

Підсумовуючи, можна дійти висновку, що генетичні алгоритми використовуються для розв'язання багатьох оптимізаційних задач, оскільки можуть добре пристосовуватися до різних умов, водночас не втрачаючи ні якості, ні швидкість прийняття рішень. Саме через це ГА і далі будуть актуальними, незважаючи на стрімкий розвиток технологій.

### Список використаних джерел

1. Мороз О. Г. Аналіз застосування генетичних алгоритмів в задачах глобальної оптимізації. *Control systems and computers*. 2018. № 2. С. 68–79.
2. Гриник Р. О. Застосування генетичного алгоритму для вирішення задач криптоаналізу. *Інформаційно-комунікаційні технології в сучасній освіті: досвід, проблеми, перспективи*: матеріали IV Міжнародної науково-практичної конференції. 21–22 жовт. 2015 р.: зб. наук. пр. Ч. 1. Львів, Вид-во ЛДУ БЖД, 2015. С. 168–170.
3. Мельник А. М., Босько В. В., Резніченко В. А. Дослідження сучасних методів роботи та напрямків застосування генетичних алгоритмів. *Інформаційні технології в економіці, медицині та освіті*: матеріали VI Міжнародної науково-практичної конференції (м. Кропивницький, 20–21 квітня 2023 р.). Кропивницький, 2023. С. 75–76.
4. Січко Т. В., Нескородєва Т. В. Методичні вказівки щодо виконання лабораторних робіт з дисципліни «Методи оптимізації та дослідження операцій» для студентів СО «Бакалавр» денної та заочної форм навчання спеціальностей 122 «Комп'ютерні науки», 113 «Прикладна математика». Вінниця: ДонНУ імені Василя Стуса. 2020, 104 с.

### УДК 004.4:004.021

*Юстименко Є. А., здобувач 3 курсу спеціальності 122 Комп'ютерні науки,  
Хмелівський Ю. С., асистент кафедри інформаційних технологій*

### МЕТОДИ ОПТИМІЗАЦІЇ АЛГОРИТМІВ У ПРОГРАМУВАННІ

*Донецький національний університет імені Василя Стуса, м. Вінниця*

Зі стрімким розвитком інформаційних технологій у світі майже кожна сфера нашого життя стала напряду пов'язана з простими або складними інформаційними системами, які виконують ту чи іншу дію. Зокрема, інформаційні системи побудовані з використанням різних алгоритмів, які і підтримують роботу таких систем. Оптимізація алгоритмів у програмуванні відіграє ключову роль

у покращенні ефективності програм та забезпеченні швидкодії і низької обчислювальної складності. Алгоритми застосовуються в різних сферах, від програмування та обчислювальної техніки до математики, науки про дані, логістики, телекомунікацій та інших галузей.

Алгоритм – набір дій та процедур, які потрібні для виконання певного завдання. Ці дії мають певну послідовність, деякі з них можуть повторюватись певну кількість разів для досягнення завдання цього алгоритму [1].

Основна мета алгоритму – розробити такий набір кроків, який буде чітко визначений та може бути виконаний механічно без будь-яких уявлень про специфічні обставини. Він може бути використаний для розв’язання певної задачі у багатьох різних умовах.

Головною характеристикою алгоритму для його оптимізації є складність алгоритму. Складність алгоритму – міра, яка визначає, наскільки затратний алгоритм під час виконання всіх дій алгоритму.

Розглянемо основні аспекти складності алгоритму:

➤ часова складність – вимірює кількість операцій або час, необхідний для виконання алгоритму у відношенні до розміру вхідних даних. Часова складність вказує, як швидко збільшується час виконання алгоритму під час збільшення об’єму вхідних даних;

➤ просторова складність – вимірює кількість пам’яті, яка використовується алгоритмом у відношенні до розміру вхідних даних. Просторова складність вказує на об’єм пам’яті, який алгоритм потребує для своєї роботи;

➤ «О» (О-нотація) – використовується для оцінки верхньої межі часової або просторової складності алгоритму у відношенні до розміру вхідних даних. У термінах О-нотації  $O(f(n))$  визначає, як змінюється часова або просторова складність алгоритму зі збільшенням розміру вхідних даних ( $n$ ).

Розглянемо основні методи оптимізації алгоритмів у програмуванні.

➤ Аналіз алгоритму – один із основних методів оптимізації алгоритмів. Оцінка часової та просторової складності дає змогу визначити найбільш обтяжливі частини коду. Змінивши ці частини коду або вилучивши непотрібні дії, можна значно знизити складність алгоритмів.

➤ Використання ефективних структур даних є ключовим для оптимізації алгоритмів. Застосування оптимальних масивів, списків, черг або дерев дає змогу значно зменшити час доступу та операцій із даними.

➤ Оптимізація циклів та рекурсії. Уникнення зайвих ітерацій у циклах, використання оптимізованих ітераційних конструкцій (як-от індексація замість пошуку) допомагає покращити швидкість виконання. Також оптимізація рекурсивних викликів (наприклад, застосування хвостової рекурсії) зменшує використання пам’яті та збільшує ефективність алгоритму.

➤ Паралельна обробка та асинхронність дає змогу використовувати ресурси більш ефективно. Використання багатопотоковості або асинхронних функцій може значно зменшити час виконання завдань.

➤ Оптимізація самого алгоритму часто є ключем до покращення продуктивності. Від вибору правильного алгоритму для конкретної задачі до оптимізації його роботи важливою стає глибока розробка самого алгоритму.

Зважаючи на постійний розвиток технологій та зростання обсягів даних, оптимізація алгоритмів стає надзвичайно важливою для розробників програмного забезпечення. Прагнення до швидкодії, оптимального використання ресурсів та забезпечення мінімальних обчислювальних витрат визначає успіх багатьох сучасних програм.

Застосування методів оптимізації, що були розглянуті, дає змогу покращити ефективність програм та алгоритмів. Використання ефективних структур даних, оптимізація циклів і рекурсії, а також паралельна обробка дають змогу створювати програмне забезпечення, яке працює швидко та ефективно за будь-яких умов.

### **Список використаних джерел**

1. Алгоритм. URL: <https://ua5.org/osnprog/187-algorithm.html> (дата звернення: 12.11.2023).

2. Algorithms optimization. URL: [https://d2l.ai/chapter\\_optimization/](https://d2l.ai/chapter_optimization/) (дата звернення: 13.11.2023).

3. Оптимізація алгоритмів. URL: <https://dou.ua/forums/topic/39967/> (дата звернення: 12.11.2023).