

## РОЗВ'ЯЗАННЯ ЗАДАЧІ ПОШУКУ ЕКСТРЕМУМУ ФУНКЦІЇ ГЕНЕТИЧНИМ АЛГОРИТМОМ

Вінницький національний технічний університет

### *Анотація*

У даній роботі проаналізовано групу адаптивних пошукових механізмів, під назвою генетичні алгоритми. Показано приклад застосування даного алгоритму для задачі пошуку екстремуму мультимодальної функції багатьох змінних.

**Ключові слова:** оптимізація, екстремум, еволюція, природний відбір, генетичний алгоритм, кроссовер, мутація, популяція.

### *Abstract*

In this paper is analyzed of the group of adaptive search engines, that are called genetic algorithms. An example of an application of this algorithm for the task of searching an extremum of the multimodal function of many variables is shown.

**Keywords:** optimization, extremum, evolution, natural selection, genetic algorithm, crossover, mutation, population.

### Вступ

Еволюційні алгоритми є основою сучасних евристичних комп'ютерних технологій оптимізації, навчання, моделювання, проектування й управління у найширшому значенні цих понять. Вони ґрунтуються на глибокій аналогії між біологічним генетичним і комп'ютерним двійковим кодом. Ця аналогія дає змогу формалізувати біологічний еволюційний процес і застосувати ідею про природний відбір для розв'язання різноманітних складних задач [1]. Фактично генетичний алгоритм – це проста модель еволюції в природі, яка реалізована у виді комп'ютерної програми, яка дозволяє розв'язати певну задачу за поліноміальний час [2]. Суть алгоритму полягає в тому, що більш пристосовані особини мають більше можливостей для виживання і розмноження, а тому дають більше нащадків. При цьому завдяки передаванню генетичної інформації (*генетичному спадкуванню*) нащадки успадковують від батьків їхні основні якості. Рекомбінація і мутація дозволяють змінюватися особинам і пристосовуватися до середовища. Таким чином, нащадки сильних індивідуумів будуть відносно добре пристосованими, а їхня частка в загальній масі особин буде зростати. Після зміни декількох десятків або сотень поколінь середня пристосованість особин даного виду помітно зростає. Дані алгоритми відносяться до адаптивних пошукових механізмів [3-5].

**Метою роботи** є аналіз генетичного алгоритму та його застосування до задачі пошуку екстремуму мультимодальної функції багатьох змінних.

### Результати дослідження

На самому початку роботи алгоритму випадковим чином ініціалізується популяція, яка має такі характеристики, як: чисельність, кількість генів у кожній особині і їх розрядність. Особини популяції оцінюються відповідно до вибраного критерію, і в результаті визначається їх пристосованість. Критерієм виступає деяка функція оптимізації  $F(g_1, g_2, \dots, g_n)$ , де  $g_i$  –  $i$ -ий ген особини, яка оцінюється. Наприклад, у теорії керування функція пристосованості може приймати вид функції похибки, а в теорії ігор – вартісної функції. На кожній ітерації генетичного алгоритму пристосованість кожної особини даної популяції оцінюється за допомогою функції пристосованості, і на цій основі створюється наступна популяція особин, які складають множину потенційних рішень задачі оптимізації. Чим вище пристосованість особини, тим більша ймовірність того, що вона візьме участь

в схрещуванні (кроссовері). Під час схрещування двох особин відбувається «обмін генетичною інформацією», тобто відповідні гени обмінюються бітами. Випадковим чином визначається точка (або точки) всередині хромосоми – точка розриву або *кроссоверу*, у якій обидві хромосоми діляться на дві частини і обмінюються ними. Ймовірність застосування оператора схрещування перевищує 50%. Потім застосовують оператор мутації, який «вибиває» популяцію з локального екстремуму і захищає від передчасної збіжності. Це досягається за рахунок того, що інвертується випадково вибраний біт у хромосомі. Ймовірність мутації значно менша ймовірності кроссовера і рідко перевищує 10-30%. Отримані в результаті даних операцій нащадки формують нове покоління, і все повторюється спочатку, але вже для щойно створеного покоління. Алгоритм припиняє роботу, якщо знайдено рішення, минув встановлений час роботи або популяція тривалий час не прогресує. Після закінчення роботи найбільш пристосована особина популяції, точніше її гени, будуть представляти рішення задачі [3, 5].

Ефективність роботи генетичного алгоритму прийнято оцінювати кількістю обчислень цільової функції (чим менше – тим краще). Наведемо приклад його роботи для функції зі складним рельєфом Растрігіна (рис. 1) [6]:

$$F(x_1, \dots, x_n) = 10 \cdot n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)), \quad x_i \in [-5.12, 5.12]. \quad (1)$$

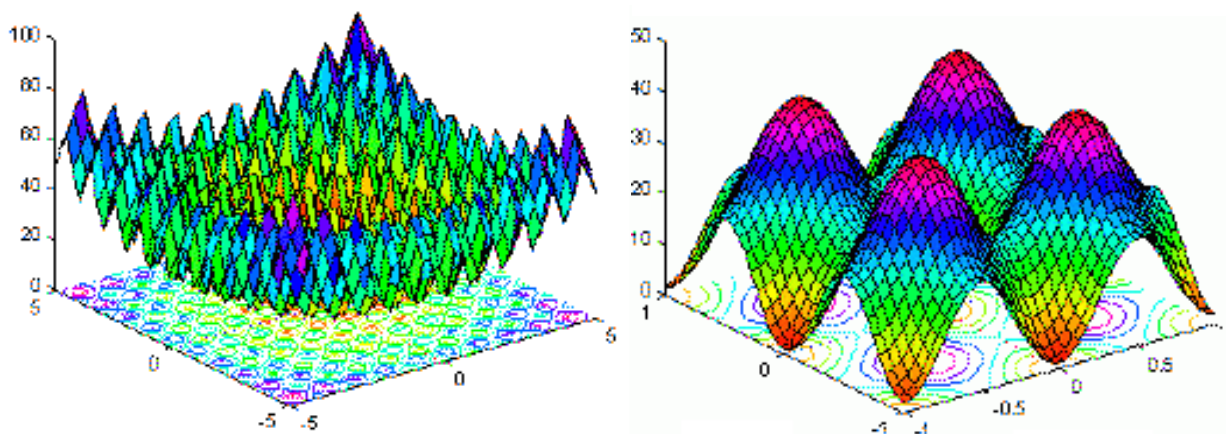


Рисунок 1 — Тестова функція Растрігіна

Результати цільової функції менші 0,001 зараховувалися як знайдений глобальний оптимум (мінімум  $F(x_i^*) = F(0) = 0$ ). Оскільки генетичні алгоритми використовують стохастичність, то для того, щоб визначити, наскільки ефективний алгоритм потрібно запустити його на одній і тій же тестовій функції кілька разів. Після цього можна проаналізувати результат. Параметри запуску програми такі: фіксований розмір популяції та розрядності генів; кількість точок кроссоверу дорівнює кількості генів (на кожен ген приходить одна точка); для схрещування відбираються 50% популяцій; видалення однакових особин з популяції відбувається за допомогою мутації; ймовірність мутації 95%; дві "елітні" особини; точність обчислень складає 0,001; 50 запусків програми [7, 8].

Результати роботи програми:

☞  $n = 20$ , кількість знахождень глобального мінімуму 16%, кількість обчислень цільової функції не більша 5000, максимальне значення функції 19,8992.

☞  $n = 30$ , кількість знахождень глобального мінімуму 26%, кількість обчислень цільової функції не більша 10000, максимальне значення функції 17,997.

☞  $n = 50$ , кількість знахождень глобального мінімуму 38%, кількість обчислень цільової функції не більша 10000, максимальне значення функції 49,869.

### Висновки

У даній роботі розглянуто застосування генетичних алгоритмів до задач пошуку екстремумів мультимодальних функцій багатьох змінних. Показано приклад розрахунків із використанням функції Растрігіна. Слід відзначити, що глобальний екстремум у більшості випадків був отриманий при

еволюціонуванні 50-60-го покоління. Інколи при вирішенні складних задач допомагає збільшення кількості генерацій для популяцій. Але з цим треба бути обережними та дозволяти алгоритму еволюціонувати тільки в тому випадку, якщо значення функції пристосованості поступово спадає (при знаходженні глобального мінімуму).

#### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Кононюк А.Ю. Нейронні мережі і генетичні алгоритми / А.Ю. Кононюк. – К., 2008. – 446 с.
2. Иванов Ю.Ю. Дискретна математика: лекції, алгоритми та задачі / Ю.Ю. Иванов. – Вінниця, 2018. – 108 с. – Режим доступу: [https://iq.vntu.edu.ua/method/read\\_url.php?tbl\\_num=2&url=/fdb/1166/Discrete\\_Math\\_by\\_IVANOV.djvu](https://iq.vntu.edu.ua/method/read_url.php?tbl_num=2&url=/fdb/1166/Discrete_Math_by_IVANOV.djvu).
3. Whitley D. A Genetic Algorithm Tutorial / Statistics and Computing // D. Whitley. – Springer Netherlands, 1994. – V. 4(2). – P. 65-85.
4. Курейчик В.М. Генетические алгоритмы / В.М. Курейчик. – Таганрог: ТРТУ, 1998. – 239 с.
5. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. – М.: Горячая линия-Телеком, 2004. – 452 с.
6. Пантелеев А.В. Методы оптимизации в примерах и задачах: учеб. пособие / А.В. Пантелеев, Т.А. Летова. – М.: Высш. шк., 2005. – 544 с.
7. Норенков И.П. Эвристики и их комбинации в генетических методах дискретной оптимизации / Информационные технологии // И.П. Норенков. – 1999. №1. – С. 2-7.
8. Кравець І.М. Оптимізація багатоекстремальних функцій за допомогою генетичного алгоритму / Проблеми інформатизації та управління // І.М. Кравець. – К.: Інститут комп'ютерних технологій Національного авіаційного університету, 2010. – № 2(30). – С. 95-99.

**Головня Віталій Вікторович** — студент групи СІ-17мс, факультет комп'ютерних систем і автоматики, Вінницький національний технічний університет, м. Вінниця.

**Кривогубченко Сергій Григорович** — канд. техн. наук, доцент кафедри автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м. Вінниця.

**Іванов Юрій Юрійович** — канд. техн. наук, старший викладач кафедри автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м. Вінниця, e-mail: Yura881990@i.ua.

**Holovnya Vitaliy V.** — student, Faculty of Computer Systems and Automation, Vinnytsia National Technical University, Vinnytsia.

**Krivogubchenko Sergiy G.** — Cand. Sc. (Eng), Docent, Faculty of Computer Systems and Automation, Vinnytsia National Technical University, Vinnytsia.

**Ivanov Yuriy Yu.** — Cand. Sc. (Eng), Senior Lecturer, Faculty of Computer Systems and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: Yura881990@i.ua.