available at: https://arxiv.org/abs/2303.10868 [Accessed 10 October 2025].

Одержано: 11.10.2025
Внутрішня рецензія отримана: 20.10.2025
Зовнішня рецензія отримана: 22.10.2025

Про авторів

[1]*Шевченко Михайло Григорович*
Бакалавр
https://orcid.org/0009-0004-5933-2349
[1]*Андрощук Максим Віталійович*
Здобувач ступеня доктора філософії
https://orcid.org/0000-0001-6183-6950

Місце роботи авторів:

[1]Національний університет
«Києво-Могилянська академія»
тел. +38-044-425-60-59
E-mail: vkd@ukma.edu.ua
https://www.ukma.edu.ua/

УДК 004.94 https://doi.org/10.15407/pp2025.03.079

В.В. Вдовиченко, В.Л. Плескач

ГЕОІНФОРМАЦІЙНІ ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ НА БАЗІ СУЧАСНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ЦИФРОВОГО МОДЕЛЮВАННЯ МІСЦЕВОСТІ

Статтю присвячено аналізу сучасних підходів до цифрового моделювання рельєфу на основі геоінформаційних інтелектуальних систем і новітніх інформаційних технологій. Висвітлено еволюцію методів моделювання місцевості – від традиційних ГІС-технологій до інтеграції методів штучного інтелекту (GeoAI). Метою дослідження є узагальнення можливостей і переваг застосування штучного інтелекту (машинного та глибинного навчання, комп'ютерного зору) у задачах цифрового моделювання місцевості та визначення перспектив розвитку таких інтелектуальних систем.

Розглянуто можливості застосування керованих алгоритмів машинного навчання (дерева рішень, Random Forest, SVM) для класифікації форм рельєфу та обробки даних дистанційного зондування, а також методів глибинного навчання для автоматичного розпізнавання шаблонів і семантичної сегментації рельєфу. Наведено приклади інтелектуальних ГІС-рішень у різних прикладних сферах: систем моніторингу стійкості схилів із прогнозуванням зсувів, моделей оцінки ризику повеней, військових систем планування місцевості з оптимізацією маршрутів дронів тощо. Основні результати демонструють, що впровадження AI-технологій забезпечує відповідну деталізацію та автоматизацію аналізу рельєфу. Сучасні геоінформаційні інтелектуальні системи здатні поєднувати різнорідні дані (LiDAR-сканування, супутникові знімки, дані дронів, наземні сенсори) й оперативно оновлювати цифрові моделі місцевості в режимі реального часу, підтримуючи ухвалення рішень у сфері управління територіями та надзвичайними ситуаціями.

Перспективи розвитку GeoAI вбачаються у подальшому поєднанні різнорідних джерел даних, переході від статичних 3D-моделей до динамічних 4D-моделей рельєфу, семантичному збагаченні цифрових моделей (онтології ландшафту, виділення об'єктів) та інтеграції з концепцією цифрових двійників територій. Зроблено висновок, що поєднання традиційних ГІС-засобів із технологіями штучного інтелекту трансформує галузь цифрового моделювання рельєфу, відкриваючи нові горизонти для наукових досліджень і практичного застосування.

Ключові слова: штучний інтелект, комп'ютерний зір, цифрове моделювання місцевості, глибинне навчання, геоінформаційна інтелектуальна система, цифрова модель рельєфу, цифрова модель висот, GeoAI.

Vdovychenko V., Pleskach V.

GEOGRAPHIC INFORMATION INTELLIGENT SYSTEMS BASED ON MODERN INFORMATION TECHNOLOGIES FOR DIGITAL TERRAIN MODELING

The article presents a comprehensive analysis of modern approaches to digital terrain modeling based on intelligent geographic information systems and advanced information technologies. It highlights the evolution of terrain modeling methods — from traditional GIS technologies to the integration of artificial intelligence approaches (the GeoAI concept). The purpose of the study is to generalize the capabilities and advantages of applying artificial intelligence (machine learning, deep learning, and computer vision) to digital terrain modeling tasks and to identify prospects for the development of such intelligent systems.

The study explores the application potential of supervised machine learning algorithms (decision trees, Random Forest, SVM) for landform classification and remote sensing data processing, as well as deep learning methods (CNN) for automatic pattern recognition and semantic terrain segmentation. Examples of intelligent GIS solutions are provided in various applied fields: slope stability monitoring systems with landslide forecasting, flood risk assessment models, and military terrain planning systems with drone route optimization. The main results demonstrate that the implementation of AI technologies enables unprecedented detail and automation in terrain analysis. Modern geospatial intelligent systems are capable of integrating heterogeneous data sources (LiDAR scans, satellite imagery, drone data, ground sensors) and updating digital terrain models in real time, thereby supporting decision-making in land management and emergency response.

The future development of GeoAI is defined by the continued integration of diverse data sources, the transition from static 3D models to dynamic 4D terrain representations, semantic enrichment of digital models (landscape ontologies, object detection), and their integration into the concept of territorial digital twins. It is concluded that the synergy of traditional GIS tools with state-of-the-art artificial intelligence technologies is transforming the field of digital terrain modeling, opening new horizons for scientific research and practical applications.
Keywords: artificial intelligence, computer vision, digital terrain modelling, deep learning, geographic information system, digital elevation model, digital elevation model, GeoAI.

Вступ

Цифрове моделювання місцевості передбачає створення та аналіз цифрових представлень топографії Землі, себто, поверхні, очищеної від об'єктів, для географічного аналізу та візуалізації. Цифрова модель рельєфу (ЦМР, Digital Elevation Model, DEM) зазвичай є тривимірним зображенням земної поверхні без рослинності, будівель та інших об'єктів [1]. Вона відрізняється від цифрової моделі поверхні (ЦМП), яка охоплює будівлі та рослинність, а також від загального терміну «цифрова модель висот» (ЦМВ), що може стосуватись обох типів. Усі ці моделі дозволяють з високою точністю відображати елементи рельєфу, такі як схили, контури та долини [2]. Вони є основою для таких застосувань, як картографування ризиків повеней, проєктування інфраструктури, ландшафтна екологія та військове планування.

Водночас широкого поширення набув підхід GeoAI. Геоінформаційні інтелектуальні системи (ГІС) дедалі частіше інтегрують машинне навчання (МН, Machine Learning, ML) і штучний інтелект (ШІ) для автоматизації складних завдань, які раніше потребували участі експертів. Наприклад, алгоритми розпізнавання шаблонів можуть ідентифікувати геоморфологічні структури (хребти, долини, карстові форми) на основі ЦМВ, а прогностичні моделі — встановлювати зв'язок між хара-ктеристиками рельєфу та явищами на кшталт зсувів або властивостей ґрунтів.

Інтеграція ШІ є визначальною для осмислення великого обсягу даних рельєфу. GeoAI «підсилює традиційний геоаналіз та картографування, змінюючи способи розуміння та управління складними системами людина-природа» [3]. У контексті моделювання рельєфу ГІС базуються на здобутках ГІС-науки, поєднуючи їх із сучасними методами здобуття знань, системами управління знаннями і МН. Вивчення ГІС на базі сучасних ІТ для цифрового моделювання місцевості є важливим кроком до створення нової парадигми просторового аналізу — динамічного, автоматизованого та здатного до прогнозування.

Фундаментальні дослідження

Цифровий аналіз і моделювання рельєфу вже десятиліттями є активною сферою досліджень, що базується на дисциплінах геоморфології, геодезії та комп'ютерних науках [4]. Класичні праці заклали основу представлення й аналізу рельєфу в геоінформаційних системах (ГІС). Зокрема, у праці *Digital Terrain Modeling: Principles and Methodology* опубліковано вагомі висновки, що систематизували основні принципи побудови та аналізу ЦМР [4], [5]. Вагомим внеском у розуміння поверхневих процесів стало засто-сування ЦМР в гідрології, геоморфології та біології [6].

Дослідження з геоморфометрії (кількісного аналізу рельєфу) [7], зокрема, огляд методів поверхневого аналізу, зробили важливий внесок у розвиток галузі. Закладено основи обчислення параметрів рельєфу — таких як схил, кривизна, водозбірні площі — на основі ЦМВ, які досі є ядром цифрового аналізу місцевості.

Із часом розвиток високоточних просторових даних та обчислювальних технологій суттєво змінив підходи до моделювання рельєфу. Досягнення в дистанційному зондуванні (LiDAR, радар, супутникові знімки високої роздільності) та національні програми збору висотних даних привели до появи великих обсягів багатоджерельної інформації про рельєф.

Відкриті ініціативи урядів і наукових установ забезпечили широкодоступність глобальних ЦМВ високої роздільності. Зараз дослідники мають безпрецедентний доступ до детальних топографічних даних на локальному та глобальному рівнях. Проте цей обсяг інформації створює не лише нові можливості, а й виклики щодо отримання осмислених результатів [4]. Традиційні ГІС (наприклад, ArcGIS від Esri, або open-source GRASS GIS і QGIS) охоплюють потужні інструменти для моделювання рельєфу, однак масштаб і різнорідність сучасних даних часто вимагають інтелектуальніших підходів.

Штучний інтелект і машинне навчання в цифровому моделюванні рельєфу

Сучасні методи ШІ значно розширили інструментарій аналізу в цифровому моделюванні рельєфу. ML та глибинне навчання (ГН, Deep Learning, DL) дозволяють автоматизувати виділення ознак і здійснювати прогнозне моделювання, доповнюючи класичні фізико-математичні ГІС-підходи.

Методи машинного навчання

Керовані алгоритми МН, такі як дерева рішень, Random Forest, метод опорних векторів (SVM) та ансамблеві методи, успішно застосовують до геопросторових даних рельєфу. Ці підходи дозволяють виявляти взаємозв'язки між рельєфними характеристиками (нахил, кривизна, тип ґрунту тощо) та цільовими явищами, часто природного чи геоморфологічного характеру.

Одним з прикладів корисного виявлення таких взаємозв'язків між рельєфними характеристиками є система моніторингу стійкості схилів. Вона використовує Python-платформу Pastas для прогнозу водовмісту та тиску порогу, а також моделі Random Forest, що була навчена на щоденних гідрометеорологічних даних, і поліноміальну регресію для оцінки коефіцієнта безпеки (FoS). Результати щодня автоматично обробляють, візуалізують в реальному часі та надсилаються на платформу NGI Live [8]. Результати моделі оцінено шляхом порівняння прогнозованих значень коефіцієнта стійкості схилу (FoS), отриманих за допомогою підходу на основі даних, із розрахованими значеннями FoS у GeoStudio для тестової та валідаційної вибірок з використанням коефіцієнта детермінації (R^2) та середньоквадратичної помилки (RMSE).

Нижче наведена формула для розрахунку коефіцієнта детермінації та середньоквадратичної помилки.

$$R^2 = 1 - \frac{\sum(y_i - v_i)^2}{\sum(y_i - \varphi)^2} \quad RMSE = \sqrt{\frac{\sum(y_i - v_i)^2}{n}},$$

де y_i — спостережуване значення, v_i — прогнозоване значення, а φ — середнє значення серед n спостережуваних значень.

У табл. 1 подано оцінювання ефективності прогнозів FoS на основі моделей МН. Жирним виділено найкращий результат для кожної моделі.

Таблиця 1.

Оцінювання ефективності прогнозів FoS на основі моделей МН.

Data-driven model	Dataset	Test		Validation	
		R^2	$RMSE$	R^2	$RMSE$
PR	Group 1	0.973	0.018	0.770	0.018
	Group 2	0.925	0.031	-3.280	0.116
	Group 3	0.966	0.018	**0.910**	**0.020**
RF	Group 1	0.975	0.016	**0.828**	**0.015**
	Group 2	0.999	0.004	0.135	0.052
	Group 3	0.978	0.014	0.782	0.032

У разі перевищення критичних порогів (рис. 1) система інформує адміністраторів. Рішення стало результатом успішної інтеграції інженерних, природничих і комп'ютерних наук.

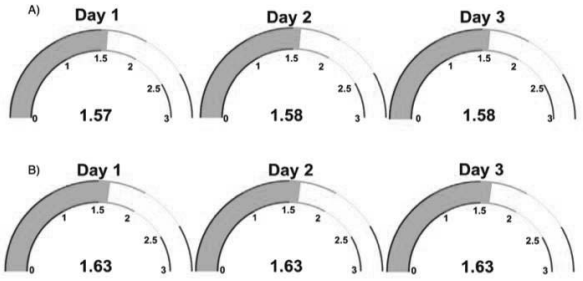


Рис. 1. Приклади вихідних прогнозованих значень: A) FoS, прогнозованого за допомогою поліноміальної регресії (PR), B) FoS, прогнозованого за допомогою Random Forest (RF). Джерело: [8].

ML також широко застосовується для класифікації рельєфу або типів земної поверхні на основі даних дистанційного зондування. Об'єктно-орієнтований аналіз зображень у ГІС часто використовує ML для розпізнавання особливостей місцевості. Метод опорних векторів і випадкові ліси застосовують для класифікації хмар точок LiDAR на точки ґрунту й неґрунту або для поєднання LiDAR з мультиспектральними знімками з метою створення карт земної поверхні [9]. Такі класифікації є критичними для створення ЦМР (вилучення «оголеної» поверхні шляхом видалення дерев і будівель) та для генерації тематичних карт — геоморфологічних, ґрунтових тощо.

Ще одним напрямом є регресійні й прогностичні задачі на основі рельєфних даних. Наприклад, моделі можуть оцінювати безперервні величини, як-от вологість ґрунту, біомасу чи мікроклімат, з огляду на рельєфні характеристики (висота, нахил, експозиція). У точному землеробстві поєднання рельєфних даних із сенсорними вимірами дозволяє моделювати врожайність або оптимізувати зрошення. Здатність ML моделювати нелінійні залежності робить його надзвичайно ефективним для подібних задач, пов'язаних із довкіллям.

Інші підходи, засновані на знаннях, зокрема, нечітка логіка чи експертні системи, також використовувались у минулому для аналізу рельєфу. Наприклад, нечіткі логічні системи дозволяли комбінувати кілька тематичних шарів (нахил, геологія, земне покриття) у єдиний індекс зсувонебезпеки, відображаючи експертні уявлення про вплив кожного чинника. Попри те, що нині ці методи менш поширені порівняно зі статистичним ML чи DL, саме вони стали одними з перших прикладів застосування ШІ в ГІС і проклали шлях до сучасних підходів, заснованих на даних.

Глибинне навчання та комп'ютерний зір

Поява DL, особливо згорткових нейронних мереж (convolutional neural network, CNN), відкрила нові горизонти в цифровому моделюванні рельєфу. CNN справляються з розпізнаванням шаблонів у ґрідованих (растрових) даних, що робить їх особливо ефективними для аналізу ЦМВ або зображень. Одним із ключових застосувань є семантична сегментація рельєфу, коли кожен піксель або клітинка матриці класифікується за типом форми рельєфу або земної поверхні.

У 2023 році було продемонстровано ефективність підходу, застосувавши глибоку згорткову мережу (Fully Convolutional Network) з архітектурою ResNet для класифікації форм рельєфу на основі ЦМВ з роздільністю 30 м [10]. Модель виконувала піксельну сегментацію висотних даних на геоморфологічні класи (наприклад, гори, пагорби, рівнини).

Модель навчалась безпосередньо на «сирих» ЦМВ — додавання похідних шарів, як-от нахил чи кривизна, суттєво не покращувало точність. Метриками оцінки, використаними в цьому експерименті, були Pixel Accuracy (PA) та Mean Intersection over Union (MIoU). Для класифікації форм рельєфу за допомогою моделі FCN-ResNet було обрано п'ять факторів рельєфу, а саме: DEM, нахил (slope), RDLS, hill-shade (тіньовий рельєф) та кривизна поверхні (surface curvature) [10]. Нижче наведено формулу для розрахунку згаданих метрик.

$$PA = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} p_{ij}}$$

$$MIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}},$$

де припускаючи наявність k класів, визначаємо:

- p_{ii} — істинно позитивні значення (True Positives, TP),
- p_{ij}— хибно позитивні значення (False Positives, FP),
- p_{ji}— хибно негативні значення (False Negatives, FN).

Порівняльну характеристику метрик різних факторів рельєфу подано у табл.2.

Таблиця 2.

Порівняльна характеристика метрик різних факторів рельєфу

Фактор рельєфу	Похибка	PA	MIoU
DEM / ЦМР	0.0482	79.80	68.49
Slope / нахил	0.0374	79.23	67.53
Hillshade / тіньовий рельєф	0.0337	79.52	68.00
RDLS	0.0388	79.96	68.64
Curvature	0.0756	75.83	63.69
DEM + Slope + Hillshade	0.0244	80.91	69.76
DEM + Hillshade + Curvature	0.0281	80.52	69.24
DEM + RDLS + Hillshade	0.0257	81.18	70.14

PA та MIoU є поширеними метриками для семантичної сегментації, і їхнє обчислення базовано на матрицях плутанини (confusion matrices), що свідчить про здатність CNN автоматично виявляти багатомасштабні рельєфні особливості, які раніше здобувалися через спеціально розроблені індекси. Таким чином, DL демонструє великий потенціал для автоматизованої класифікації форм рельєфу та є потужним інструментом геоморфології.

DL також сприяє розвитку виявлення та виділення об'єктів рельєфу. CNN-моделі для розпізнавання об'єктів можна навчити ідентифікувати конкретні геоморфологічні структури на ЦМВ або супутникових зображеннях. Наприклад, є моделі, які виявляють зсуви на картах схилів або ідентифікують руслові мережі на ЦМР за допомогою глибоких нейронних мереж. Особливо показовим є підхід об'єднання різних джерел даних у ГН.

У 2021 році було розроблено GeoAI-конвеєр, який поєднував супутникові знімки з ЦМР у єдиній моделі для розпізнавання природних об'єктів [11]. Було застосовано, як злиття на рівні даних (стекування каналів зображення та ЦМР), так і злиття ознак всередині CNN, щоб навчити модель одночасно зображувальній і висотній інформації, що покращило якість виявлення об'єктів (наприклад, річок або хребтів), на відміну від використання лише одного джерела.

Ще один напрям — підвищення якості ЦМР за допомогою DL. Для побудови точних моделей рельєфу необхідно фільтрувати неґрунтові об'єкти (дерева, будівлі) з хмар точок, зокрема, LiDAR. Традиційні фільтри (морфологічні операції, порогові значення схилу) часто не справляються у складних умовах. DL-мережі можуть навчатися на прикладах розрізняти ґрунтові та неґрунтові точки. У 2024 році було запропоновано глибоку енкодер-декодерну мережу для виявлення та усунення об'єктів, розта-шованих над землею, у хмарах точок LiDAR [12]. Модель захоплює багатомасштабні ознаки та поєднує локальний і глобальний контекст для кращої ідентифікації. Результати показали, що підхід перевершує класичні методи фільтрації за точністю та надійністю.

Аналогічно моделі суперрезолюції на основі DL використовуються для підвищення просторової роздільності ЦМР — наприклад, перетворення даних з роздільністю 30м на 5м, навчаючись від детальніших зразків [13], [14]. Результати показали, що ця трансформерна мережа забезпечує найкращі показники за всіма метриками, зокрема середньоквадратичною похибкою (RMSE) для висоти, нахилу, експозиції та кривизни, доводячи, що мережі на базі Transformer мають перевагу над CNN та GAN у навчанні рельєфних ознак цифрових моделей висот.

До цифрового моделювання рельєфу долучають й інші методи комп'ютерного зору, не пов'язані безпосередньо з CNN.

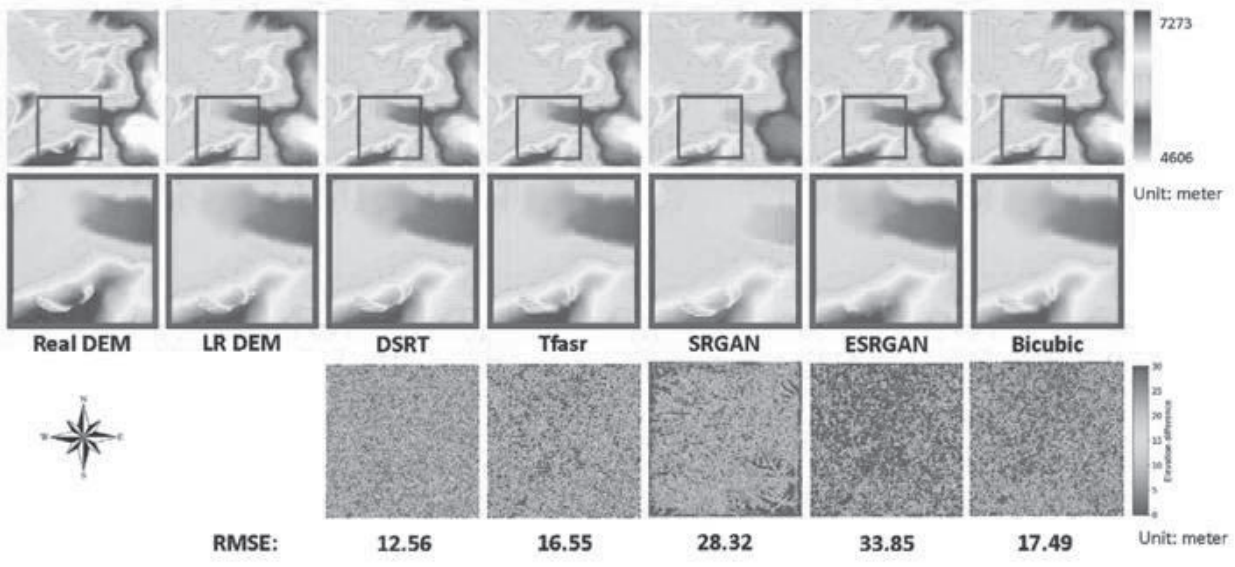


Рис. 2. Оцінка продуктивності на прикладі ЦМВ Гімалаїв з роздільністю 512 × 512, що згенеровані різними методами, та відмінності їх висот. Джерело: [13].

Наприклад, фотограмметричні алгоритми типу Structure-from-Motion (SfM), хоча й не належать до ML, використовують комп'ютерний зір для побудови 3D-хмар точок та ЦМР із знімків, отриманих з БПЛА або літаків. Багато сучасних фотограмметричних конвеєрів вже інтегрують ШІ на етапі поєднання зображень або фільтрації точок. Наприклад, відповідність ключових точок у SfM можна покращити за допомогою дескрипторів, створених нейронними мережами, що підвищує надійність роботи в складних ландшафтах.

Також досліджуються генеративні змагальні мережі (GAN), які можуть створювати реалістичні поверхні рельєфу або заповнювати пропуски в ЦМР — хоча цей напрям поки залишається нішевим.

У цілому, ГН та комп'ютерний зір дозволили перейти від ручного, експертного картографування до автоматизованого аналізу рельєфу — виявлення шаблонів, аномалій та змін у топографії з мінімальним втручанням людини.

Сучасні тенденції та виклики

Інтеграція та консолідація даних та інтеграція з багатьох джерел

Однією з провідних тенденцій є інтеграція різнорідних геопросторових даних для створення інформативніших моделей рельєфу. Жоден сенсор не дає повної картини, приміром LiDAR дозволяє точно фіксувати мікрорельєф під кронами дерев, оптичні зображення — забезпечують спектральний контекст, радар — «бачить» крізь хмари, а наземні сенсори додають точкові вимірювання. Сучасні ГІС поєднують ці джерела, щоб максимально використати їхні переваги.

Як зазначено вище, у фреймворку GeoAI було поєднано знімки високої роздільності з ЦМР у моделі ГН що дало змогу навчатись на обох типах даних одночасно [11]. Така мультиджерельна інтеграція суттєво покращує виявлення та класифікацію об'єктів рельєфу. Подібні підходи застосовують у створенні цифрових мозаїк висот, які поєднують LiDAR, ЦМР з фотограмметрії та глобальні моделі для узгодженості й заповнення прогалин.

Проте реалізація безшовного об'єднання супроводжується низкою викликів: дані мають різну роздільність, системи координат і рівень шумів. ГІС мають вирішувати задачі спільної геоприв'язки, узгодження роздільностей і зважування даних. Перспективним напрямом є адаптивне об'єднання за допомогою ML або трансферне навчання, коли знання з регіонів із насиченими даними (наприклад, з LiDAR-зондуванням) застосовуються до малодо-

сліджених територій (наприклад, лише з супутниковими ЦМР) [18], [19]. Об'єднання також розширюється на атрибутивні дані, наприклад, поєднання рельєфу з ґрунтовими картами чи земним покривом для створення комплексних ландшафтних моделей. Водночас важливо уникати поширення похибок одного джерела на всю модель, що робить цю сферу постійним предметом досліджень.

Обробка в реальному часі та великі дані

З удосконаленням технологій зондування й комунікаційних мереж зростає попит на моделі рельєфу, які оновлюються в реальному або майже реальному часі. БПЛА здатні за лічені хвилини отримати зображення чи LiDAR-дані місцевості, а мережі сенсорів IoT безперервно передають екологічні параметри, що відкриває можливості для створення динамічних моделей рельєфу, які автоматично оновлюються за надходженням нових даних, наприклад, оновлення карти затоплення на основі зростання рівня води, зафіксованого дощомірами.

Одна з тенденцій — це розроблення ГІС, здатних обробляти геопросторові великі дані «на льоту». Хмарні обчислення й розподілені ГІС-архітектури (наприклад, сервіси зображень або мапи на тайлах) є основними рушіями, що дозволяють масштабувати задачі аналізу. Наприклад, *Esri World Elevation Services* надає онлайн-доступ до глобальних ЦМР і підтримує динамічні функції, приміром, побудова профілів висоти чи тіньового рельєфу [20]. Шари висот, доступні в ArcGIS Online від Esri, зображено на рис. 3.

Також використовується потокова розподілена обробка, де моделюється стійкість схилів з інтеграцією даних від сенсорів (опади, вологість ґрунту тощо) з прогнозуванням на три доби вперед [15]. Автоматизовані пайплайни збирають дані, оновлюють геотехнічну модель і генерують прогноз стійкості схилів, що дозволяє своєчасно реагувати на потенційні обвали.

Водночас реальна обробка даних ГІС має низку технічних викликів. Дані рельєфу часто дуже обсяжні (хмари точок LiDAR можуть містити мільярди точок), тому алгоритми потребують оптимізації через GPU-прискорення або багаторівневі структури даних. Іноді використовуються спрощені або сурогатні моделі для швидких оцінок. Інша проблема — затримки даних і їхня синхронізація: потоки сенсорів можуть мати різні частоти та часові мітки, які потрібно узгоджувати.

Незважаючи на складнощі, попит на геопросторову інтелектуальну обробку в реальному часі зростає, особливо для реагування на надзвичайні ситуації, де актуальна інформація про рельєф має вирішальне значення.

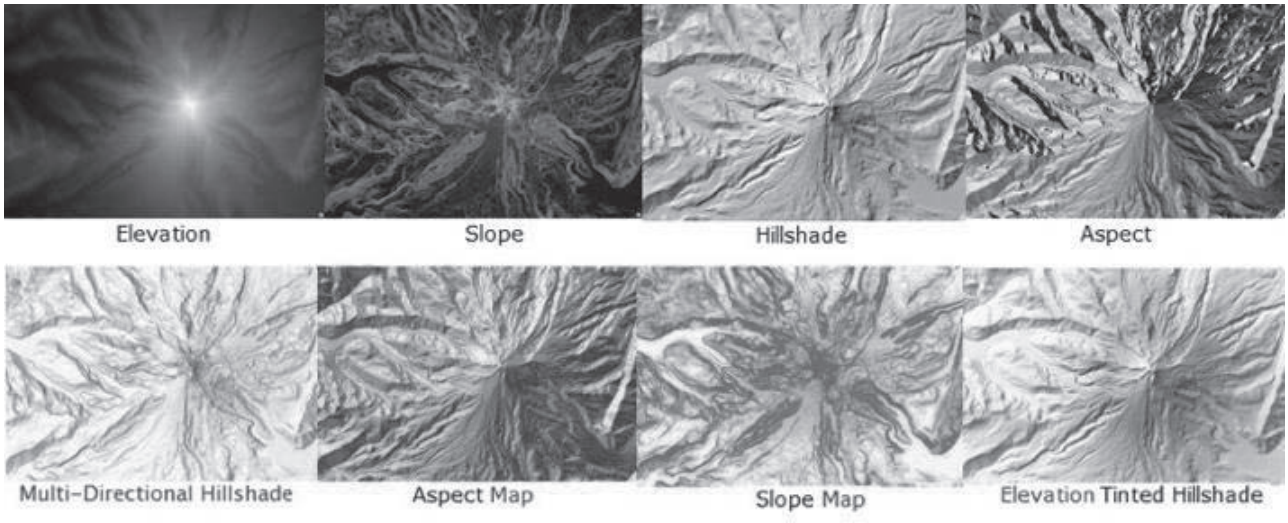


Рис. 3. Шари висот, доступні в ArcGIS Online від Esri. Джерело: [20]

Семантична сегментація та виділення об'єктів

У міру розвитку методів ШІ спостерігається чітка тенденція до уточнення трактування поняття рельєф через семантичну сегментацію та автоматичне виділення об'єктів. Замість того, щоб стврювати модель «оголеної» поверхні, фахівці прагнуть автоматично інтер-претувати значення форм рельєфу. Йдеться про виокремлення одиниць ландшафту (рівнини, долини, хребти), виявлення гідрологічних об'єктів (русло, басейн), а також ідентифікацію антропогенних структур (дамби, насипи доріг тощо).

Базовим методом, що дозволяє реалізувати цей підхід, є семантична сегментація ЦМР або ортофотознімків на основі ГН [10]. Кожна клітинка (піксель) у таких моделях маркується певним класом (наприклад, вода, ліс, забудова, тип ґрунту), що перетворює «сирі» дані на інформаційно насичені карти. Наприклад, сегментація дозволяє розрізнити вкриту рослинністю місцевість і відкритий ґрунт, що є важливим, як для екології, так і для точного формування ЦМР. Також такі моделі можуть виявляти пласкі заплавні ділянки, на відміну від крутих схилів, що використовується у моделях затоплення або ерозії. Приклад сегментації рельєфу зображено на рис. 4.

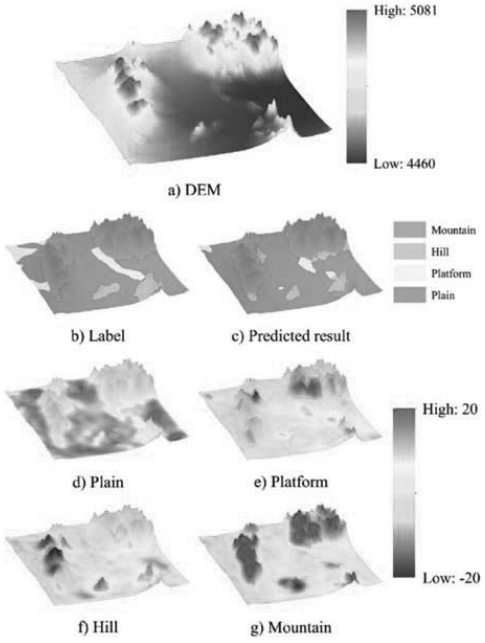


Рис. 4. Приклад сегментації рельєфу. Джерело: [10]

Однією з головних проблем цього підходу є необхідність у якісних навчальних даних, тобто створення валідних датасетів із підписаними класами рельєфу або поверхні, що є трудомістким і часто вимагає участі експертів (наприклад, щоб чітко визначити межу долини). Останні розробки спрямовані на застосування некерованого навчання або самонавчання на великих неанотованих ЦМР, щоб моделі могли спершу навчатися загального представлення рельєфу, а потім «підлаштовуватись» під конкретні задачі.

Інший важливий аспект семантичного аналізу — це виділення окремих об'єктів або структур із даних рельєфу. Приклади охоплюють автоматичне картографування зсувів на основі ЦМР, пошук карстових западин або воронок, а також виявлення штучних елементів, таких як насипи доріг або виїмки. Традиційні ГІС-підходи часто передбачали ручну векторизацію або застосування загальних фільтрів (наприклад, порогових значень кривизни для пошуку вглиблень). Сучасні ГІС використовують алгоритми виявлення об'єктів (наприклад, регіональні CNN), щоб автоматично знаходити ці особливості.

Основним викликом у цьому контексті є стійкість моделей до змін умов — природний рельєф дуже варіативний, і зсуви, зокрема, можуть мати різний вигляд залежно від геології чи рослинного покриву. Отже, моделі, натреновані на одному регіоні, можуть не узагальнюватись на інші без залучення великої кількості різноманітних тренувальних даних або адаптації до нових доменів. Незважаючи на це, тренд є очевидним: автоматизація виділення об'єктів рельєфу поступово замінює питання «яка тут висота» на питання «що це за форма рельєфу», відкриваючи шлях до глибшого розуміння простору в автоматизованому режимі.

Інтеграція з IoT та БПЛА

БПЛА та сенсори Інтернету речей (IoT) докорінно змінюють підходи до збору та використання даних рельєфу. Особливо потужним інструментом для локального моделювання місцевості стала технологія дронів. БПЛА, оснащені камерами або Li-

DAR, можуть оперативно обстежувати території й створювати високоточні ЦМР з роздільністю до сантиметрів чи дециметрів. Все більше інженерів і науковців використовують дрони для знімання топографічних даних і оперативного оновлення моделей рельєфу [16]. Концептуальне представлення застосування дронів та ГІС в управлінні розумними містами подано на рис.5.

Водночас виникають виклики: необхідність обробки великих обсягів зображень, забезпечення точності через наземні контрольні точки, а також проблема оклюзії (дрон не «бачить» під густим лісом, тому іноді потрібні LiDAR-дрони або наземне доповнення).

Інтеграція IoT-сенсорів є джерелом істинності даних для ГІІС моделювання рельєфу. Сенсори IoT — це такі як приймачі GNSS, тискоміри, екологічні сенсори, що забезпечують постійні точкові вимірювання, пов'язані з рельєфом. Наприклад, мережа GPS-сенсорів на зсувонебезпечному схилі може фіксувати мікрорухи ґрунту. Датчики рівня води вздовж річки здатні передавати дані до моделі затоплення на основі ЦМР. Таким чином, ЦМР стає частиною динамічної системи моніторингу даних. Прикладом є цифровий двійник стійкості схилів, у якому гідрологічні IoT-сенсори передають дані до моделі, яка з урахуванням геометрії рельєфу прогнозує зсуви. У цій системі вебсервіс щоденно автоматично отримує сенсорні дані та запускає модель стійкості, демонструючи тісну інтеграцію IoT та аналізу рельєфу [15].

Основні виклики інтеграції IoT і БПЛА — це різнорідність даних і складність побудови узгоджених систем. IoT-сенсори генерують часові ряди (наприклад, рівень води в часі), а ЦМР — просторові дані. Для поєднання необхідні просторово-часові моделі — наприклад, прив'язка кількості опадів до ЦМР для обчислення шляхів стікання. Необхідно забезпечити інтер-операбельність, зокрема, узгодження часових міток, координат сенсорів тощо. Також важливим є контроль якості: сенсори можуть виходити з ладу, потребувати фільтрації або калібрування перед використанням у моде-люванні.

Попри складнощі, інтеграція даних із сенсорів з ЦМР відкриває шлях до створення адаптивних ГІІС, зокрема, систем раннього попередження про природні катастрофи чи інфраструктури «розумного міста», що реагує на зміну довкілля в реальному часі.

Інші виклики

Окрім описаних вище тенденцій, сфера інтелектуального моделювання рельєфу стикається з низкою загальних проблем, які залишаються актуальними. Однією з них є якість даних і невизначеність**.** Вища роздільна здатність не гарантує вищої точності. Набори рельєфних даних можуть містити похибки — шумові відгуки LiDAR, артефакти інтерполяції, порожні області в ЦМР. ГІІС мають вміти враховувати ці похибки. Частина досліджень спрямована на кількісну оцінку та поши-

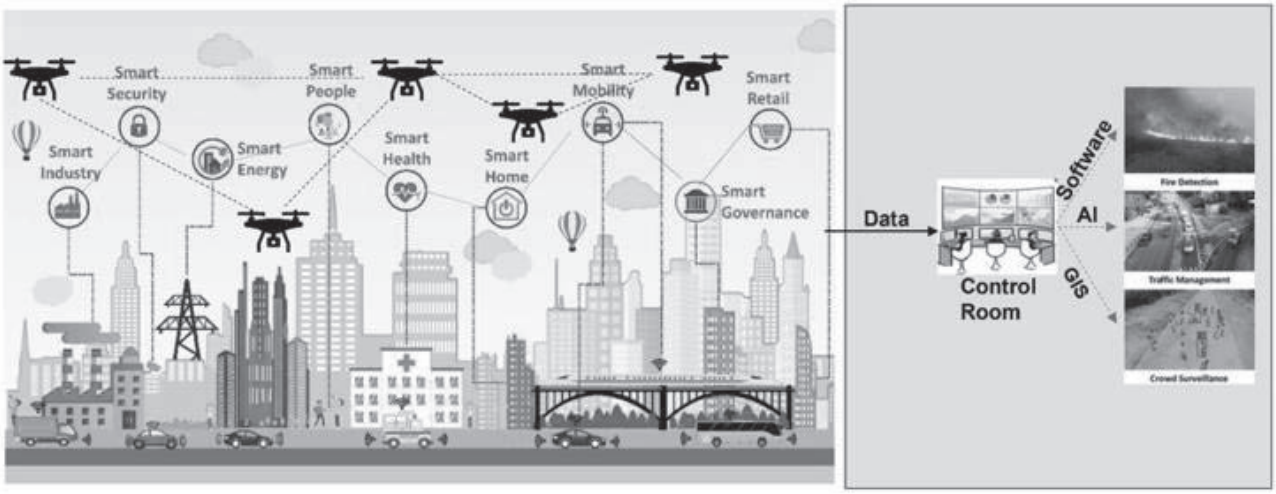


Рис. 5. Концептуальне представлення застосування дронів та ГІІС в управлінні розумними містами. Джерело: [16]

рення невизначеності в аналізі рельєфу, зокрема, використання ансамблів варіантів ЦМР у гідромоделюванні для оцінки варіативності результатів. Самі ж моделі ШІ також мають внутрішню невизначеність, тому зростає інтерес до пояснюваного ШІ (explainable AI) та оцінки довіри до результатів GeoAI — наприклад, наскільки можна покладатися на автоматично виявлений елемент рельєфу.

Ще один значний виклик — це масштабованість та зберігання. Високоточні ЦМР для великих територій породжують колосальні обсяги даних (терабайти). Для їх збереження й оброблення використовують просторові бази даних та хмарні сховища, але обмеження за вартістю та продуктивністю залишаються. Визначальними стають методи стиснення даних**,** рівневої деталізації (Level of Detail, LOD), наприклад, потокове відображення багаторівневих сіток для візуалізації та ефективної індексації (квадродерева чи октодерева для рельєфу). Просторові БД (наприклад, PostGIS, Oracle Spatial) забезпечують просторову індексацію (R-дерева) для швидкого пошуку рельєфом, а також зберігають растрові ЦМР у тайловому вигляді для оброблення великих територій.

Однак інтеграція таких баз даних із AI-пайплайнами перебуває нині в процесі розвитку, тобто, часто дані необхідно експортувати й обробляти в середовищах на кшталт Python, що додає затримки й нових обчислювальних витрат. Перспективною вважається тісніша інтеграція, як-от, вбудоване навчання моделей безпосередньо всередині БД, що може зменшити накладні витрати.

Також важливим є питання сумісності даних і застосування міжнародних стандартів. Геоінформаційна спільнота широко використовує стандарти (OGC, ISO) для форматів даних і вебсервісів. Для того щоб ГІС моделювання рельєфу стали частиною наявної інфраструктури ГІС, важливо забезпечити їхню інтероперабельність, зокрема, підтримку стандартних форматів ЦМР, вебкартографічних сервісів тощо. Ініціативи на кшталт Sensor Web Enablement (для IoT-сенсорів) та стандарти 3D-рельєфу від OGC спрямовано саме на забезпечення цієї сумісності. Попри ці виклики, вектор розвитку галузі чітко спрямовано у бік інтегрованих, автоматизованих та ГІС моделювання місцевості. Поєднання гетерогенних джерел даних, потужних алгоритмів ШІ та зв'язку в реальному часі відкриває нові можливості, які раніше здавались неможливими.

Висновки та перспективи

Цифрове моделювання рельєфу еволюціонувало від вузькоспеціалізованої задачі в межах ГІС до динамічної міждисциплінарної галузі на перетині геонаук, ШІ та ІТ-інфраструктури. Нині ГІС здатні будувати й аналізувати моделі рельєфу з небаченою деталізацією й автоматизацією. Сучасні методи МН та ГН застосовують до рельєфних даних, забезпечуючи, автоматичну класифікацію форм рельєфу, інтеграцію даних із різних сенсорів, прогнозування ризиків у реальному часі тощо.

Актуальні тренди вказують на інтеграцію даних, тобто поєднання різнорідних джерел (супутникові, аерофото, наземні сенсори); комбінування фізичних та data-driven моделей; інтеграцію аналізу рельєфу до систем підтримки ухвалення рішень (наприклад, платформи управління містом або реагування на надзвичайні ситуації).

Перспективні напрями розвитку ГІС. Вища роздільна здатність і великі геодані

З розвитком сенсорних технологій (щільніші LiDAR-сканування, супутникові системи, постійний моніторинг із дронів) рельєфні дані ставатимуть обсяжнішими та точнішими. Для оброблення таких обсягів потрібні нові рішення у сфері керування даними, зокрема, використання ШІ всередині БД чи обчислення на периферії (edge computing).

Моделювання рельєфу в реальному часі у 4D

Потенційно, моделювання можна здійснювати на різних типах даних: від статичних моделей до динамічних 4D-моделей, що враховують зміну рельєфу в часі. Наприклад, модель русла річки, яка оновлюється в реальному часі залежно від пе-

реміщення наносів. ШІ-моделі, здатні прогнозувати зміни рельєфу (наприклад, прибережну ерозію в умовах змін клімату), стануть особливо цінними.

Поглиблене семантичне розуміння рельєфу

Є потенціал для перетворення моделей рельєфу з геометричних у семантичні (цифрове знання про ландшафт), що може охоплювати онтології форм рельєфу та застосування ШІ, який пов'язує рельєф з екологічними або антропогенними процесами.

Інтеграція з ініціативами цифрових двійників

Багато урядів і компаній створюють цифрові двійники фізичних середовищ. Рельєф є базовим шаром такого двійника, а ГІС можуть стати «двигунами» цього шару, синхронізуючи його з реальністю через сенсори та дозволяючи моделювати сценарії для планування й управління. ГІС для цифрового моделювання рельєфу стають незамінними інструментами для вирішення реальних задач — від підвищення стійкості міст до надзвичайних ситуацій, до військових операцій і збереження природного середовища. Спираючись на теоретичну основу ГІС і використовуючи AI-технології, ці системи перетворюють сирі висотні дані на прикладну інформацію. Подальші наукові дослідження та розробки, особливо співпраця між геоінформатиками та фахівцями з ШІ/Data Science, ще більше розширять можливості моделювати й розуміти земну поверхню.

Отже, цифрове моделювання рельєфу нині перебуває на перетині геонаук, інформаційно-комунікаційних технологій і ШІ, формуючи міждисциплінарну парадигму просторового аналізу. ГІС здатні не лише точно моделювати місцевість, а й автоматично класифікувати її елементи, інтегрувати багатоджерельні дані, здійснювати прогнозування ризиків і ухвалення рішень у режимі реального часу. Зазначимо, що актуальні напрями розвитку вказаних досліджень охоплюють, зокрема, збільшення точності й обсягів геоданих; перехід до динамічних 4D-моделей; семантичне тлумачення рельєфу; впровадження цифрових двійників територій тощо. Використання AI у ГІС сприяє трансформації рутинного аналізу в автоматизовані, інтероперабельні системи підтримки рішень, що мають прикладне значення у містобудуванні, екології, обороні, управлінні надзвичайними ситуаціями. Подальший розвиток ГІС залежитиме від синергії між геоінформатиками, фахівцями з даних і розробниками ШІ.

References

1. NASA Earthdata. (n.d.). *Digital Elevation / Terrain Model (DEM)*. Retrieved from https://www.earthdata.nasa.gov/topics/land-surface/digital-elevation-terrain-model-dem
2. Satpalda. (2024). *The Role of Digital Terrain Models in Modern Geospatial Analysis*. Retrieved from https://satpalda.com/the-role-of-digital-terrain-models-in-modern-geospatial-analysis/
3. Yongze Song, Margaret Kalacska, Mateo Gašparović, Jing Yao, Nasser Najibi, Advances in geocomputation and geospatial artificial intelligence (GeoAI) for mapping, International Journal of Applied Earth Observation and Geoinformation, Volume 120, 2023, https://doi.org/10.1016/j.jag.2023.103300.
4. Sofia, Giulia, Anette Eltner, Efthymios Nikolopoulos, and Christopher Crosby. 2019. "Leading Progress in Digital Terrain Analysis and Modeling" *ISPRS International Journal of Geo-Information* 8, no. 9: 372. https://doi.org/10.3390/ijgi8090372
5. Li, Z., Zhu, C., & Gold, C. (2004). Digital Terrain Modeling: Principles and Methodology (1st ed.). CRC Press. https://doi.org/10.1201/9780203357132
6. Moore, I.D., Grayson, R.B. and Ladson, A.R. (1991), Digital terrain modelling: A review of hydrological, geomorphological, and biological applications. Hydrol. Process., 5: 3-30. https://doi.org/10.1002/hyp.3360050103
7. Tomislav Hengl, Hannes I. Reuter, Geomorphometry: Concepts, Software, Applications http://scholar.google.com/scholar_lookup?title=Geomorphometry:+Concepts,+Software,+Applications&author=Hengl,+T.&author=Reuter,+H.I.&publication_year=2008
8. Luca Piciullo, Minu Treesa Abraham, Ida Norderhaug Drøsdal, Erling Singstad Paulsen, An operational IoT-based slope stability forecast using a digital twin, Environmental Modelling & Software, Volume 183, 2025, https://doi.org/10.1016/j.envsoft.2024.106228.

9. Sizhe Wang, Wenwen Li. GeoAI in terrain analysis: Enabling multi-source deep learning and data fusion for natural feature detection, 2021, Computers, Environment and Urban Systems, p. 101715, https://doi.org/10.1016/j.compenvurbsys.2021.101715
10. Yang, Jiaqi & Xu, Jun & Lv, Yunshuo & Zhou, Chenghu & Zhu, Yunqiang & Cheng, Weiming. (2023). Deep learning-based automated terrain classification using high-resolution DEM data. International Journal of Applied Earth Observation and Geoinformation. 118. 103249. 10.1016/j.jag.2023.103249
11. Wang, Sizhe & Li, Wenwen. (2021). GeoAI in terrain analysis: Enabling multi-source deep learning and data fusion for natural feature detection. Computers, Environment and Urban Systems. 90. 101715. https://doi.org/10.1016/j.compenvurbsys.2021.101715
12. Luo, Wenjun & ma, Hongchao & Yuan, Jialin & Zhang, Liang & Ma, Haichi & Cai, Zhan & Zhou, Weiwei. (2023). High-Accuracy Filtering of Forest Scenes Based on Full-Waveform LiDAR Data and Hyperspectral Images. Remote Sensing. 15. 3499. https://doi.org/10.3390/rs15143499
13. Zhuoxiao Li, Xiaohui Zhu, Shanliang Yao, Yong Yue, Ángel F. García-Fernández, Eng Gee Lim, Andrew Levers, A large scale Digital Elevation Model super-resolution Transformer, International Journal of Applied Earth Observation and Geoinformation, Volume 124, 2023, https://doi.org/10.1016/j.jag.2023.103496
14. Yu, Jie & Li, Yangtenglong & Bai, Xuan & Yang, Ronghao & Cui, Mengxue & Wu, Haohao & Li, Zheng & Su, Fangzheng & Li, Ze & Liang, Taohuai & Yan, Hongliang. (2025). A DEM super resolution reconstruction method based on normalizing flow. Scientific Reports. 15. 10681. https://doi.org/10.1038/s41598-025-94274-w
15. TheHut-Nexus. 2024. "An Operational IoT-Based Slope Stability Forecast Using a Digital Twin." *TheHut-Nexus*. Accessed May 2, 2025. https://thehut-nexus.eu/an-operational-iot-based-slope-stability-forecast-using-a-digital-twin/
16. Quamar, Md Muzakkir, Baqer Al-Ramadan, Khalid Khan, Md Shafiullah, and Sami El Ferik. 2023. "Advancements and Applications of Drone-Integrated Geographic Information System Technology—A Review" *Remote Sensing* 15, no. 20: 5039. https://doi.org/10.3390/rs15205039
17. NATO Science and Technology Organization. (2007). *RTO-TR-SET-118: Terrain Characterization for Military Operations in Urban Terrain*. Retrieved from https://www.sto.nato.int/publications/STO Technical Reports/RTO-TR-SET-118/$$TR-SET-118-ALL.pdf
18. Maxwell, A. E., Odom, W. E., Shobe, C. M., Doctor, D. H., Bester, M. S., & Ore, T. (2023). Exploring the influence of input feature space on CNN-based geomorphic feature extraction from digital terrain data. *Earth and Space Science*, 10, e2023EA002845. https://doi.org/10.1029/2023EA002845
19. Ruiz-Lendínez, Juan J., Francisco J. Ariza-López, Juan F. Reinoso-Gordo, Manuel A. Ureña-Cámara, and Francisco J. Quesada-Real. 2023. "Deep Learning Methods Applied to Digital Elevation Models: State of the Art." *Geocarto International* 38 (1). https://doi.org/10.1080/10106049.2023.2252389
20. Esri. (2023). Introducing Esri's World Elevation Services. Retrieved from https://www.esri.com/arcgis-blog/products/analytics/analytics/introducing-esris-world-elevation-services

Одержано: 10.05.2025
Внутрішня рецензія отримана: 18.05.2025
Зовнішня рецензія отримана: 22.05.2025

Про авторів:

Плескач Валентина Леонідівна,
доктор економічних наук,
кандидат технічних наук.
http://orcid.org/0000-0003-0552-0972

Вдовиченко Владислав Володимирович,
аспірант факультету інформаційних технологій
http://orcid.org/0009-0005-0231-2299.

Місце роботи авторів:
[1,2] Факультет інформаційних технологій
Київського національного університету імені Т.Шевченка
v.pleskach64@gmail.com

УДК 004.8: 001.89 https://doi.org/10.15407/pp2025.03.091

С.В. Поперешняк

ВИКОРИСТАННЯ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ОПТИМІЗАЦІЇ МЕТОДОЛОГІЇ ТА ОРГАНІЗАЦІЇ НАУКОВИХ ДОСЛІДЖЕНЬ

Стрімка цифровізація науки висуває нові вимоги до методології та організації досліджень. Інструменти штучного інтелекту (ШІ) здатні прискорювати аналіз даних, підтримувати формування гіпотез і автоматизувати рутинні процедури, однак водночас загострюють питання відтворюваності, прозорості та академічної доброчесності. Мета роботи — обґрунтувати та апробувати підходи до інтеграції технологій ШІ для оптимізації методології й організації наукових досліджень у сфері цифрового розвитку. В роботі було застосовано системний аналіз сучасних практик, математичне моделювання процесів (оптимізаційні постановки вибору режимів «людина/ШІ/гібрид», байєсівське оцінювання достовірності, регуляризація упередженості та інтерпретованості, POMDP-планування наукових циклів), а також проєктування модульної архітектури підтримки досліджень. Емпіричну перевірку здійснено на прототипах робочих потоків: автоматизований огляд літератури, інтелектуальна обробка експериментальних даних, підготовка публікаційних матеріалів. Запропоновано інтегровану модель управління науковим процесом, що поєднує: (i) формальний вибір ролей людини й ШІ під обмеження ресурсів і якості; (ii) агрегування доказів із людських і машинних каналів через байєсівську схему; (iii) одночасне обмеження алгоритмічної упередженості та підвищення пояснюваності; (iv) стратегічне POMDP-планування експериментів з урахуванням ризиків і вартості. В роботі було показано, що використання запропонованої моделі забезпечує скорочення часу на підготовку оглядів і аналіз даних, зростання відтворюваності висновків та прозорості рішень, а також зменшення ризиків, пов'язаних із «чорною скринькою» і зміщеннями даних. Сформульовано практичні рекомендації щодо впровадження ШІ у дослідницькі підрозділи: регламенти розкриття використання ШІ, контроль якості даних, вимоги до пояснюваності моделей і ролі дослідника як відповідального інтерпретатора. ШІ доцільно розглядати як інструмент підсилення наукової праці. Інтеграція запропонованих моделей у методологію та організацію досліджень підвищує ефективність, відтворюваність і етичну надійність наукових результатів, відкриваючи шлях до масштабованих, ресурсоефективних дослідницьких практик.

Ключові слова: штучний інтелект; методологія наукових досліджень; організація досліджень; відтворюваність; пояснюваність; байєсівські методи; оптимізація; POMDP; упередженість даних; академічна доброчесність.

S. Popereshnyak

USING ARTIFICIAL INTELLIGENCE TECHNOLOGIES TO OPTIMIZE METHODOLOGY AND ORGANIZATION OF SCIENTIFIC RESEARCH

The rapid digitalization of science puts forward new requirements for the methodology and organization of research. Artificial intelligence (AI) tools are able to accelerate data analysis, support hypothesis generation, and automate routine procedures, but at the same time exacerbate issues of reproducibility, transparency, and academic integrity. The purpose of the work is to substantiate and test approaches to integrating AI technologies to optimize the methodology and organization of scientific research in the field of digital development. The work applied a systematic analysis of modern practices, mathematical modeling of processes (optimization of the choice of modes "human/AI/hybrid", Bayesian assessment of reliability, regularization of bias and interpretability, POMDP-planning of scientific cycles), as well as the design of a modular architecture for supporting research. Empirical testing was carried out on prototypes of workflows: automated literature review, intelligent processing of experimental data, preparation of publication materials. An integrated model of scientific process management was proposed, combining: (i) formal selection of human and AI roles under resource and quality constraints; (ii) aggregation of evidence from human and machine channels through a Bayesian scheme; (iii) simultaneous limitation of algorithmic bias and increase of explainability; (iv) strategic POMDP-planning of experiments taking into account risks and costs. The paper showed that the use of the proposed model reduces the time for preparing reviews and analyzing data, increases the reproducibility of conclusions and transparency of decisions, as well as reduces the risks associated with the “black box” and data bias. Practical recommendations for implementing AI in research units are formulated: regulations for

disclosing the use of AI, data quality control, requirements for the explainability of models, and the role of the researcher as a responsible interpreter. AI should be considered as a tool for strengthening scientific work. The integration of the proposed models into the methodology and organization of research increases the efficiency, reproducibility, and ethical reliability of scientific results, opening the way to scalable, resource-efficient research practices.
Key words: artificial intelligence; scientific research methodology; research organization; reproducibility; explainability; Bayesian methods; optimization; POMDP; data bias; academic integrity.

Вступ

Сучасна наука перебуває у стані глибокої трансформації під впливом цифрових технологій, серед яких провідне місце посідають методи штучного інтелекту (ШІ). Використання алгоритмів машинного навчання, обробки природної мови, комп'ютерного зору та інтелектуальних систем управління дедалі частіше стає невід'ємною складовою досліджень у різних галузях знань. Це зумовлено потребою у швидкому опрацюванні великих масивів даних, підвищенні точності аналітики, автоматизації рутинних процедур і забезпеченні відтворюваності експериментів.

Інтеграція ШІ у дослідницьку практику відкриває нові можливості для генерації знань, які ще донедавна видавалися недосяжними, та дозволяє оптимізувати організаційні процеси на всіх етапах — від формування гіпотези до підготовки публікації. Водночас таке впровадження потребує переосмислення традиційної методології та розробки нових підходів до управління знаннями. Безсистемне або суто технічне використання інтелектуальних інструментів може призвести до методологічних похибок, викривлення результатів чи зниження довіри до наукових висновків.

Отже, актуальним завданням є комплексний аналіз можливостей, переваг і обмежень застосування ШІ в контексті методології та організації наукових досліджень. Цифрові технології мають розглядатися не як заміна, а як інструмент підсилення наукової праці, що здатний забезпечити баланс між автоматизацією та академічною доброчесністю.

Аналіз останніх досліджень і публікацій.

У сучасних наукових дослідженнях дедалі більше уваги приділяється використанню технологій штучного інтелекту. Роботи західних авторів демонструють, що ШІ здатний не лише прискорювати обробку великих масивів даних, а й формувати нові підходи до генерації гіпотез та побудови моделей наукових експериментів [1]. Водночас підкреслюються ризики, пов'язані з відтворюваністю результатів і залежністю від алгоритмічних рішень [2].

Окрему увагу приділяють етичним викликам, зокрема, питанням прозорості використання ШІ та необхідності обов'язкового розкриття його застосування у наукових роботах [3]. Досліджується також потенціал ШІ у вдосконаленні рецензування, де інтелектуальні системи допомагають підвищити швидкість та якість експертної оцінки, хоча й виникають побоювання щодо збереження академічної доброчесності [4].

В українському науковому просторі акцент робиться на можливостях ШІ у сфері освіти та наукової комунікації. Зокрема, у [5] узагальнено сучасні тенденції розвитку технологій, а також перспективи їх інтеграції в освітній процес. Вітчизняні дослідники також підкреслюють як переваги використання інтелектуальних систем (швидкий доступ до знань, автоматизація рутинних процесів), так і ризики, пов'язані з можливими методологічними похибками та алгоритмічними викривленнями [6; 7].

Таким чином, аналіз останніх публікацій засвідчує, що питання інтеграції ШІ в методологію та організацію наукових досліджень перебуває у фокусі як міжнародної, так і національної наукової спільноти, а актуальними залишаються проблеми ефективності, прозорості та етичної відповідальності.

Мета та завдання дослідження. Метою статті є обґрунтування та розробка концептуальних підходів до інтеграції тех-

нологій штучного інтелекту в методологію та організацію наукових досліджень у сфері цифрового розвитку. Особлива увага приділяється оцінці ефективності ШІ як інструмента оптимізації дослідницьких процесів, підвищення відтворюваності результатів та забезпечення академічної доброчесності.

Виклад основного матеріалу.

Сучасні дослідження активно інтегрують інструменти штучного інтелекту, які змінюють процес здобуття й організації знань. Машинне навчання дозволяє аналізувати великі масиви даних із високою швидкістю, забезпечуючи нові можливості для прогнозування та виявлення закономірностей у різних галузях. Водночас воно породжує питання відтворюваності результатів та залежності від коректності навчальних вибірок.

Зростає роль технологій обробки природної мови, які полегшують роботу з науковими текстами, але водночас несуть ризики методологічних спрощень і некритичного використання автоматично створених матеріалів. Подібні виклики виникають і у сфері комп'ютерного зору, що значно прискорює аналіз візуальних даних, проте часто перетворює алгоритми на «чорні скриньки», складні для інтерпретації.

Інтелектуальні системи управління та цифрові сервіси підтримки дослідницької діяльності оптимізують роботу лабораторій і процес публікацій, але водночас створюють ризик надмірної залежності від автоматизації.

Таким чином, ШІ відкриває нові перспективи для науки, проте його використання потребує критичного осмислення, збереження академічної доброчесності та розробки методологічних засад, що гарантують баланс між технічною ефективністю та науковою відповідальністю.

Проте, використання штучного інтелекту у науковій практиці поєднує значний потенціал з низкою методологічних ризиків. Для кращого розуміння цього балансу доцільно узагальнити ключові напрями застосування ШІ разом із можливостями, які вони відкривають, та обмеженнями, що супроводжують їх використання (рис. 1, табл. 1).

Таблиця1.

Можливості та ризики використання ШІ в науці

Напрям застосування ШІ	Можливості	Ризики та обмеження
Обробка великих даних	Швидке опрацювання терабайтів інформації; виявлення прихованих закономірностей	Помилки через «шумні» дані; залежність від якості вибірки
Комп'ютерний зір та аналіз сигналів	Виявлення деталей, непомітних для людини; автоматизований моніторинг	Чутливість до умов середовища, високі обчислювальні вимоги
Обробка природної мови	Систематизація наукової літератури; автоматизоване виявлення трендів	Ризик некоректної інтерпретації термінів; проблема доброчесності при генерації текстів
Формування гіпотез і прогнозів	Моделювання сценаріїв; підвищення точності передбачень	«Чорна скринька» моделей, складність перевірки коректності
Організація дослідницької діяльності	Автоматизація підготовки звітів, публікацій, планів	Надмірна залежність від технічних систем, загроза втрати критичного аналізу

Рис. 1. Взаємодія можливостей і ризиків ШІ у наукових дослідженнях

Узагальнене співвідношення між перевагами й викликами використання ШІ можна також візуалізувати у вигляді схеми, яка демонструє взаємодію потенціалу технологій та супровідних ризиків.

Отже, сучасні інструменти штучного інтелекту поєднують у собі колосальний потенціал і водночас суттєві обмеження. Вони здатні не лише оптимізувати окремі етапи дослідження, а й трансформувати саму структуру наукової діяльності. Однак для цього потрібне критичне осмислення їхньої ролі та формування нових методологічних рамок, які гарантуватимуть надійність і відтворюваність наукових результатів.

Основні напрями застосування ШІ в дослідженнях.

Сучасна наукова діяльність дедалі більше потребує інтеграції інтелектуальних цифрових технологій, серед яких штучний інтелект (ШІ) посідає провідне місце. Його застосування в методології та організації досліджень можна розглядати за кількома ключовими напрямами, що визначають характер трансформацій у науковому середовищі. У сукупності ці напрями відображають комплексний вплив ШІ на сучасну науку — від зміни методологічних засад до трансформації організаційних практик (табл. 2.). Важливим завданням залишається гармонізація технічних можливостей і традиційних наукових принципів, що забезпечує збалансованість між інноваціями та академічною надійністю.

Інтеграція штучного інтелекту у методологію та організацію наукових досліджень не є лінійним процесом. Кожен етап природно переходить у наступний, утворюючи замкнуте коло безперервного вдосконалення знань (Рис. 2). Такий підхід дозволяє підвищувати якість досліджень, але водночас вимагає постійного контролю за ризиками автоматизації.

Циклічність демонструє, що наука в умовах цифрової трансформації перетворюється на інтерактивний і безперервний процес: результати одного етапу стають підґрунтям для нового циклу. ШІ виступає каталізатором цього процесу, прискорюючи обіг знань, але водночас посилюючи вимоги до методологічної строгості та контролю якості.

Таблиця 2.

Основні напрями застосування ШІ у наукових дослідженнях

Напрям застосування	Можливості	Ризики / Обмеження
1. Автоматизація пошуку та систематизації знань	Швидкий аналіз великих обсягів публікацій; формування бібліографічних оглядів; виявлення прихованих зв'язків між дослідженнями	Залежність від алгоритмічних рекомендацій; втрата критичного осмислення джерел
2. Генерація та перевірка гіпотез	Прогнозування на основі великих даних; підтримка вибору напрямів дослідження	Обмежена пояснюваність («чорна скринька»); можливі методологічні викривлення
3. Інтелектуальна обробка експериментальних даних	Висока точність і швидкість аналізу даних; автоматизація роботи з різнорідними джерелами (зображення, сенсори)	Ризик помилкових інтерпретацій; потреба у валідації результатів
4. Оптимізація організаційних процесів	Управління проєктами, ресурсами, координація команд; підвищення ефективності наукових колаборацій	Ймовірність упереджених управлінських рішень; питання етики та прозорості
5. Підтримка академічної доброчесності	Виявлення плагіату, некоректних цитувань, фабрикацій даних; підвищення довіри до результатів	Формалізація оцінки наукової роботи; ризик хибнопозитивних результатів
6. Автоматизована підготовка наукових матеріалів	Швидке створення звітів, статей, презентацій; зменшення рутинної роботи	Питання авторства та оригінальності; зниження ролі творчого внеску дослідника
7. Віртуальні лабораторії та моделювання	Економія ресурсів; можливість моделювати експерименти та створювати цифрові двійники	Потреба у верифікації моделей у реальних умовах; ризик надмірної довіри до симуляцій

Рис. 2. Циклічна модель застосування ШІ у наукових дослідженнях

Методологічні та етичні виклики.

Впровадження технологій штучного інтелекту у методологію та організацію наукових досліджень відкриває нові горизонти для автоматизації та підвищення продуктивності наукової праці. Водночас цей процес супроводжується низкою викликів, які стосуються не лише технічної реалізації, а й принципових основ наукового пізнання та академічної доброчесності.

З методологічної точки зору основним викликом є ***прозорість і відтворюваність результатів***. Алгоритми машинного навчання часто функціонують як «чорні скриньки», ускладнюючи пояснення отриманих висновків. Це створює ризик зниження довіри до результатів, особливо в міждисциплінарних дослідженнях, де важливо продемонструвати логіку обчислень і коректність інтерпретацій. Іншою проблемою є ***упередженість даних***, яка неминуче переноситься на моделі та може призвести до спотворення результатів. Таким чином, виникає потреба у створенні стандартів контролю якості даних, алгоритмів та їхньої валідації.

Ще один аспект, пов'язаний із ***балансом між автоматизацією та творчістю***. Використання генеративних моделей чи інтелектуальних систем для формування гіпотез або підготовки текстів наукових статей здатне значно прискорити дослідження. Проте надмірна залежність від автоматизованих інструментів може призвести до зниження оригінальності мислення та «стандартизації» наукових результатів. Виникає ризик того, що науковець із суб'єкта пізнання перетвориться на спостерігача за роботою алгоритмів.

Етичні виклики стосуються насамперед ***академічної доброчесності***. Використання ШІ у написанні текстів, аналізі джерел чи візуалізації даних ставить питання про межу між «допомогою» та «авторством». Необхідно чітко визначити правила цитування результатів, згенерованих алгоритмами, а також умови верифікації даних, отриманих за допомогою автоматизованих систем. Особливої уваги потребує проблема ***плагіату та фальсифікацій***, які можуть бути замасковані під результати роботи ШІ.

Крім того, варто враховувати ***питання етичного використання даних***. Дослідження, що ґрунтуються на великих масивах інформації, часто включають персональні або чутливі дані. Відповідальність за їх збереження та анонімізацію покладається не лише на дослідника, а й на системи, якими він користується. Порушення цього принципу може спричинити не лише репутаційні, а й правові наслідки.

Щоб чіткіше окреслити потенційні проблеми інтеграції ШІ у наукові дослідження, узагальнимо їх у вигляді порівняльної таблиці, яка відображає виклики, їхні наслідки та можливі шляхи подолання.

Отож, методологічні та етичні виклики використання ШІ у наукових дослідженнях можна охарактеризувати як **подвійну дилему**:

- з одного боку, вони стимулюють пошук нових стандартів наукової діяльності,
- а з іншого — потребують переосмислення ролі дослідника у взаємодії з цифровими технологіями.

Вирішення цих питань неможливе без комплексного підходу, який поєднує технічні, філософські, правові та організаційні аспекти.

У зв'язку з цим постає потреба не лише у якісному аналізі можливостей і ризиків, а й у формалізації самого процесу інтеграції ШІ у дослідницьку діяльність. Математичні моделі дозволяють виразити ключові закономірності організації наукової роботи у вигляді оптимізаційних завдань, забезпечити відтворюваність результатів і задати критерії для порівняння альтернативних рішень.

Таблиця 3.

Методологічні та етичні виклики використання ШІ у наукових дослідженнях

Виклик	Потенційні наслідки	Можливі шляхи подолання
Непрозорість алгоритмів («чорна скринька»)	Зниження довіри до результатів; складність інтерпретації висновків	Використання explainable AI (XAI), розробка прозорих моделей, документування процесів
Упередженість даних	Спотворення результатів; методологічні помилки	Стандарти контролю якості даних, багатоджерельна валідація, балансування наборів
Автоматизація vs. творчість	Зниження оригінальності мислення; стандартизація результатів	Збереження ролі дослідника як інтерпретатора; регулювання обсягу автоматизації
Проблеми академічної доброчесності	Плагіат, викривлення авторства, сумнівні публікації	Введення чітких правил цитування ШІ, перевірка унікальності текстів
Фальсифікації даних	Поширення недостовірних результатів; зниження наукової репутації	Верифікація результатів, незалежне рецензування, відкриті репозиторії даних
Етичне використання даних	Ризики розкриття персональних або чутливих даних; правові наслідки	Анонімізація, дотримання GDPR та локальних стандартів захисту даних
Залежність від алгоритмів	Зміщення ролі дослідника з суб'єкта на спостерігача	Формування «гібридної моделі» дослідження: людина + ШІ як партнери

Математична модель інтеграції ШІ у методологію та організацію наукових досліджень.

Сучасні підходи до використання штучного інтелекту в організації наукових досліджень не можуть обмежуватися лише якісними описами. Для забезпечення відтворюваності, обґрунтованості та оптимального використання ресурсів необхідно формалізувати процеси у вигляді математичних моделей. Нижче подано кілька взаємопов'язаних моделей, що дозволяють розглядати завдання наукової діяльності як оптимізаційні проблеми із врахуванням якості, ризиків, інтерпретованості та етичних обмежень.

Модель 1. Оптимізація вибору режиму виконання завдань

Кожне дослідницьке завдання може виконуватися людиною (H), штучним інтелектом (A) або у гібридному режимі ($H+A$). Вводяться змінні:

$$x_i^{(H)}, x_i^{(A)}, x_i^{(HA)} \in \{0,1\},$$
$$x_i^{(H)} + x_i^{(A)} + x_i^{(HA)} \leq 1,$$

що відповідає способу виконання завдання i.

Цільова функція враховує наукову цінність v_i, точність α, ризики r_i та штраф за низьку інтерпретованість:

$$\max \sum_{i=1}^{n} v_i \cdot \alpha_i (1 - r_i) - \lambda(1 - \iota_i).$$

Обмеження накладаються на ресурси:

$$\sum_{i=1}^{n} \bar{d}_i \le T_{\max},$$
$$\sum_{i=1}^{n} \bar{c}_i \le C_{\max},$$
$$\sum_{i=1}^{n} \alpha_{ik} x_i \le R_k.$$

Отож, модель дозволяє формалізувати питання «які завдання і в якому режимі виконувати», забезпечуючи баланс між якістю, часом та витратами.

Модель 2. Байєсівське оцінювання достовірності результатів

Кожна гіпотеза $\theta_i \in \{0,1\}$ має апріорну ймовірність істинності π_i. Людські та машинні результати комбінуються через правдоподібності:

$$P(\theta_i = 1 | D^{(H)}, D^{(A)}) = \frac{\pi_i L^{(H)}(D^{(H)}|1) L^{(A)}(D^{(A)}|1)}{\pi_i L^{(H)}(D^{(H)}|1) L^{(A)}(D^{(A)}|1) + (1 - \pi_i) L^{(H)}(D^{(H)}|0) L^{(A)}(D^{(A)}|0)}.$$

Таким чином, рішення про публікацію чи продовження дослідження ухвалюється не лише на основі кількісних даних, а й з урахуванням інтегрованої доказовості.

Модель 3. Регуляризація упередженості та інтерпретованості

У процесі навчання моделі ШІ мінімізується функція:

$$\min_w \mathcal{L}_{task}(w) + \mu R_{bias}(w) + \eta R_{expl}(w),$$

де R_{bias} — штраф за статистичні зсуви, а R_{expl} — за відсутність інтерпретованості. Це забезпечує баланс між точністю, етичністю та прозорістю досліджень.

Модель 4. Планування досліджень як POMDP

Науковий процес моделюється як задача частково спостережуваного марковського процесу (POMDP), де стан s_t відображає поточний рівень підтвердження гіпотез, ресурси та ризики.

Нагорода визначається як:

$$R(s_t, a_t) = \Delta\text{цінності знання} - \text{витрати} - \kappa \cdot \text{ризик}.$$

Оптимальна політика $\pi*$ забезпечує максимізацію довгострокової користі від експериментів у разі обмежених ресурсів і врахуванні етичних норм.

Розглянемо схему, що демонструє взаємодію чотирьох моделей: від вибору завдань і режиму їх виконання — до оцінки достовірності результатів, контролю упередженості ШІ та стратегічного планування дослідницьких циклів (Рис. 3).

Отже, математична модель не лише формалізує ключові етапи інтеграції ШІ у науковий процес, а й забезпечує основу для побудови інструментів аналізу, які враховують як ефективність, так і етичні вимоги до досліджень.

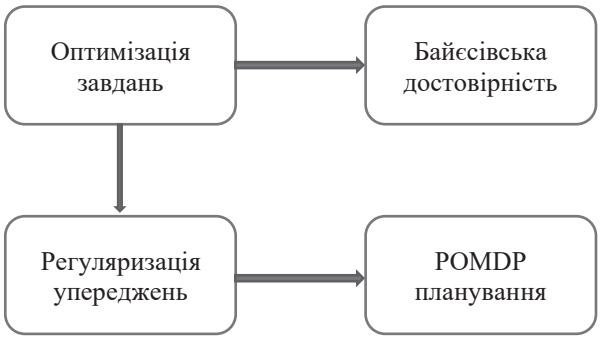


Рис. 3. Схема зв'язків моделей

Розроблені математичні підходи до інтеграції ШІ у методологію та організацію наукових досліджень мають практичний потенціал у різних аспектах дослідницького процесу. Для підтвердження їхньої доцільності розглянемо кілька сценаріїв, що ілюструють роботу моделей у реальних умовах (Табл. 4).

Таблиця 4.

Приклади застосування моделей у різних галузях

Модель	Сфера застосування	Очікуваний результат
Оптимізація завдань	Аналіз наукових публікацій	Мінімізація часу обробки даних
Байєсівська достовірність	Біомедичні дослідження	Підвищення надійності гіпотез
Регуляризація упереджень	Соціологія, освіта	Зменшення впливу системних викривлень
POMDP планування	Міждисциплінарні проекти	Адаптивне управління ресурсами

Як бачимо, математичні моделі не є суто теоретичними, а мають безпосереднє прикладне значення. Їх інтеграція у дослідницьку практику дозволяє зробити процес більш ефективним, прозорим та етично обґрунтованим.

Обговорення результатів та перспективи розвитку

Отримані результати демонструють, що інтеграція технологій штучного інтелекту у методологію та організацію наукових досліджень може суттєво підвищити ефективність роботи дослідницьких колективів. Застосування оптимізаційних моделей дозволяє структурувати процеси планування, зменшуючи часові витрати на рутинні операції та формуючи прозорі алгоритми вибору дослідницьких завдань. Байєсівські методи підвищують достовірність висновків, забезпечуючи адаптивну оцінку даних у динамічних і неповних інформаційних умовах. Регуляризаційні підходи роблять результати більш об'єктивними, знижуючи вплив прихованих викривлень і підвищуючи рівень академічної доброчесності.

Особливу увагу заслуговує використання моделей динамічного управління науковим процесом, таких як POMDP, що відкривають можливість створення систем підтримки ухвалення рішень у масштабних міждисциплінарних проєктах. Це важливо в умовах глобалізації науки, коли дослідження дедалі частіше проводяться у великих міжнародних колаборціях.

Разом з тим результати підкреслюють наявність низки викликів. Серед них — необхідність розробки стандартів використання ШІ у науковій діяльності, запобігання надмірній залежності від автоматизованих систем, а також вироблення механізмів контролю прозорості алгоритмів. Вирішення цих проблем вимагає міждисциплінарного підходу, що поєднує зусилля фахівців у галузі комп'ютерних наук, методології, етики та права.

Перспективним напрямом є створення інтегрованих платформ, які поєднуватимуть аналіз даних, управління експериментами та автоматизовану підготовку результатів до публікації. Такі системи здатні забезпечити не лише ефективність, а й відтворюваність наукових досліджень. Додаткові можливості відкриває застосування генеративних моделей ШІ для пошуку нових гіпотез, формування експериментальних сценаріїв та симуляції результатів ще до проведення реальних дослідів.

У цілому можна стверджувати, що використання штучного інтелекту у методології та організації наукових досліджень є не лише інструментом підвищення ефективності, а й фактором трансформації самої наукової практики. Подальші дослідження мають бути зосереджені на розробці прозорих, етично обґрунтованих та масштабованих рішень, здатних інтегруватися у різні галузі знань.

Висновки.

У статті проаналізовано сучасний стан і тенденції використання штучного інтелекту в науковій практиці, зокрема, в аспектах методології досліджень та організації наукової діяльності. Визначено ключові можливості застосування алгоритмів машинного навчання, обробки природної мови, систем комп'ютерного зору та сенсорного злиття даних для прискорення обробки інформації, формування гіпотез, підвищення точності прогнозування та автоматизації організаційних процесів.

Окремо акцентовано увагу на ризиках і обмеженнях використання ШІ, пов'язаних із відтворюваністю результатів, академічною доброчесністю, прозорістю алгоритмів та етичними викликами. Показано, що ефективна інтеграція інтелектуальних інструментів у дослідницький процес можлива лише за умови поєднання технічної ефективності з дотриманням наукових стандартів.

Запропоновано узагальнену модель інтеграції ШІ в методологію наукових досліджень, яка базується на математичному описі процесу обробки даних, адаптивних механізмах прогнозування та балансі між автоматизацією й людським контролем.

В роботі обґрунтовано доцільність впровадження ШІ у навчально-наукову діяльність, зокрема, в аспектах автоматизації аналітики, управління дослідницькими процесами, моделювання та формування цифрових компетентностей студентів. Це сприятиме підвищенню якості досліджень, масштабованості наукових експериментів та інтеграції в міжнародний академічний простір.

Отримані результати підтверджують, що штучний інтелект є не лише технічним інструментом, а й чинником трансформації методології науки, здатним забезпечити поєднання ефективності, відтворюваності та інноваційності у сучасних дослідженнях.

Література

1. Gao J., Wang D. Quantifying the Benefit of Artificial Intelligence for Scientific Research [Електронний ресурс] // *arXiv preprint arXiv:2304.10578*. – 2023. – DOI: 10.48550/arXiv.2304.10578.
2. The Potential and Concerns of Using AI in Scientific Research [Електронний ресурс] // *Frontiers in Artificial Intelligence*. – 2023. – DOI: 10.3389/frai.2023.10636627.
3. Disclosing artificial intelligence use in scientific research [Електронний ресурс] // *Accountability in Research*. – 2025. – Vol. 32, Iss. 3. – P. 176–190. – DOI: 10.1080/08989621.2025.2481949.
4. Artificial intelligence in peer review: Enhancing efficiency while maintaining integrity [Електронний ресурс] // *Journal of Korean Medical Science*. – 2025. – Vol. 40, e92. – DOI: 10.3346/jkms.2025.40.e92.
5. Штучний інтелект у науці та освіті (AISE 2024): збірник матеріалів міжнародної наукової конференції (Київ, 1–2 березня 2024 р.) [Електронний ресурс] / упоряд. Є. Завальнюк, В. Матусевич, В. Коваленко. – Київ: УкрІНТЕІ, 2024. – DOI: 10.35668/978-966-479-141-7. – Режим доступу: http://visnyk.naps.gov.ua
6. Дашко І., Череп О., Михайліченко Л. Розвиток штучного інтелекту: переваги та недоліки [Електронний ресурс] // *Economy & Society*. – 2024. – DOI: 10.32782/2524-0072/2024-67-31.
7. Куцак Л. В. Штучний інтелект у сучасній освіті: перспективи застосування та виклики [Електронний ресурс] // *Вісник ВНЗ*. – 2025. – DOI: 10.31652/2412-1142-2024-74-27-37. – Режим доступу: http://vspu.net

References

1. Gao, J., & Wang, D. (2023). Quantifying the Benefit of Artificial Intelligence for Scientific Research. *arXiv preprint* arXiv:2304.10578. https://doi.org/10.48550/arXiv.2304.10578
2. The Potential and Concerns of Using AI in Scientific Research. (2023). *Frontiers in Artificial Intelligence.* https://doi.org/10.3389/frai.2023.10636627
3. Disclosing artificial intelligence use in scientific research. (2025). *Accountability in Research, 32*(3), 176–190. https://doi.org/10.1080/08989621.2025.2481949
4. Artificial intelligence in peer review: Enhancing efficiency while maintaining integrity. (2025). *Journal of Korean Medical Science, 40*, e92. https://doi.org/10.3346/jkms.2025.40.e92
5. Zavalnyuk, Ye., Matusevych, V., & Kovalenko, V. (Comps.). (2024). *Shtuchnyi*

intelekt u nautsi ta osviti (AISE 2024): zbirnyk materialiv mizhnarodnoi naukovoi konferentsii (Kyiv, 1–2 bereznia 2024 r.) [Artificial Intelligence in Science and Education (AISE 2024): Proceedings] (Electronic resource). Kyiv: UkrINTEI. https://doi.org/10.35668/978-966-479-141-7. Retrieved from http://visnyk.naps.gov.ua [in Ukrainian].

6. Dashko, I., Cherep, O., & Mykhailichenko, L. (2024). Rozvytok shtuchnoho intelektu: perevahy ta nedoliky [Development of artificial intelligence: advantages and disadvantages]. *Economy & Society.* https://doi.org/10.32782/2524-0072/2024-67-31 [in Ukrainian].
7. Kutsak, L. V. (2025). Shtuchnyi intelekt u suchasnii osviti: perspektyvy zastosuvannia ta vyklyky [Artificial intelligence in modern education: prospects and challenges]. *Visnyk VNZ.* https://doi.org/10.31652/2412-1142-2024-74-27-37. Retrieved from http://vspu.net [in Ukrainian].

Одержано: 17.10.2025
Внутрішня рецензія отримана: 25.10.2025
Зовнішня рецензія отримана: 24.10.2025

Про авторів:

Поперешняк Світлана Володимирівна,
к.ф.-м.н., доцент
http://orcid.org/0000-0002-0531-9809.

Місце роботи авторів:

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»,
тел. +38-098-645-54-62
E-mail: spopereshnyak@gmail.com

ВИМОГИ ДО ОФОРМЛЕННЯ СТАТЕЙ

1. Загальні положення

У журналі "Проблеми програмування" публікуються наукові матеріали, які раніше не публікувалися в інших виданнях.

Мова статті: українська, англійська. 16.07.2020 р. набули чинності положення Закону України «Про забезпечення функціонування української мови як державної». Відповідно до статті 22 «Державна мова у сфері науки» у наукових виданнях не повинно бути вміщено матеріалів іншими мовами, окрім державної, англійської та мов ЄС.

Обсяг статті - від 6 до 16 сторінок формату А4. Документ зберігається у форматі doc або docx.

Назва файлу включає транслітерацію прізвища автора (авторів), наприклад, “Petrenko.doc”.

Стаття надається без нумерації сторінок.

Для передачі до редакції тексту статті, ділової переписки та правки при коректурі автори користуються електронною поштою редакції: alengoro@isofts.kiev.ua, alengoro2022@gmail.com. Для надійності інформаційного обміну в умовах можливих відключень електрики – прохання надсилати матеріали на обидві електронні пошти одночасно. Телефон: +380 (96) 418 3082.

2. Оформлення файлу з текстом статті

При підготовці файлу використовуються: стиль нормальний (звичайний) або normal; шрифт Times New Roman, розмір шрифту 12 пт.; міжрядковий інтервал – 1,0; абзацний відступ -1,25 см; вирівнювання – по ширині. У тексті не допускається вирівнювання пропусками; розстановка переносів – автоматична. Формат паперу А4, розміри полів документа – 20 мм. Текст статті після зазначення авторів, назви і анотацій (двома мовами) має бути оформлений **у 2 колонки,** ширина яких – 7,86 см, а пробіл між ними – 1,27 см.

3. Послідовність розміщення та оформлення матеріалу статті

1. ***Верхній колонтитул:*** назва рубрики відповідно до переліку, прийнятому редакцією журналу *(пропонується авторами, остаточно уточняється редакцією).*

УДК *(зліва під рискою верхнього колонтитулу)**:*** індекс за універсальною десятковою класифікацією. ***DOI:*** в тому ж рядку правіше *(заповнюється редакцією)*.

Автори: ініціали та прізвища авторів, курсив *(світлий)*.

Заголовок 1 ***(назва статті)***: не містить абревіатур та строго відповідає змісту статті. Шрифт 15 пт, напівжирний, регістр верхній, вирівнювання по центру.

Анотація: 1800-2100 знаків враховуючи пробіли, не має бути абревіатур. Шрифт 10 пт, звичайний.

Ключові слова: не більше 10 слів, не містить абревіатур, подаються в називному відмінку, розділені комами. Шрифт 10 пт, звичайний.

УВАГА!

Автори, заголовок статті, анотація і ключові слова зазначаються **ДВІЧІ:** українською і англійською мовами. Спочатку мовою статті, потім іншою мовою.

2. ***Нижній колонтитул*** (тільки для першої сторінки) включає стандартну інформацію Copyright: перший рядок – прізвища авторів, рік; другий рядок – номер ISSN, назва журналу, рік, номер випуску.
3. **Заголовок 2** ***(назва розділу)***: шрифт 14 пт, напівжирний; абзац із центральним вирівнюванням, без переносів. Заголовки нижчого рівня *(пункти і т.п.)* у

самостійний абзац не виділяються і проходять першим реченням текстового абзацу, шрифт 12 пт, напівжирний.

4. **Формули** створюються в редакторі Microsoft Equation 3.0 або MathType. Формули, на які є посилання в тексті, повинні мати наскрізну нумерацію. Номер формули друкується в круглих дужках біля краю правого поля. Розмір основного шрифту редактора формул – 12 пт. Розміри символів у формулах: звичайний – 12 пт, великий індекс – 9 пт, дрібний індекс – 7 пт, великий символ – 18 пт, дрібний символ – 11 пт. Не допускається масштабування формульних об'єктів. Великі формули мають бути розбиті на декілька рядків.

Наприклад:

$$\frac{\partial T}{\partial t}+\frac{u}{a\cos\varphi}\frac{\partial T}{\partial \lambda}+\frac{v}{a}\frac{\partial T}{\partial \varphi}+w\frac{\partial T}{\partial z}=\frac{\delta}{c_V\rho}, \quad (1)$$

де λ – довгота, φ – широта, z – висота над рівнем моря, $V=(u,v,w)$, a – радіус Землі, ω – швидкість добового обертання Землі, $F_f=(F_\lambda, F_\varphi, F_z)$.

5. **Рисунки** мають бути створені вбудованим редактором Word Picture або експортовані з прикладних програм Windows у графічних форматах (bmp, pcx, gif, jpg або tif). Рисунки розташовуються по центру. Нумерація рисунків здійснюється відповідно до порядку згадування у тексті. Нумеровані підписи розміщуються під рисунком з позначенням "Рис. ", далі вказується номер рисунка і текст підпису.
6. **Таблиці** мають бути підготовлені стандартним вбудованим в Word інструментарієм "Таблиця". Таблиці нумеруються за порядком згадування. На номер таблиці мають бути посилання в тексті. Номер таблиці вказується в окремому рядку з вирівнюванням по правій стороні (наприклад: "Таблиця 1"). Назви таблиць розміщуються над таблицею з вирівнюванням по центру. Мінімальний розмір шрифту в таблицях – 11 пт.

При посиланні на формулу, рисунок, таблицю або літературне джерело, використовуйте наступні позначення відповідно: (1), (1, 2); Рис.1, Рис.1, 2; Табл.1., Табл.1, 2; [1], [1, 2].

7. **Основний текст статті** має такі необхідні елементи:

– постановка проблеми в загальному вигляді і її зв'язок з важливими науковими або практичними завданнями;

– аналіз останніх досліджень і публікацій, у яких розпочато рішення даної проблеми і на які спирається автор, виділення невирішених раніше частин загальної проблеми, яким присвячується дана стаття;

– формулювання цілей статті (постановка задачі);

– виклад основного матеріалу дослідження з повним обґрунтуванням отриманих наукових результатів;

– висновки з даного дослідження і перспективи подальших розробок у даному напрямку;

– подяка (за наявності такої).

Застосований у статті маркований список (наведений вище) має наступні параметри: маркер має відступ 1,25 см, текст для першого рядка – 1,9 см. Аналогічні відступи слід підтримувати і для нумерованого списку.

8. **Література**: єдиний нумерований список джерел згідно ДСТУ 8302:2015 від 01.07.2016 р., шрифт 11 пт, відступ: спеціальний, навислий, 0,63 см.
9. **References:** література англійською мовою подається як список використовуваних джерел згідно *Harvard Style*. Джерела з заголовками на латиниці наводяться без

перекладу. Для літератури джерел на мовах, що не використовують латинський алфавіт, необхідно забезпечити переведення назв джерел і вказати після них у дужках мову оригіналу. Прізвища та ініціали авторів, слід транслітерувати за правилами як для закордонного паспорта.

Література, що надана другою мовою не враховується при підрахунку кількості сторінок статті. У випадках, коли список джерел включає джерела тільки однією мовою, він подається один раз.

10. ***Дата надходження статті*** позначається редакцією цифрами окремим рядком після слова «Одержано:»/”Received:”.

11. ***Дата надходження внутрішньої рецензії*** позначається редакцією цифрами окремим рядком після слів «Внутрішня рецензія отримана»/”Internal review received:”.

12. ***Дата надходження зовнішньої рецензії*** позначається редакцією цифрами окремим рядком після слів «Зовнішня рецензія отримана:»/” External review received:”.

Відомості про рецензентів конкретної статті не розголошуються для підвищення об’єктивності рецензування.

13. ***Дані про авторів*****:** мають починатися рядком “*Про авторів:*”, напівжирний курсив. Далі вказуються для кожного з авторів ПІБ повністю, вчений ступінь, наукове звання, посада, обов’язково номер ORCID (сайт ORCID http://orcid.org/). Особисті телефони та електронні пошти авторів вказуються тут тільки, якщо автор хоче, щоб вони були опубліковані в журналі.

Перелік авторів подається під номерами (надстроковим шрифтом), що відповідають нумерації місць роботи, де вони працюють (надстроковим шрифтом).

14. ***Дані про місце роботи авторів*****:** починаються рядком “*Місце роботи авторів:*”, напівжирний курсив. Далі вказуються місце роботи, телефон, електронна пошта, веб-сайт.

15. Обов’язково вказати мобільний телефон і e-mail відповідального виконавця для роботи з редактором при підготовці статті до друку. Ця інформація не публікується і призначена виключно для контакту редактора з авторами.

    Для полегшення підготовки статей, що задовольняють вищенаведеним вимогам, редакція журналу розробила файл шаблону статті “shablon.dot”, який можуть використовувати автори.