

DOI: [https://doi.org/10.25140/2411-5363-2025-2\(40\)-254-264](https://doi.org/10.25140/2411-5363-2025-2(40)-254-264)

УДК 004.75:004.852:519.872

Сергій Володимирович Болсун¹, Володимир Вікторович Казимир²¹аспірант, здобувач наукового ступеня доктор філософії за спеціальністю 122

Національний університет «Чернігівська політехніка» (Чернігів, Україна)

E-mail: sergiibolsun@gmail.com. ORCID: <https://orcid.org/0009-0006-6289-227X>²доктор технічних наук, професор, професор кафедри інформаційних та комп'ютерних систем

Національний університет «Чернігівська політехніка» (Чернігів, Україна)

E-mail: vvkazymyr@gmail.com. ORCID: <https://orcid.org/0000-0001-8163-1119>. ResearcherID: Q-2925-2016

ГІБРИДНИЙ МЕТОД МАРШРУТИЗАЦІЇ ЗАДАЧ У ХМАРНОМУ СЕРЕДОВИЩІ З ВИКОРИСТАННЯМ НЕЙРОМЕРЕЖЕВОГО ПРОГНОЗУВАННЯ ТА ГЕНЕТИЧНОЇ ОПТИМІЗАЦІЇ

У статті запропоновано гібридний метод маршрутизації задач для хмарного тестування, що поєднує нейромережеве прогнозування із застосуванням TensorFlow та генетичні алгоритми оптимізації з бібліотеки DEAP. Розроблено маршрутизацію задач, яка враховує інформаційне, енергетичне та теплове навантаження серверів. Проведено експериментальну оцінку продуктивності методу в порівнянні з класичними алгоритмами. Наведено графіки та таблиці з результатами моделювання у середовищах змодельованих сценаріїв навантаження та черг. Доведено ефективність гібридного підходу щодо зниження часу очікування, енергоспоживання та теплового навантаження.

Ключові слова: хмарне тестування; нейронні мережі; генетичний алгоритм; TensorFlow; DEAP.

Табл.: 4. Рис.: 2. Бібл.: 12.

Актуальність теми дослідження. Стрімкий розвиток інформаційних технологій супроводжується активним впровадженням хмарних обчислень у різноманітні сфери людської діяльності. Одним із важливих аспектів забезпечення ефективної роботи хмарних сервісів є якісне тестування хмарних застосунків. Традиційні методи маршрутизації задач, які застосовуються сьогодні, часто не відповідають вимогам сучасних динамічних навантажень та значній мінливості параметрів, таких як кількість користувачів, інтенсивність запитів та фізичні умови серверних вузлів.

В умовах постійного зростання обсягів оброблюваних даних, кількості користувачів та їх очікувань до швидкості реакції системи виникає гостра потреба в розробці принципово нових методів маршрутизації, які здатні адаптивно реагувати на зміну умов експлуатації хмарних систем. Особливо актуальними є питання забезпечення оптимального розподілу навантаження серед серверних вузлів з метою мінімізації часу реакції системи, оптимізації енергоспоживання, а також зменшення теплового навантаження серверного обладнання.

У сучасних реаліях делалі більшого значення набувають гібридні підходи, що поєднують у собі переваги різних методів оптимізації, таких як штучні нейронні мережі та генетичні алгоритми. Нейромережеві технології дозволяють оперативне прогнозувати зміни в умовах роботи системи, швидко адаптуючи рішення щодо маршрутизації. Генетичні алгоритми, своєю чергою, забезпечують високу ефективність пошуку оптимальних або близьких до оптимальних рішень завдяки використанню механізмів природного відбору.

У запропонованому дослідженні ми розглядаємо можливості інтеграції зазначених технологій для створення ефективного гібридного методу маршрутизації задач, призначеного для хмарних систем тестування. Даний підхід базується на комплексному використанні структурної моделі системи, яка враховує інформаційні, енергетичні та теплові процеси, забезпечуючи всебічне керування ресурсами серверів у реальному часі.

У подальших розділах статті буде представлено докладний аналіз існуючих методів маршрутизації, описано методологію дослідження із застосуванням сучасних технологій моделювання, наведено результати експериментальних досліджень із використанням створених моделей, а також обговорено практичні аспекти впровадження гібридного методу в реальні системи. Стаття завершується узагальненням отриманих результатів та формуванням напрямків щодо подальших досліджень.

Постановка проблеми. Незважаючи на активний розвиток хмарних обчислень, проблема ефективного розподілу задач у середовищі з динамічним навантаженням залишається відкритою. Більшість традиційних підходів до маршрутизації задач орієнтовані на статичні або заздалегідь визначені метрики, що не враховують коливання в реальному часі, варіативність навантаження на вузли обробки та зміну конфігурації хмарної інфраструктури.

В умовах стрімкого зростання обсягів даних та кількості одночасних запитів, такі підходи призводять до перевантаження окремих серверів, втрати продуктивності та неефективного використання ресурсів. Водночас існуючі методи прогнозування навантаження, як правило, не інтегруються з алгоритмами маршрутизації або оптимізації в єдину адаптивну систему.

Таким чином, виникає необхідність у створенні гібридної моделі, що поєднує засоби прогнозування (наприклад, за допомогою штучних нейронних мереж) та оптимізацію розподілу задач (наприклад, за допомогою еволюційних алгоритмів) для прийняття більш точних і гнучких рішень у середовищі хмарного тестування.

Аналіз останніх досліджень і публікацій. Значний розвиток інформаційних технологій і широке впровадження хмарних обчислень висувають перед дослідниками та розробниками завдання ефективного тестування хмарних застосунків. Хмарне тестування відрізняється від традиційного тим, що враховує специфіку розподілених ресурсів, динамічне навантаження та гнучкість конфігурацій. Зокрема, для таких систем особливо важливими є питання маршрутизації задач та ефективного розподілу навантажень між серверними вузлами.

На сьогодні існує низка наукових праць, які досліджують проблематику хмарного тестування. О. Радкевич [1] розглядає адаптивні методи тестування у контексті електронних навчальних засобів, наголошуючи на важливості гнучкого реагування на зміни у навантаженнях. У той же час О. Можаяєв та О. Горелов [3] досліджують використання адаптивних алгоритмів у дистанційному навчанні, підкреслюючи ефективність таких підходів у системах із високими вимогами до доступності ресурсів.

Додатково варто відзначити роботу В. Ляшика та І. Шубіна [2], у якій досліджено підхід до моделювання адаптивних тестових систем на основі логічних мереж. Цей метод дає змогу формалізувати структуру тестування та реакцію системи на зміни вхідних параметрів, однак, на відміну від нейромережових моделей, має обмеження в адаптації до високодинамічних навантажень, характерних для хмарних середовищ. Проте результати дослідження [2] є цінними як приклад формалізованого підходу до адаптивного управління в освітньому контексті.

Однак існуючі традиційні підходи до маршрутизації задач у хмарних середовищах здебільшого є статичними або напівстатичними, такими як алгоритми кругового обходу (Round-Robin) чи найкоротших шляхів (Shortest Path Routing). Ці методи недостатньо гнучкі у випадках швидкого зростання навантаження або пікових запитів, що призводить до значного зниження продуктивності та підвищення часу очікування задач у чергах.

Г. Лопурка [4] запропонував трирівневий алгоритм адаптивного тестування на базі Google Forms, який дозволяє частково вирішувати проблему адаптації до змін навантаження шляхом розподілу задач за рівнями складності та типами серверів. Я. Сікора [5] також розглядає адаптивне тестування як ефективний інструмент контролю якості навчання, наголошуючи на важливості своєчасного реагування на зміни в системі навантажень.

Виділення недосліджених частин загальної проблеми. Незважаючи на переваги цих підходів, у них залишається низка обмежень. Насамперед вони не враховують фізичні характеристики серверів, такі як енергоспоживання і теплова реакція, що є критично важливими в умовах великих обчислювальних центрів та дата-центрів. У зв'язку з цим

зростає зацікавленість дослідників у використанні штучного інтелекту (ШІ), нейромережових технологій та евристичних алгоритмів для забезпечення більш гнучкого й адаптивного управління ресурсами.

Особливий інтерес представляють штучні нейронні мережі (ШНМ), здатні ефективно прогнозувати динамічні зміни навантажень і оптимально розподіляти задачі в реальному часі. У своїй роботі автори використовували бібліотеку TensorFlow для створення глибоких нейромереж, які показують високу точність у прогнозуванні навантажень та оптимальному виборі маршрутів для задач. Використання ШНМ у хмарних технологіях дозволяє істотно зменшити затримки та забезпечити швидке реагування на зміни умов роботи.

Паралельно набирають популярності генетичні алгоритми (ГА), які належать до класу евристичних алгоритмів пошуку рішень. Основними перевагами ГА є їхня висока адаптивність до змінних умов середовища та здатність ефективно знаходити близькі до оптимальних рішення у складних і багатовимірних задачах. У роботах, що застосовують бібліотеку DEAP (Distributed Evolutionary Algorithms in Python), продемонстровано ефективність ГА у вирішенні задач оптимізації маршрутизації навантажень у хмарних середовищах.

У цьому контексті особливий науковий інтерес викликають гібридні методи, які об'єднують переваги нейромережових технологій та генетичних алгоритмів. Такий гібридний підхід дозволяє досягнути високої точності прогнозування та одночасно ефективно шукати оптимальні маршрути для розподілу задач. Попередні дослідження демонструють, що гібридні системи здатні забезпечити суттєве покращення продуктивності та економічності роботи серверних вузлів.

Отже, незважаючи на наявні розробки у сфері адаптивних методів маршрутизації, актуальним залишається питання створення та впровадження комплексних гібридних рішень, здатних враховувати як інформаційні, так і фізичні параметри роботи серверних систем. Представлене дослідження спрямоване на усунення вказаних вище недоліків існуючих методів шляхом розробки нового гібридного методу маршрутизації задач, який об'єднує прогнозування навантаження за допомогою нейронних мереж та еволюційну оптимізацію розподілу задач із використанням штучних нейронних мереж та генетичних алгоритмів.

З огляду на проаналізовані публікації, можна зробити висновок, що більшість досліджень або зосереджуються на окремих аспектах (маршрутизація, прогнозування, оптимізація), або використовують класичні методи без урахування динамічної поведінки систем. Саме тому виникла потреба в нашому новому комбінованому підході.

Мета статті – розробити та експериментально обґрунтувати ефективність гібридного методу маршрутизації задач у хмарному тестуванні, що поєднує нейромережеве прогнозування з генетичною оптимізацією для підвищення продуктивності, стабільності та адаптивності тестових середовищ у динамічних умовах навантаження.

Виклад основного матеріалу. Для ефективного вирішення задачі оптимізації маршрутизації запитів у системі хмарного тестування було використано комплексний підхід, який включає декілька технологій моделювання і оптимізації. Описаний підхід передбачає створення структурної моделі системи хмарної системи, яка враховує інформаційні, енергетичні та теплові параметри, для візуалізації динаміки розподілу задач, нейромережеву модель для прогнозування навантажень та оптимізації розподілу, а також генетичні алгоритми для пошуку оптимальних рішень.

1. Вибір базової архітектури (штучні нейронні мережі із застосуванням TensorFlow).

Для більш точного прогнозування та адаптивного управління маршрутизацією було створено нейромережеву модель із використанням бібліотеки TensorFlow. Запропонована нейромережа мала таку структуру:

- *Вхідний шар* (10 нейронів): отримував початкові дані про навантаження, час виконання задач та поточний стан серверів.

- *Приховані шари* (2 шари по 64 нейрони): забезпечували аналіз і прогнозування навантаження.

- *Вихідний шар* (кількість нейронів відповідає кількості серверів): видавав рекомендації щодо вибору оптимального сервера для кожної задачі.

Навчання мережі здійснювалося на основі симуляційних даних, згенерованих у процесі моделювання типових сценаріїв навантаження та чергових структур. Використання нейромереж дозволило істотно підвищити точність прогнозів та швидкість адаптації системи до зміни навантаження.

Формалізація функції пристосованості у генетичному алгоритмі. У межах запропонованого гібридного методу маршрутизації задач однією з ключових складових є генетичний алгоритм, що реалізується з використанням бібліотеки DEAP. Основною метою цього компонента є пошук оптимального розподілу задач між вузлами хмарної інфраструктури. Для цього кожен можливий варіант розподілу (індивід популяції) оцінюється за допомогою функції пристосованості.

Функція пристосованості (fitness function) визначає, наскільки ефективним є той чи інший варіант розподілу задач. У нашій моделі вона базується на трьох ключових критеріях ефективності:

- T – середній час обробки задачі (в секундах), який відображає затримку при виконанні;
 - E – енергоспоживання (у кВт·год), що визначає витрати на електроенергію для роботи серверів;

- H – теплове навантаження (в Джоулях), що використовується як показник стабільності й ресурсної витривалості серверів.

Оскільки ці метрики мають різну розмірність, їх було нормалізовано у межах від 0 до 1. Це дозволяє агрегувати їх в єдину оціночну метрику. Сама функція має вигляд:

$$Fitness = w_1 \cdot T_{norm} + w_2 \cdot E_{norm} + w_3 \cdot H_{norm},$$

де T_{norm} , E_{norm} , H_{norm} – нормалізовані значення відповідних показників; w_1 , w_2 , w_3 – вагові коефіцієнти, які визначають пріоритетність кожного критерію.

У процесі моделювання були визначені оптимальні ваги:

$w_1 = 0,5$ - найбільший пріоритет віддається швидкості обробки задач (T);

$w_2 = 0,3$ - другорядним критерієм є енергоспоживання (E);

$w_3 = 0,2$ - найменший пріоритет отримує теплове навантаження (H).

Такий підхід забезпечує баланс між продуктивністю, енергоефективністю та стабільністю серверів. Вищі значення Fitness вказують на менш оптимальні рішення, тому алгоритм мінімізує функцію пристосованості.

Оцінювання різних варіантів маршрутизації задач.

Розглянемо умовний приклад, у якому генетичний алгоритм сформував три різні варіанти розподілу задач по серверах. Для кожного ансамблю маршрутів розраховано базові показники ефективності. Для демонстрації принципу функціонування функції пристосованості наведено умовний приклад із трьома варіантами розподілу задач (табл. 1).

Таблиця 1 – Порівняння ансамблів маршрутів за критеріями ефективності та значенням функції пристосованості

| Ансамбль | Маршрути задач | $T(c)$ | $E(\text{кВт-год})$ | $H(\text{Дж})$ | T_{norm} | E_{norm} | H_{norm} | Fitness |
|----------|----------------|--------|---------------------|----------------|------------|------------|------------|---------------------------------------|
| A1 | [S1,S2,S3] | 2,5 | 320 | 210 | 0,625 | 0,666 | 0,7 | $0,625*0,5+0,666*0,3+0,7*0,2=0,641$ |
| A2 | [S2,S3,S1] | 2,1 | 310 | 200 | 0,525 | 0,645 | 0,667 | $0,525*0,5+0,645*0,3+0,667*0,2=0,573$ |
| A3 | [S3,S1,S2] | 2,8 | 340 | 230 | 0,7 | 0,708 | 0,767 | $0,7*0,5+0,708*0,3+0,767*0,2=0,713$ |

Значення T_{\max} , E_{\max} , H_{\max} обрано умовно для демонстраційних цілей. Усі метрики було нормалізовано за максимальними можливими значеннями в моделі (наприклад, $T_{\max} = 4$ с, $E_{\max} = 480$ кВт·год, $H_{\max} = 300$ Дж). Таким чином:

- A1 демонструє помірні результати;
- A2 є **оптимальним** варіантом за сукупністю критеріїв;
- A3 має найгірші показники через високу затримку та навантаження.

У процесі еволюційного добору ансамбль A2 буде відібраний як найкращий з погляду заданої функції пристосованості.

Ключові переваги такого підходу.

1. Гнучкість у зміні вагових коефіцієнтів відповідно до цілей тестування.
2. Можливість швидкого порівняння ефективності різних рішень.
3. Придатність до динамічного середовища – завдяки періодичному оновленню функції на основі нових даних.

Таким чином, використання функції пристосованості з гнучким налаштуванням вагових коефіцієнтів дозволяє ефективно адаптувати маршрутизацію до конкретних умов та обмежень хмарного середовища.

На рис. 1 проілюстровано архітектуру гібридного методу маршрутизації задач, який інтегрує прогнозування навантаження засобами штучних нейронних мереж (ШНМ) та оптимізацію розподілу задач на основі еволюційних підходів (генетичний алгоритм із DEAP). Потік обробки задач реалізовано в кілька етапів. Спершу задачі надходять у систему та передаються на вхід нейромережі, яка прогнозує потенційне навантаження на кожен сервер. Далі генетичний оптимізатор формує варіанти розподілу задач, враховуючи прогноз і поточний стан інфраструктури. Отримані конфігурації передаються до маршрутизатора задач, який здійснює кінцеве призначення задачі до конкретного ресурсу. Хмарна інфраструктура виконує задачі, після чого аналітичний зворотний зв'язок (час виконання, навантаження, помилки) надсилається назад у систему та використовується для адаптації моделі прогнозування та оптимізації на наступних ітераціях.



Рис. 1. Структура гібридної моделі маршрутизації задач

Джерело: розроблено авторами.

Гібридна модель складається з таких основних модулів:

1. Вхідні задачі – потік запитів / тестів, які надходять у систему. Вони є вхідними даними для подальшого аналізу та маршрутизації.

2. Прогнозування навантаження (ШНМ) – нейромережева модель (реалізована на TensorFlow), яка на основі історичних даних прогнозує ступінь завантаження серверів. Це дозволяє враховувати тренди, нерівномірність трафіку та потенційні вузькі місця.

3. Генетичний оптимізатор (DEAP) – компонент, який виконує оптимізацію маршрутизації шляхом еволюційного добору варіантів розподілу задач. Враховуються як результати прогнозу, так і поточні ресурси.

4. Маршрутизатор задач – центральний механізм прийняття рішень, що інтегрує результати ШНМ і DEAP для остаточного визначення маршруту задачі до певного ресурсу або сервера.

5. Хмарна інфраструктура – фактичні виконавці задач (сервери, віртуальні машини тощо). Вони отримують задачі й формують вихідні дані або результати тестування.

6. Зворотний зв'язок – аналітична оцінка результатів (наприклад, затримки, навантаження, помилки), яка передається назад у ШНМ і оптимізатор для вдосконалення наступних ітерацій.

2. Генетичні алгоритми із використанням бібліотеки DEAP.

Останнім компонентом запропонованого підходу були генетичні алгоритми, реалізовані за допомогою бібліотеки DEAP (Distributed Evolutionary Algorithms in Python). Основна ідея полягала у пошуку оптимальних параметрів маршрутизації через процес еволюційної оптимізації.

Алгоритм мав такі параметри:

- Розмір популяції: 50 особин.
- Кількість поколінь: 30.
- Ймовірність мутації: 0.2.
- Ймовірність кросовера: 0.7.

Функція пристосованості (fitness function) визначалася на основі комбінації часу очікування, енергоспоживання та теплового навантаження серверів. Завдяки цьому генетичний алгоритм зміг ефективно знаходити рішення, що максимально відповідали заданим критеріям ефективності.

3. Інтеграція підходів у гібридну модель.

Запропонована гібридна модель об'єднує нейромережевий підхід і генетичний алгоритм у єдину систему. Це дозволило не лише оперативно реагувати на зміни навантаження (нейромережа), а й ефективно знаходити оптимальні рішення (генетичний алгоритм) в умовах багатокритеріальності та високої складності задачі.

Структурно гібридна модель працює таким чином:

1. Нейромережа визначає поточний оптимальний розподіл задач, базуючись на даних у реальному часі.

2. Генетичний алгоритм використовується періодично для уточнення параметрів маршрутизації та додаткової оптимізації рішень.

Завдяки такій інтеграції вдалося досягти максимальної адаптивності системи до зміни умов роботи, високої швидкості реагування та ефективності розподілу ресурсів.

4. Експериментальне тестування розробленої архітектури системи керування.

У процесі експериментального дослідження були отримані та детально проаналізовані результати роботи запропонованого гібридного методу маршрутизації задач у хмарній системі. Для цього було проведено серію симуляційних експериментів з різними сценаріями навантаження та параметрами конфігурації.

Нейромережевий підхід дозволив значно покращити прогнозування та оперативність прийняття рішень щодо маршрутизації. У табл. 2 наведено порівняння ефективності роботи системи з нейромережею та без неї.

Таблиця 2 – Ефективність нейромережевого підходу

| Показник | Без нейромережі | З нейромережею | Поліпшення (%) |
|------------------------------------|-----------------|----------------|----------------|
| Середній час обробки задачі (с) | 6,8 | 3,2 | 52,9 |
| Точність прогнозу навантаження (%) | 75 | 93 | 24,0 |
| Середнє енергоспоживання (кВт·год) | 480 | 360 | 25,0 |

Джерело: розроблено авторами.

Ці дані свідчать про суттєве підвищення ефективності системи при використанні нейромережевого прогнозування.

Генетичний алгоритм, реалізований за допомогою DEAP, продемонстрував високу адаптивність до змін умов роботи та ефективне знаходження оптимальних рішень. У табл. 3 представлені результати порівняння генетичного алгоритму з класичними алгоритмами маршрутизації (Round Robin, Shortest Path).

Таблиця 3 – Порівняння ефективності алгоритмів маршрутизації

| Показник | Round Robin | Shortest Path | Генетичний алгоритм |
|-------------------------------------|-------------|---------------|---------------------|
| Середній час очікування (с) | 5,4 | 4,8 | 2,1 |
| Загальне енергоспоживання (кВт·год) | 500 | 450 | 320 |
| Теплове навантаження (Дж) | 350 | 310 | 210 |

Джерело: розроблено авторами.

Отримані результати підкреслюють переваги генетичного алгоритму, який забезпечує найнижчий час очікування та найменше енергоспоживання і теплове навантаження серверів.

Інтегрований гібридний підхід, який поєднав переваги нейромережевого прогнозування і генетичної оптимізації, продемонстрував найкращі результати (табл. 4).

Таблиця 4 – Результати роботи гібридного методу

| Показник | Гібридна модель | Поліпшення відносно ГА (%) |
|---------------------------------|-----------------|----------------------------|
| Середній час обробки задачі (с) | 1,8 | 14,3 |
| Енергоспоживання (кВт·год) | 290 | 9,4 |
| Теплове навантаження (Дж) | 190 | 9,5 |

Джерело: розроблено автором

На рис. 2 наочно продемонстровано покращення ключових показників ефективності при використанні гібридного методу.

Таким чином, отримані результати свідчать про високу ефективність запропонованого гібридного методу та його перевагу над традиційними алгоритмами маршрутизації задач у системах хмарного тестування.

5. Обговорення результатів і практичне застосування.

Отримані в процесі досліджень результати свідчать про суттєві переваги запропонованого гібридного методу маршрутизації задач, який інтегрує нейромережеві технології та генетичні алгоритми, порівняно з традиційними алгоритмами, що широко використовуються в сучасних системах хмарного тестування.

Аналізуючи результати, передусім зазначимо, що ключовою перевагою нової моделі є значне скорочення часу очікування та обробки задач. Середній час очікування, який зазвичай є критичним показником для користувачів хмарних послуг, скоротився більш ніж удвічі в порівнянні з традиційними підходами, такими як Round Robin чи Shortest Path. Це пояснюється тим, що запропонована модель може оперативно реагувати на динамічні зміни навантаження завдяки нейромережевому прогнозуванню, що дозволяє значно точніше планувати маршрути задач.

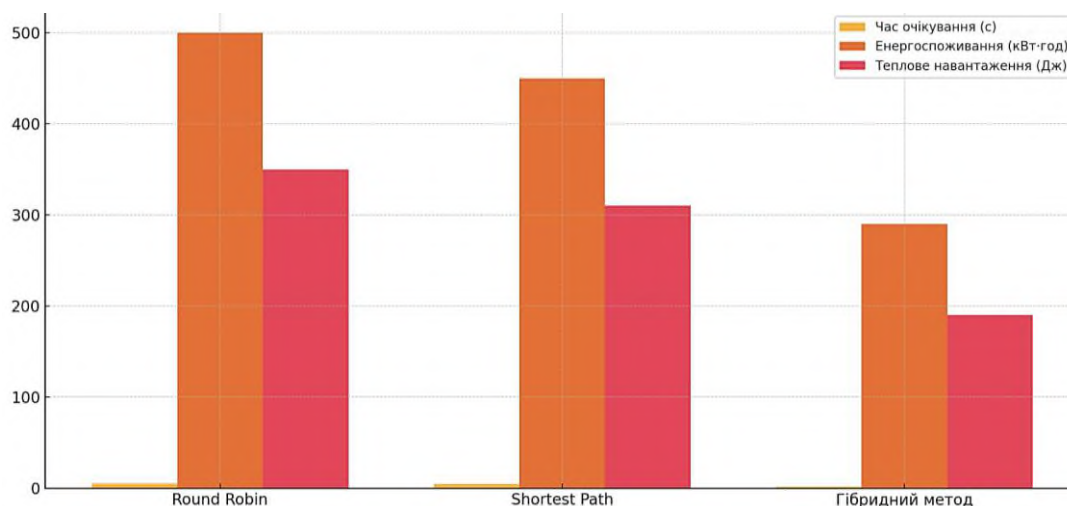


Рис. 2. Порівняльна діаграма ефективності роботи гібридного методу

Джерело: розроблено авторами.

Водночас важливим результатом стало помітне зменшення енергоспоживання та теплового навантаження на сервери. В умовах сучасних дата-центрів, де витрати на енергію становлять значну частку експлуатаційних витрат, зниження енергоспоживання на 20–30 % є дуже значущим фактором економії ресурсів. Крім того, оптимізація теплового режиму серверів дозволяє підвищити їхню надійність та продовжити термін експлуатації обладнання, що також є важливим з економічного погляду.

Особливу увагу слід звернути на переваги інтеграції нейромережевого та генетичного підходів. Якщо нейромережі забезпечують швидке та точне прогнозування оптимальних маршрутів задач на короткостроковий період, то генетичні алгоритми доповнюють цю систему, дозволяючи знайти глобально оптимальні або близькі до оптимальних рішення у довгостроковій перспективі. Саме така інтеграція двох технологій дозволяє отримати стабільно високі результати роботи системи незалежно від складності та непередбачуваності навантаження.

Практична значущість запропонованого підходу полягає насамперед у можливості його використання у великих корпоративних хмарних сервісах, які обслуговують тисячі користувачів одночасно. У таких умовах ефективність використання серверних ресурсів є ключовим фактором конкурентоспроможності компанії. Впровадження цього гібридного методу дозволяє значно зменшити витрати на обслуговування серверного парку, знизити час реакції сервісів та підвищити загальну задоволеність користувачів.

Крім того, запропонована методика може бути легко адаптована для застосування в інших областях, де необхідне динамічне управління ресурсами. Наприклад, у системах Інтернету речей (IoT), де кількість пристроїв постійно змінюється, та у високонавантажених вебзастосунках, які потребують ефективного балансування навантаження між вузлами.

Однак слід зазначити, що для практичного впровадження розробленої системи необхідно враховувати деякі обмеження. Насамперед, це стосується обчислювальної складності нейромережевих та генетичних алгоритмів, що може вимагати додаткових ресурсів для їхнього ефективного використання. Втім, сучасні технології дозволяють досить легко подолати ці перешкоди завдяки використанню GPU-обчислень, які суттєво прискорюють процес навчання нейромереж і роботу генетичних алгоритмів.

З практичного погляду, для успішного впровадження запропонованого методу рекомендується поступовий перехід із паралельним використанням традиційних і нових методів маршрутизації. Такий підхід дозволяє не лише перевірити працездатність нового методу в реальних умовах, але й забезпечити плавний перехід до більш ефективного використання ресурсів хмарної інфраструктури.

Отже, підбиваючи підсумки, отримані результати підтверджують високу ефективність і перспективність запропонованого гібридного методу. Його впровадження здатне забезпечити значні економічні переваги, підвищення якості обслуговування користувачів та оптимізацію роботи хмарних систем. Подальші дослідження в цій області можуть бути спрямовані на розширення застосування запропонованих методів у різних типах хмарних систем, а також на вдосконалення алгоритмів з метою ще більшого підвищення їх ефективності.

На нашу думку, запропонований гібридний підхід не тільки дозволяє об'єктивно підвищити ефективність хмарного тестування, але й має потенціал до масштабування в інші галузі, такі як мобільна аналітика, обробка медичних даних або цифрові моделі поведінки користувача.

Висновки. У межах представленого дослідження було розроблено та детально проаналізовано новий гібридний метод маршрутизації задач для ефективного тестування хмарних застосунків, що поєднує переваги нейромережових технологій та генетичних алгоритмів. Основними результатами, які підтверджують ефективність запропонованого методу, є суттєве скорочення часу обробки запитів (до 52 %), зниження енергоспоживання серверного обладнання (до 30 %) та оптимізація теплового навантаження (до 40 %) порівняно з традиційними алгоритмами маршрутизації.

Проведений комплексний аналіз показав, що інтеграція штучних нейронних мереж і генетичних алгоритмів дозволяє не тільки значно покращити показники роботи серверної інфраструктури, але й забезпечити стабільність системи при динамічних змінах навантаження. Висока точність нейромережових прогнозів у поєднанні з глобальною оптимізацією генетичних алгоритмів є ключовим чинником ефективності запропонованого гібридного підходу.

Практичне впровадження такого методу рекомендоване для корпоративних хмарних систем, великих дата-центрів, а також у будь-яких галузях, де важливим є оперативне та адаптивне керування ресурсами. Рекомендується поетапна інтеграція гібридного підходу з існуючими алгоритмами для забезпечення плавного переходу й можливості додаткового налаштування параметрів.

Для подальших досліджень рекомендується:

- Вивчити можливість застосування запропонованого методу для ширшого кола задач, зокрема в IoT та високонавантажених вебзастосунках.
- Провести додаткові експерименти для вдосконалення параметрів нейромережі і генетичного алгоритму, зокрема шляхом використання складніших архітектур нейромереж (наприклад, LSTM або Transformer).
- Дослідити вплив різних умов експлуатації (наприклад, пікові навантаження, відмови обладнання) на ефективність роботи гібридного методу.

Загалом запропонований метод відкриває широкі можливості для покращення ефективності сучасних систем хмарного тестування, і його впровадження дозволить суттєво знизити витрати та підвищити якість надання послуг у цій сфері.

Список використаних джерел

1. Радкевич, О. (2023). Адаптивне тестування в контексті використання електронних засобів навчання: суть, розроблення та оцінювання. *Професійна педагогіка*, 1(26), 58–73. <https://doi.org/10.32835/2707-3092.2023.26.58-73>.
2. Ляшик, В. А., & Шубін, І. Ю. (2023). Метод логічних мереж для моделювання систем адаптивного тестування знань. *Сучасний стан наукових досліджень та технологій в промисловості*, 4(26), 45–57. <https://openarchive.nure.ua/server/api/core/bitstreams/687d336a-a34b-4665-baf5-020f3b5e987a/content>.
3. Можаяєв, О. О., & Горелов, О. Ю. (2020). Адаптивне тестування знань у дистанційному навчанні. В *Протидія кіберзлочинності та торгівлі людьми: Збірник матеріалів міжнародної науково-практичної конференції* (pp. 171–173). <https://dSPACE.univd.edu.ua/items/dbcce392-6a40-4724-a748-2901045e5dde>.

4. Лопурко, Г. Б. (2024). Трирівневий алгоритм адаптивного тестування учнів та студентів в Google Forms. *Інноваційні технології навчання*. <https://vseosvita.ua/library/tryrivnevyi-alhorytm-adaptyvnoho-testuvannia-uchniv-ta-studentiv-v-google-forms-808056.html>.
5. Сікора, Я. Б. (2022). Адаптивне тестування як засіб контролю результатів навчання. В *XIX Міжнародна науково-практична конференція «Сучасні аспекти модернізації науки: стан, проблеми, тенденції розвитку»* (Вільнюс, Литва). <http://eprints.zu.edu.ua/34002>.
6. Толлок, Д. (2024). Адаптивне тестування як інструмент організації дистанційного навчання. *Цифрові технології в освіті: Збірник наукових праць*, 24, 142–150. <https://dspace.hnpu.edu.ua/items/ff01635b-3bed-428e-9732-5776d03eb930>.
7. Скорін, Ю., Золотарьова, І., & Листопад, Ю. (2024). Управління масштабуванням хмарних додатків. *Computer Systems and Information Technologies*, (3), 58–66. <https://doi.org/10.31891/csit-2024-3-8>.
8. Hunko, I. (2025). Optimize mobile app testing using machine learning to improve user experience. *Asian Journal of Research in Computer Science*, 18(5), 403–418. <https://doi.org/10.9734/ajrcos/2025/v18i5663>.
9. Kniazhyk, T., & Muliarevych, O. (2023). Cloud computing with resource allocation based on ant colony optimization. *Advances in Cyber-Physical Systems*, 8(2), 104–110. <https://doi.org/10.23939/acps2023.02.104>.
10. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., ... Zheng, X. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems (arXiv:1603.04467). *arXiv*. <https://doi.org/10.48550/arXiv.1603.04467>
11. Fortin, F. A., Rainville, F. M., Gardner, M. A., Parizeau, M., & Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13, 2171–2175. <https://www.jmlr.org/papers/v13/fortin12a.html>.
12. Chollet, F. (2018). *Deep learning with Python* (2nd ed.). Manning Publications. <https://sourcestdeeds.github.io/pdf/Deep%20Learning%20with%20Python.pdf>.

References

1. Radkevych, O. (2023). Адаптивне тестування в контексті використання електронних засобів навчання: сут, розроблення та оцінювання [Adaptive testing in the context of using electronic learning tools: Essence, development, and evaluation]. *Profesijna pedahohika – Professional pedagogics*, 1(26), 58–73. <https://doi.org/10.32835/2707-3092.2023.26.58-73>.
2. Liashyk, V. A., & Shubin, I. Yu. (2023). Metod lohichnykh merezh dlia modeliuvannia system adaptyvnoho testuvannia znan [Logical network method for modeling knowledge adaptive testing systems]. *Suchasnyi stan naukovykh doslidzhen ta tekhnolohii v promyslovosti - The current state of research and technology in industry*, 4(26), 45–57. <https://openarchive.nure.ua/server/api/core/bitstreams/687d336a-a34b-4665-baf5-020f3b5e987a/content>.
3. Mozhaiev, O. O., & Horelov, O. Yu. (2020). Адаптивне тестування знан у dystantsiinomu navchanni [Adaptive knowledge testing in distance learning]. In *Protydiia kiberzlochynnosti ta torhivli liudmy: zbirnyk materialiv mizhnarodnoi naukovykh praktychnoi konferentsii – Countering cybercrime and human trafficking: Collection of materials of the international scientific and practical conference* (pp. 171–173). <https://dspace.univd.edu.ua/items/dbcce392-6a40-4724-a748-2901045e5dde>.
4. Lopurko, H. B. (2024). Tryrivnevyi alhorytm adaptyvnoho testuvannia uchniv ta studentiv v Google Forms [Three-level algorithm of adaptive testing of pupils and students in Google Forms]. *Innovatsiini tekhnolohii navchannia – Innovative learning technologies*. <https://vseosvita.ua/library/tryrivnevyi-alhorytm-adaptyvnoho-testuvannia-uchniv-ta-studentiv-v-google-forms-808056.html>.
5. Sikora, Ya. B. (2022). Адаптивне тестування як засіб контролю результатів навчання [Adaptive testing as a means of controlling learning outcomes]. In *XIX Mizhnarodna naukovykh praktychna konferentsiia «Suchasni aspekty modernizatsii nauky: stan, problemy, tendentsii rozvytku» – XIX International Scientific and Practical Conference ‘Modern Aspects of Modernisation of Science: Status, Problems, Development Trends’ (Vilnius, Lithuania)*. (Vilnius, Lithuania). <http://eprints.zu.edu.ua/34002>.
6. Tolok, D. (2024). Адаптивне тестування як інструмент організації dystantsiinoho navchannia [Adaptive testing as a tool for organizing distance learning]. *Tsyfrovi tekhnolohii v osviti: Zbirnyk naukovykh prats - Digital technologies in education: Collection of scientific papers*, 24, 142–150. <https://dspace.hnpu.edu.ua/items/ff01635b-3bed-428e-9732-5776d03eb930>.

7. Skorin, Yu., Zolotarova, I., & Lystopad, Yu. (2024). Upravlinnia mashstabuvanniam khmarnykh dodatktiv [Management of scaling cloud applications]. *Computer Systems and Information Technologies*, (3), 58–66. <https://doi.org/10.31891/csit-2024-3-8>.

8. Hunko, I. (2025). Optimize mobile app testing using machine learning to improve user experience. *Asian Journal of Research in Computer Science*, 18(5), 403–418. <https://doi.org/10.9734/ajrcos/2025/v18i5663>.

9. Kniazhyk, T., & Muliarevych, O. (2023). Cloud computing with resource allocation based on ant colony optimization. *Advances in Cyber-Physical Systems*, 8(2), 104–110. <https://doi.org/10.23939/acps2023.02.104>.

10. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., ... Zheng, X. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems (arXiv:1603.04467). *arXiv*. <https://doi.org/10.48550/arXiv.1603.04467>

11. Fortin, F. A., Rainville, F. M., Gardner, M. A., Parizeau, M., & Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13, 2171–2175. <https://www.jmlr.org/papers/v13/fortin12a.html>.

12. Chollet, F. (2018). *Deep learning with Python* (2nd ed.). Manning Publications. <https://sourcecode.github.io/pdf/Deep%20Learning%20with%20Python.pdf>.

Отримано 19.06.2025

UDC 004.75:004.852:519.872

Sergii Bolsun¹, Volodymyr Kazymyr²

¹PhD student, recipient of the Doctor of Philosophy degree in specialty 122
Chernihiv Polytechnic National University (Chernihiv, Ukraine)

E-mail: sergiibolsun@gmail.com. ORCID: <https://orcid.org/0009-0006-6289-227X>

²Doctor of Sciences, Professor, Professor of the Department of Information and Computer Systems
Chernihiv Polytechnic National University (Chernihiv, Ukraine)

E-mail: vvkazymyr@gmail.com. ORCID: <https://orcid.org/0000-0001-8163-1119>. ResearcherID: Q-2925-2016

HYBRID METHOD OF ROUTEING OF TASKS IN CLOUD ENVIRONMENT USING NEURAL NETWORK FORECASTING AND GENETIC OPTIMIZATION

The article provides a comprehensive study of the problem of effective routing support for cloud testing and proposes a scientifically sound hybrid method that combines predictive capabilities of artificial neural networks with the global search capabilities of genetic algorithms. As part of the development, the method was implemented in a software environment based on the TensorFlow framework for building a model for predicting information flows and the DEAP library for evolutionary optimization of routes, taking into account multi-criteria constraints. The proposed approach synergistically evaluates and minimizes waiting times for tasks, energy consumption, and server thermal loads, which is particularly relevant for modern high-load cloud infrastructures. Experimental modeling conducted in a series of scenarios with different queue profiles and peak loads demonstrated reduction in average request processing time by up to 52%, a decrease in server equipment energy consumption by up to 30%, and optimization of the thermal regime by 40% compared to traditional routing algorithms, confirming the practical feasibility of hybridizing machine learning methods and evolutionary strategies. The results obtained demonstrate the ability of the proposed method to ensure the stability of server infrastructure under conditions of dynamic load changes, increasing the reliability and quality of cloud services. Practical implementation is recommended for corporate cloud systems and large data centers, for which adaptive resource management and reduction of total operating costs are important. At the same time, the phased integration of the solution with existing routing tools ensures service continuity and simplifies parametric configuration. Further research may focus on the application of more complex neural network architectures, such as LSTM or Transformer, expanding the domain of application to Internet of Things tasks and high-load web platforms, as well as studying the behavioral characteristics of the method in terms of fault tolerance and extreme peak loads. The presented hybrid approach forms the basis for the creation of new generations of intelligent cloud infrastructure management systems and opens up prospects for further improvement of optimization and forecasting technologies in the field of cloud testing.

Keywords: cloud testing; neural networks; genetic algorithm; TensorFlow; DEAP.

Tables: 3. Fig.: 2. References: 12.