УДК 004.4

І. І. КУЗНЕЦЬ
Л.Б. ЛІЩИНСЬКА

АВТОМАТИЧНИЙ РЕФАКТОРИНГ ПРОГРАМНОГО КОДУ НА ОСНОВІ ГЕНЕТИЧНИХ АЛГОРИТМІВ ТА ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ У ВИСОКОЕФЕКТИВНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ

Вінницький національний технічний університет

Анотація

У роботі розглядається модель автоматичного рефакторингу програмного коду, в якій генетичний алгоритм формує змішані послідовності каталожних і LLM-згенерованих рефакторингових перетворень. Мета – багатокритеріальне покращення метрик якості та підтримуваності коду у високоефективних інформаційних системах, що є важливим для розвитку сучасних цифрових технологій та механізмів управління.

Ключові слова: автоматичний рефакторинг, генетичні алгоритми, великі мовні моделі, метрики якості коду, високоефективні інформаційні системи, цифрові технології.

Abstract

The paper considers a model of automated source-code refactoring in which a genetic algorithm generates hybrid sequences of catalog-based and LLM-generated refactoring transformations. The goal is multi-objective improvement of code quality and maintainability metrics in high-performance information systems, which is important for the development of modern digital technologies and management mechanisms.

Keywords: automated refactoring, genetic algorithms, large language models, code quality metrics, high-performance information systems, digital technologies.

Актуальність

Сучасні високоефективні інформаційні системи, які підтримують критично важливі бізнес-процеси, наукові дослідження та цифрове управління, характеризуються значною складністю програмного коду. З часом зростають когнітивна та цикломатична складність, глибина вкладеності, дублювання, погіршується підтримуваність. Традиційні принципи та каталоги рефакторингів, сформульовані М. Фаулером [1], переважно застосовуються вручну або як окремі інструментальні операції й не гарантують системного багатокритеріального покращення якості коду у великих проєктах. Огляди стану автоматичного рефакторингу підтверджують, що проблема зниження технічного боргу залишається актуальною [8]. Тому виникає потреба в автоматизації рефакторингу з урахуванням цілого комплексу метрик якості.

Постановка проблеми та аналіз досліджень

Наявні інструменти, такі як IDE-рефакторинги, лінтери, платформи на кшталт SonarQube та LLM-асистенти, або використовують фіксовані каталожні перетворення, або пропонують окремі варіанти переписаних фрагментів коду, не виконуючи систематичного пошуку послідовностей рефакторингів за сукупністю метрик якості [1; 8]. У наукових роботах виділяють два близькі напрями. Перший – search-based рефакторинг та автоматичний рефакторинг на основі еволюційних методів, де послідовності каталожних рефакторингів розглядаються як особини, що оптимізуються генетичним алгоритмом [2-5; 8]. Другий – рефакторинг із використанням великих

мовних моделей, коли LLM генерує варіанти перетворення коду за семантичними шаблонами [6; 7]. Роботи [2-5; 8] демонструють потенціал генетичних алгоритмів для пошуку послідовностей каталожних рефакторингів, а дослідження [6; 7] показують можливості LLM у задачах переписування коду. Водночас у проаналізованих джерелах відсутні моделі, де генетичний алгоритм працює безпосередньо в просторі змішаних послідовностей каталожних і LLM-згенерованих кроків; таке рішення пропонується в межах даного дослідження.

Мета дослідження

Метою є багатокритеріальне покращення показників якості та підтримуваності програмного коду у високоефективних інформаційних системах шляхом розроблення моделі й методів автоматичного рефакторингу на основі генетичних алгоритмів, у межах яких застосовуються змішані послідовності рефакторингових перетворень двох типів – стандартних каталожних та згенерованих великою мовною моделлю.

Представлення особини

У роботах [2-5; 8] запропоновано розглядати послідовності каталожних рефакторингів як особини генетичного алгоритму, що оптимізуються за метриками якості коду. Роботи [6; 7] демонструють використання великих мовних моделей для автоматизованого перетворення програмного коду. У даному дослідженні ці ідеї об’єднано в єдиній моделі автоматичного рефакторингу, де особина описує послідовність рефакторингових кроків двох типів: каталожні мікро-рефакторинги (перейменування, виділення методу, спрощення умов, усунення дублювання тощо), сформульовані у класичних каталогах рефакторингів [1], та перетворення, згенеровані великою мовною моделлю за формалізованими підказками, зорієнтованими на зменшення складності та покращення читабельності коду [6; 7].

Простір пошуку

Простір пошуку визначається множиною допустимих операторів рефакторингу та правилами їх застосування до вибраних фрагментів коду. Для кожної особини передбачено послідовність перевірок: успішна компіляція, статичний аналіз, за необхідності – прогін тестів. Особини, що порушують коректність, відсіюються.

Функції оцінювання

Оцінювання здійснюється за системою метрик якості коду: когнітивна та цикломатична складність, глибина вкладеності, дублювання, показники підтримуваності тощо. Перелік таких метрик широко використовується у дослідженнях автоматичного рефакторингу та оцінювання підтримуваності програмного забезпечення [5; 8]. Оскільки метрики можуть конфліктувати, задача формулюється як багатокритеріальна: застосовується або агрегація з ваговими коефіцієнтами, або формування Парето-оптимальної множини рішень.

Генетичні оператори

Ініціалізація популяції виконується випадковими, але синтаксично коректними послідовностями змішаних рефакторингових кроків. Селекція орієнтується на особини, що покращують значення цільових метрик без порушення коректності, що узгоджується з практикою search-based рефакторингу [2-5]. Оператори кросоверу та мутації комбінують і варіюють каталожні та LLM-кроки, забезпечуючи дослідження різних комбінацій перетворень.

Програмна реалізація та прикладне використання

Реалізація моделі передбачає модульну архітектуру: модуль обчислення метрик та інтеграції зі статичним аналізом коду (з використанням метрик, прийнятих у дослідженнях підтримуваності [5; 8]); модуль керування LLM-перетвореннями (формування промптів, обмеження обсягу змін, перевірка результатів) [6; 7]; ядро

генетичного алгоритму, що працює з представленням особин і реалізує операції селекції, кросоверу й мутації [2-5]. Для прикладного використання у цифрових технологіях та механізмах управління розглядається сценарій, у якому автоматичний рефакторинг виконується як окремий етап процесу супроводу великих сервісів (систем електронного документообігу, фінансових чи аналітичних платформ). На кожному циклі оновлення коду система формує декілька альтернативних послідовностей змін, оцінює їх за метриками, а розробники обирають варіант, який найкраще відповідає технічним та організаційним вимогам.

Отримані та очікувані результати

На концептуальному рівні сформовано формалізоване представлення змішаних послідовностей каталожних і LLM-згенерованих рефакторингових кроків як особин генетичного алгоритму, систему метрик якості коду для оцінювання кожної особини та структуру генетичного алгоритму, орієнтовану на пошук послідовностей рефакторингів у великому комбінаторному просторі. Очікується, що комбінування каталожних і LLM-згенерованих рефакторингів у межах єдиної моделі дасть змогу зменшити когнітивну та цикломатичну складність коду, знизити дублювання та підвищити підтримуваність без втрати коректності, що є суттєвим для високоефективних інформаційних систем.

Висновки та перспективи подальших досліджень

Розглянута модель автоматичного рефакторингу програмного коду об’єднує пошук генетичним алгоритмом у просторі змішаних послідовностей рефакторингів та використання великих мовних моделей для побудови змістовних перетворень коду. Багатокритеріальна оцінка на основі метрик якості дозволяє орієнтуватися не лише на локальне спрощення окремих фрагментів, а й на реальну підтримуваність коду у високоефективних інформаційних системах. Подальша робота пов’язана з розширенням набору підтримуваних метрик, масштабуванням моделі на великі промислові проєкти, дослідженням впливу різних стратегій багатокритеріальної оптимізації на результати та глибшою інтеграцією з інструментами статичного аналізу й моніторингу якості у процесах DevOps.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Fowler M. Refactoring: Improving the Design of Existing Code. – 2nd ed. – Boston : Addison-Wesley, 2019. – 448 с.
2. O’Keeffe M., Ó Cinnéide M. Search-based refactoring: An empirical study // Journal of Software Maintenance and Evolution: Research and Practice. – 2008. – Vol. 20, No. 5. – С. 345–364.
3. O’Keeffe M., Ó Cinnéide M. Search-based refactoring for software maintenance // Journal of Systems and Software. – 2008. – Vol. 81, No. 4. – С. 502–516.
4. Kebir S., Borne I., Meslati D. A genetic algorithm-based approach for automated refactoring of component-based software // Information and Software Technology. – 2017. – Vol. 88. – С. 17–36.
5. Saheb Nasagh R., Shahidi M., Ashtiani M. A fuzzy genetic automatic refactoring approach to improve software maintainability and flexibility // Soft Computing. – 2021. – Vol. 25, No. 6. – С. 4295–4325.
6. Liu B., Jiang Y., Zhang Y., Niu N., Li G., Liu H. Exploring the potential of general-purpose LLMs in automated software refactoring: An empirical study // Automated Software Engineering. – 2025. – Vol. 32. – Article 26.
7. Shirafuji A., Oda Y., Suzuki J., Morishita M., Watanobe Y. Refactoring programs using large language models with few-shot examples // Proceedings of the 30th Asia-Pacific Software Engineering Conference (APSEC 2023). – 2023. – С. 151–160.
8. Baqais A. A. B., Alshayeb M. Automatic software refactoring: A systematic literature review // Software Quality Journal. – 2020. – Vol. 28, No. 2. – С. 459–502.

Кузнець Ілля Ігорович – студент групи F2-125a, факультет інформаційних технологій та комп’ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: illiakuznets@gmail.com

Ліщинська Людмила Броніславівна – д-р техн. наук, професор, професор кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: llb@vntu.edu.ua

Kuznets Illia Ihorovych – student of group F2-125a, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: illiakuznets@gmail.com

Lishchynska Lyudmyla Bronislavivna – Dr. Sc. (Eng.), Full Professor, Professor of Program Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: llb@vntu.edu.ua