

УДК 004.415.538

DOI <https://doi.org/10.32689/maup.it.2025.3.11>

Кирило КОХАН

аспірант кафедри комп'ютерних інформаційних систем і технологій,
Національний університет біоресурсів і природокористування України,
kokhan.kyrylo@gmail.com

ORCID: 0009-0002-8878-8527

Олексій ТКАЧЕНКО

кандидат технічних наук, доцент кафедри теорії та технології програмування,
Київський національний університет імені Тараса Шевченка,
otkachenko@knu.ua

ORCID: 0000-0002-9514-516X

ОГЛЯД ТА ПРОПОЗИЦІЯ ОПТИМІЗАЦІЇ ОПТИМАЛЬНИХ КОНФІГУРАЦІЙ ДЛЯ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ БАГАТОКОМПОНЕНТНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

Анотація. У статті проаналізовано сучасні підходи до вибору оптимальних конфігурацій для автоматизованого тестування багатокomпонентних інформаційних систем (ІС), що є основою функціональності цифрових платформ. Визначено ключові поняття: конфігурація як комбінація параметрів (версії програмного забезпечення, бази даних, браузеру), оптимальна конфігурація як мінімальний набір комбінацій для покриття критичних сценаріїв за умов обмеження ресурсів, багатокomпонентна ІС як сукупність взаємопов'язаних компонентів (фронтенд, бекенд, API, бази даних).

Метою дослідження є оцінка сучасних методів та створення інтегрованого підходу, що поєднує комбінаторні методи, генетичні алгоритми та CI/CD для автоматизації вибору конфігурацій у реальному часі.

Методологія дослідження включає систематичний огляд літератури за останні 5 років, порівняльний аналіз із використанням вагових коефіцієнтів (кількість тестів, покриття сценаріїв, адаптивність, інтеграція з CI/CD, ресурси), математичне моделювання та апробацію на прикладі хмарних платформ і систем електронної комерції.

Наукова новизна полягає в розробці інтегрованого підходу, який скорочує кількість тестів до 5–10% від повного набору (наприклад, із 243 до 12–24 конфігурацій для системи з 5 параметрами), забезпечуючи при цьому 90–95% покриття критичних сценаріїв та високу адаптивність до змін компонентів. Унікальність підходу – інтеграція з CI/CD-процесами та використання вагового аналізу для вибору оптимальних конфігурацій.

Висновки. Запропонований підхід дозволяє оптимізувати тестування в умовах складних ІС, поєднуючи точність комбінаторних методів, ефективність генетичних алгоритмів та автоматизацію CI/CD. Перспективи подальших досліджень – використання ШІ для прогнозування дефектів і автоматичний аналіз результатів тестування.

Ключові слова: автоматизоване тестування, багатокomпонентні інформаційні системи, оптимальні конфігурації, Pairwise Testing, генетичні алгоритми, CI/CD.

Kyrylo KOKHAN, Oleksii TKACHENKO. INFORMATION TECHNOLOGY FOR OPTIMAL CONFIGURATION SELECTION IN AUTOMATED TESTING OF MULTICOMPONENT INFORMATION SYSTEMS: REVIEW AND PROPOSAL

Abstract. The article analyzes modern approaches to selecting optimal configurations for automated testing of multi-component information systems (IS), which are the basis of the functionality of digital platforms. Key concepts are defined: configuration as a combination of parameters (software versions, databases, browsers), optimal configuration as a minimum set of combinations to cover critical scenarios under resource constraints, multi-component IS as a set of interconnected components (frontend, backend, API, databases).

The objective of the study is to evaluate current methods and create an integrated approach that combines combinatorial methods, genetic algorithms, and CI/CD to automate configuration selection in real time.

Methodology includes a systematic review of the literature over the last 5 years, comparative analysis using weighting factors (number of tests, scenario coverage, adaptability, integration with CI/CD, resources), mathematical modeling, and testing on the example of cloud platforms and e-commerce systems.

The scientific novelty lies in developing an integrated approach that reduces the number of tests to 5–10% of the full set (for example, from 243 to 12–24 configurations for a system with 5 parameters), while ensuring 90–95% coverage of critical scenarios and high adaptability to component changes. The uniqueness of the approach is integration with CI/CD processes and the use of weight analysis to select optimal configurations.

Conclusions. The proposed approach allows optimizing testing in complex IS environments, combining the accuracy of combinatorial methods, the efficiency of genetic algorithms, and CI/CD automation. Prospects for further research include the use of AI for defect prediction and automatic analysis of test results.

Key words: automated testing, multi-component information systems, optimal configurations, Pairwise Testing, genetic algorithms, CI/CD.

© К. Кохан, О. Ткаченко, 2025

Стаття поширюється на умовах ліцензії CC BY 4.0

Постановка проблеми. Сучасні багатокомпонентні інформаційні системи (ІС), такі як хмарні платформи, системи електронної комерції чи мікросервісні архітектури, характеризуються високою складністю через велику кількість взаємопов'язаних компонентів (фронтенд, бекенд, бази даних, API), кожен із яких може мати різні версії, працювати в різних середовищах (локальне, production) і залежати від зовнішніх параметрів (браузери, бази даних). Це призводить до стрімкого зростання кількості тестових конфігурацій, що робить їх повне тестування неможливим через обмеження часу та ресурсів [6]. Наприклад, система з 5 параметрами по 3 значення кожен має $3^5 = 243$ конфігурації, що ускладнює забезпечення якості програмного забезпечення без автоматизації. Ряд сучасних інструментів, зокрема заснованих на методах штучного інтелекту, не завжди допомагає вирішити цю проблему [12]. Необхідність створення ефективних методів вибору оптимальних конфігурацій для автоматизованого тестування є актуальним завданням, яке потребує поєднання високого покриття критичних сценаріїв, мінімальної кількості тестів і адаптивності до змін системи [1].

Аналіз останніх досліджень і публікацій. Проблеми тестування багатокомпонентних ІС активно досліджуються в науковій літературі. Ручний вибір конфігурацій є поширеним у невеликих проєктах, але його ефективність знижується зі зростанням складності ІС через суб'єктивність і трудомісткість. Комбінаторні методи, такі як Pairwise Testing, скорочують кількість тестів шляхом покриття всіх пар значень параметрів, що дозволяє виявити до 85–90% дефектів, викликаних взаємодією двох параметрів [6; 10]. Orthogonal Arrays забезпечують збалансоване покриття, але їх генерація складніша [9]. Генетичні алгоритми (ГА) застосовуються для оптимізації вибору конфігурацій шляхом ітеративного пошуку, однак вони потребують налаштування параметрів і мають обмежену інтеграцію з CI/CD [13]. Методи на основі машинного навчання (ML) прогнозують проблемні конфігурації, але потребують значних обсягів даних і обчислювальних ресурсів [11; 4; 12]. У низці досліджень пропонують поєднувати комбінаторні методи з ГА чи ML, однак такі підходи рідко враховують повну автоматизацію в CI / CD-процесах [1; 3; 2]. Питання інтеграції комбінаторних методів із ГА та CI/CD залишається недостатньо дослідженим [7].

Метою даної статті є оцінити сучасні підходи щодо вибору оптимальних конфігурацій для автоматизованого тестування багатокомпонентних інформаційних систем та запропонувати інтегрований підхід, спрямований на оптимізацію параметрів автоматизованого тестування.

Виклад основного матеріалу. Матеріали та методи дослідження. Дослідження базується на аналізі наукової літератури з комбінаторного тестування, генетичних алгоритмів та методів ML. Використано комбінацію комбінаторних методів (Pairwise Testing, Orthogonal Arrays), генетичних алгоритмів та інтеграцію з CI/CD-пайплайнами. Для порівняльного аналізу запропоновано набір вагових коефіцієнтів, визначених на основі огляду наукових робіт [1–13], досвіду автора та експертів у галузі, що дозволило коректно оцінити практичну значущість кожного параметра. Слід зазначити, що вагові коефіцієнти можуть бути змінені залежно від конкретних завдань тестування. Наприклад, для фінансових систем пріоритетним є покриття критичних сценаріїв, а для високонавантажених сервісів – ефективність використання ресурсів. Матеріалами є технічні специфікації, UML-діаграми, конфігураційні файли (JSON, YAML) та вимоги до продуктивності реальних багатокомпонентних ІС, таких як хмарні платформи та системи електронної комерції. Методи включають математичне моделювання простору конфігурацій, оптимізацію за інтегральним критерієм та статистичну обробку результатів за допомогою вагових коефіцієнтів.

У якості матеріалів використано:

- технічні специфікації ІС, UML-діаграми та JSON/YAML-файли конфігурацій;
- реальні приклади CI/CD-процесів (GitLab CI, Jenkins) [3];
- інструменти автоматизації тестування (Selenium, Postman, JMeter) [12];
- наукові роботи, що обґрунтовують ефективність комбінаторних підходів [6; 9; 10] та еволюційних алгоритмів [13].

Методи дослідження включають:

- Pairwise Testing та Orthogonal Arrays, що дозволяють зменшити кількість тестів у 8–12 разів [6; 9; 8];
- генетичні алгоритми (GA), які використовуються для оптимізації вибору конфігурацій [13];
- метод вагових коефіцієнтів, що забезпечує багатокритеріальне оцінювання ефективності підходів [1; 4; 12].

Запропонований підхід побудовано у вигляді модульної схеми (рис. 1). Він передбачає п'ять етапів:

1. *Модуль аналізу системи.* Здійснює формалізований збір інформації про компоненти ІС, їх функціональні залежності та критичні сценарії. Результатом є структурована модель предметної області.
2. *Модуль моделювання конфігурацій.* Формує простір усіх можливих варіантів та зменшує його за допомогою Pairwise Testing і Orthogonal Arrays [6, 9] (наприклад, з 243 до 20–30 комбінацій).



Рис. 1. Структура запропонованого підходу

3. *Модуль оптимізації*. Виконує відбір підмножини конфігурацій на основі інтегральної оцінки, яка враховує покриття, час і ресурси. Для оптимізації застосовуються як класичні методи (симплекс-метод), так і евристики (генетичні алгоритми) [13].

4. *Модуль тестування*. Інтегрується з інструментами (Selenium, Postman, JMeter) та забезпечує виконання UI-, API- і навантажувальних тестів [4; 11].

5. *Модуль аналізу результатів*. Збирає інформацію про виявлені дефекти й метрики покриття. Отримані дані використовуються для уточнення вагових коефіцієнтів і повторного циклу оптимізації [12; 3].

Підхід комбінує переваги комбінаторних методів і генетичних алгоритмів [5]: Pairwise Testing забезпечує високу ефективність із покриттям 85–90% дефектів, генетичні алгоритми оптимізують вибір конфігурацій із урахуванням пріоритетів критичних сценаріїв, а відсутність залежності від великих даних чи складних обчислень відрізняє його від ШІ-підходів [4; 11].

Порівняльний аналіз підходів. Об'єктивне порівняння методів проведено за допомогою інтегральної оцінки:

$$I(j) = \sum_{i=1}^n W_i * P_{ij}$$

де W_i – ваговий коефіцієнт параметра, P_{ij} – нормалізоване значення параметра для методу j , n – кількість параметрів.

Для запропонованого підходу інтегральна оцінка становить:

$$I = 0.20 \cdot 0.95 + 0.25 \cdot 0.95 + \dots + 0.10 \cdot 1.00 + 0.05 \cdot 0.70 = 0.9075$$

Отримане значення свідчить про перевагу запропонованого методу над класичними підходами. У (табл. 1) наведено порівняльне оцінювання підходів у тестуванні на основі вагових коефіцієнтів.

Обговорення результатів. Аналіз таблиці показує:

- ручний підхід має низьку ефективність (0.395) через надлишкову кількість тестів і відсутність автоматизації [9];
- Pairwise Testing забезпечує високе покриття з мінімальною кількістю тестів (0.8025), що підтверджено працями Куна та колег [6; 10];
- генетичні алгоритми покращують адаптивність, але вимагають значних обчислювальних ресурсів [13; 1];
- ML-методи демонструють високе покриття (0.95), однак залежать від великих обсягів історичних даних [11; 4; 12];
- запропонований підхід отримав найвищу інтегральну оцінку (0.9075), поєднавши переваги комбінаторних і еволюційних методів із простотою CI/CD-інтеграції [7; 3].

Таблиця 1

Порівняльне оцінювання підходів у тестуванні на основі вагових коефіцієнтів

Параметр	W(i)	Ручний вибір	Pairwise Testing	Генетичні алгоритми	ШІ (ML)	Запропонований підхід
Кількість тестів	0.20	0.10	0.95	0.70	0.70	0.95
Покриття критичних сценаріїв	0.25	0.50	0.85	0.80	0.95	0.95
Адаптивність до змін	0.15	0.10	0.10	0.60	0.60	0.95
Використання історичних даних	0.10	1.00	1.00	0.70	0.10	0.70
Інтеграція з CI/CD	0.15	0.10	0.50	0.30	0.30	1.00
Обчислювальні ресурси	0.10	1.00	1.00	0.60	0.20	1.00
Складність впровадження	0.05	1.00	0.70	0.30	0.30	0.70
Інтегральна оцінка	1.00	0.395	0.8025	0.6400	0.5725	0.9075

Джерело: сформовано автором на підставі (Kuhn, Kacker та Lei, 2013; Mandl, 1985; Nie та Leung, 2011; Suafel та Harman, 2019; Segall та Tzoref-Brill, 2018; Durelli, Durelli та Endo, 2019; Grindal, Offutt та Andler, 2005; Kuhn, Wallace та Gallo, 2004; Cohen, Gibbons, Mugridge та Colbourn, 2003; Lei та Tai, 1998).

Висновки. У процесі дослідження запропоновано інтегрований підхід, який поєднує комбінаторні методи (Pairwise Testing, Orthogonal Arrays) із генетичними алгоритмами та інтеграцією з CI/CD-пайплайнами, є одним із найбільш ефективних для вибору оптимальних конфігурацій у тестуванні багатокomпонентних інформаційних систем (ІС), і йому слід приділити більше уваги. Розроблені вагові коефіцієнти та порівняльний аналіз дозволяють, з одного боку, визначити сильні й слабкі сторони існуючих методів (ручний відбір, генетичні алгоритми, ШІ-підходи); з іншого – синтезувати підхід, що забезпечує високу ефективність, адаптивність і мінімальні ресурси. Використання цього підходу дозволяє структурувати процес тестування, оптимізувати вибір конфігурацій і адаптуватися до змін у системах, скорочуючи кількість тестів до 5–10% від повного набору при збереженні 90–95% покриття критичних сценаріїв. Встановлено, що серед досліджених методів найбільш адаптивними є комбінаторні методи, зокрема Pairwise Testing, доповнені генетичними алгоритмами та автоматизацією через CI/CD. Подальшого дослідження потребує інтеграція ШІ-компонентів для прогнозування дефектів на основі обмежених даних, розробка методів автоматичного аналізу результатів тестування за допомогою статистичних моделей, а також адаптація підходу до специфічних галузей, таких як фінанси чи інфраструктурні проекти, де критичність сценаріїв і складність систем вимагають особливої уваги.

Список використаних джерел:

1. Bansal S. Empirical Studies on Automated Software Testing Practices : монографія. USC, 2022. URL: https://www.researchgate.net/publication/369475828_Empirical_Studies_on_Automated_Software_Testing_Practices (дата звернення: 22.09.2025).
2. Cohen M. B., Gibbons P. B., Mugridge W. B., Colbourn C. J. Constructing Test Suites for Interaction Testing : матеріали конференції. Proceedings of the 25th International Conference on Software Engineering. 2003. P. 38–48.
3. De Sousa Ribeiro Filho F. Automated security testing in DevSecOps pipelines : стаття. WJARR. 2025. URL: <https://wjarr.com/sites/default/files/WJARR-2024-1083.pdf> (дата звернення: 22.09.2025).
4. Durelli W. H., Durelli R. S., Endo A. T. Applying Machine Learning to Software Testing: A Systematic Review : стаття. IEEE Transactions on Reliability. 2019. Vol. 68, No 3. P. 1189–1212.
5. Grindal M., Offutt J., Andler S. F. Combination Testing Strategies: A Survey : стаття. Software Testing, Verification and Reliability. 2005. Vol. 15, No 3. P. 167–199.
6. Kuhn R., Kacker R., Lei Y. Introduction to Combinatorial Testing : монографія. Boca Raton : CRC Press, 2013. 333 с.
7. Kuhn D. R., Wallace D. R., Gallo A. M. Software Fault Interactions and Implications for Software Testing : стаття. IEEE Transactions on Software Engineering. 2004. Vol. 30, No 6. P. 418–421.
8. Lei Y., Tai K. C. In-Parameter-Order: A Test Generation Strategy for Pairwise Testing : матеріали конференції. Proceedings of the 3rd IEEE International High-Assurance Systems Engineering Symposium. 1998. P. 254–261.
9. Mandl R. Orthogonal Latin Squares: A Tool for the Design of Experiments in Testing : стаття. Software Testing, Verification and Reliability. 1985. Vol. 2, No 2. P. 23–31.

10. Nie C., Leung H. A Survey of Combinatorial Testing : стаття. *ACM Computing Surveys*. 2011. Vol. 43, No 2. P. 1–29.
11. Segall I., Tzoref-Brill R. Using Machine Learning to Improve Test Case Generation : стаття. *IEEE International Conference on Software Testing, Verification and Validation*. 2018. P. 45–53.
12. Sharma A. Test Suite Optimization Using Machine Learning Techniques : монографія. DSU, 2024. URL: https://www.researchgate.net/publication/385478252_Test_Suite_Optimization_Using_Machine_Learning_Techniques_A_Comprehensive_Study (дата звернення: 22.09.2025).
13. Suafel L., Harman M. Evolutionary Algorithms for Software Testing: A Survey : стаття. *Journal of Systems and Software*. 2019. Vol. 152. P. 112–124.

Дата надходження статті: 22.09.2025

Дата прийняття статті: 20.10.2025

Опубліковано: 04.12.2025