###### **МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**З В І Т**

**Лабораторна робота №7  
з дисципліни**

**«Сучасні методи та моделі  
інтелектуальних систем керування»**

Виконавець:

аспірант групи АКІТР-23-1а Косей М.П.

Керівник:

викладач Тиханський М. П.

2024

**Лабораторна робота №7**

# Стислі теоретичні відомості

Основи теорії генетичних алгоритмів сформульовані Холландом в його основоположній роботі і в подальшому були розвинені рядом дослідників. Найбільш відомою і цитованої в даний час є монографія Д. Голдберга, де систематично викладено основні результати та області практичного застосування генетичних алгоритмів.

Генетичні алгоритми використовують принципи і термінологію, запозичені у біологічної науки – генетики. У генетичних алгоритмах кожна особина представляє потенційне рішення деякої проблеми. У класичному генетичному алгоритмі особина кодується рядком двійкових символів – хромосомою, кожен біт якої називається геном. Множина особин – потенційних рішень становить популяцію. Пошук (суб) оптимального вирішення проблеми виконується в процесі еволюції популяції – послідовного перетворення однієї кінцевої множини рішень у іншу за допомогою генетичних операторів репродукції, кросинговеру та мутації.

Еволюційні обчислення використовують наступні три механізми природної еволюції:

* перший принцип заснований на концепції виживання найсильніших і природного відбору за Дарвіном який був сформульований їм у 1859 році в книзі «Походження видів шляхом природного відбору». Згідно Дарвіну особини, які краще здатні вирішувати завдання в своєму середовищі, виживають і більше розмножуються (репродуцюють). У генетичних алгоритмах кожна особина представляє собою рішення деякої проблеми аналогією з цим принципом особини з кращими значеннями цільової (фітнес) функції мають великі шанси вижити і репродуцювати. Формалізація цього принципу дає оператор репродукції;
* другий принцип обумовлений тим фактом, що хромосома нащадка складається з частин, отриманих з хромосом батьків. Цей принцип був відкритий в 1865 році Менделем. Його формалізація дає основу для оператора схрещування (кросинговеру);
* третій принцип заснований на концепції мутації, відкритої в 1900 році де Вре. Спочатку цей термін використовувався для опису істотних (різких) змін властивостей нащадків і набуття ними властивостей, відсутніх у батьків. За аналогією з цим принципом генетичні алгоритми використовують подібний механізм для різкої зміни властивостей нащадків і тим самим, підвищують різноманіття особин в популяції (множині рішень).

Ці три принципи складають ядро еволюційних обчислень. Використовуючи їх, популяція (множина рішень даної проблеми) еволюціонує від покоління до покоління.

Генетичний алгоритм бере множину параметрів оптимізаційної проблеми та кодує їх послідовностями кінцевої довжини в деякому кінцевому алфавіті (в найпростішому випадку двійковий алфавіт «0» і «1»).

Простий генетичний алгоритм випадковим чином генерує початкову популяцію стрінгів (хромосом). Потім алгоритм генерує наступне покоління (популяцію), за допомогою трьох основних генетичних операторів:

* оператор репродукції (ОР);
* оператор схрещування (кросинговеру, ОК);
* оператор мутації (ОМ).

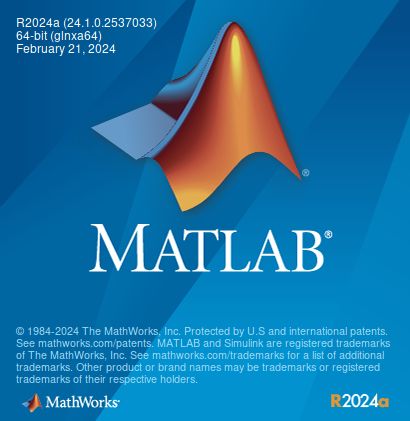
Генетичні оператори є математичної формалізацією наведених вище трьох основоположних принципів Дарвіна, Менделя і де Вре природної еволюції.

Генетичний алгоритм працює до тих пір, поки не буде виконано задану кількість поколінь (ітерацій) процесу еволюції або на деякій генерації буде отримано задану якість або внаслідок передчасної збіжності при попаданні в деякий локальний оптимум.

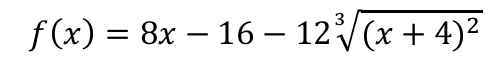
У кожному поколінні множина штучних особин створюється з використанням старих і додаванням нових з хорошими властивостями. Генетичні алгоритми – не просто випадковий пошук, вони ефективно використовують інформацію, накопичену в процесі еволюції. У процесі пошуку рішення необхідно дотримувати баланс між «експлуатацією» отриманих на поточний момент кращих рішень і розширенням простору пошуку. Різні методи пошуку вирішують цю проблему по- різному.

**2) Практична частина**

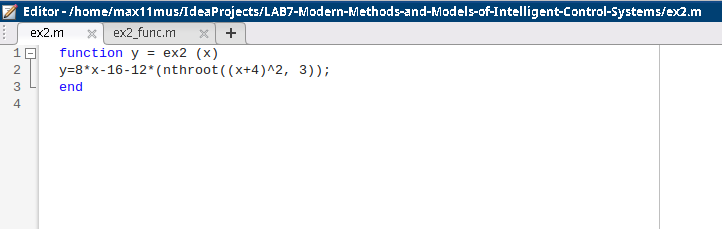
Використовуємо **MATLAB** версія R2024a для Linux.

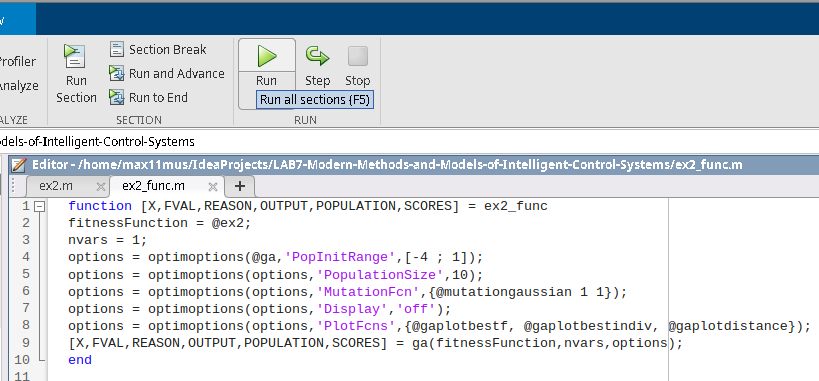


Необхідно мінімізувати функцію однієї змінної:

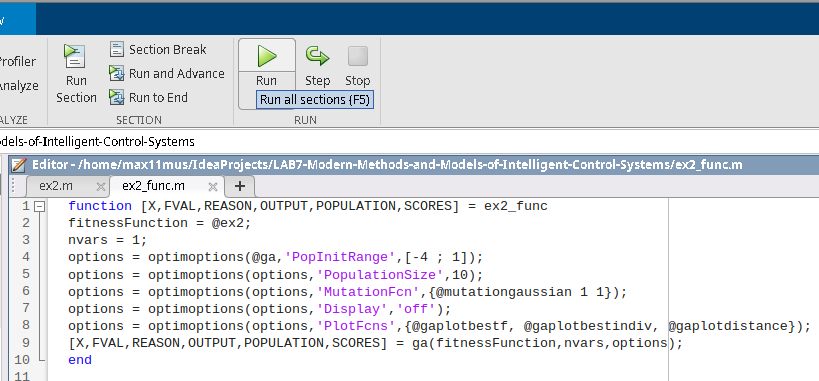


Для цього, у середовищі *MATLAB®* необхідно створити два скрипти ex2.m та ex2\_func.m.



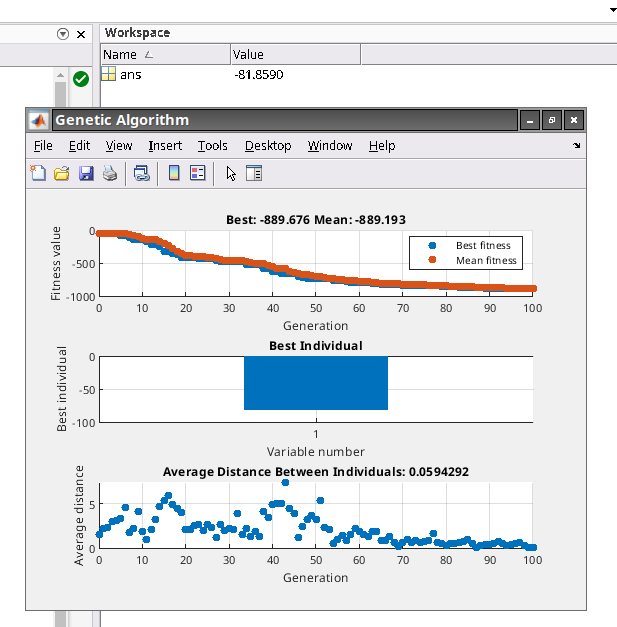


Запускаємо декілька разів оптимізацію функції

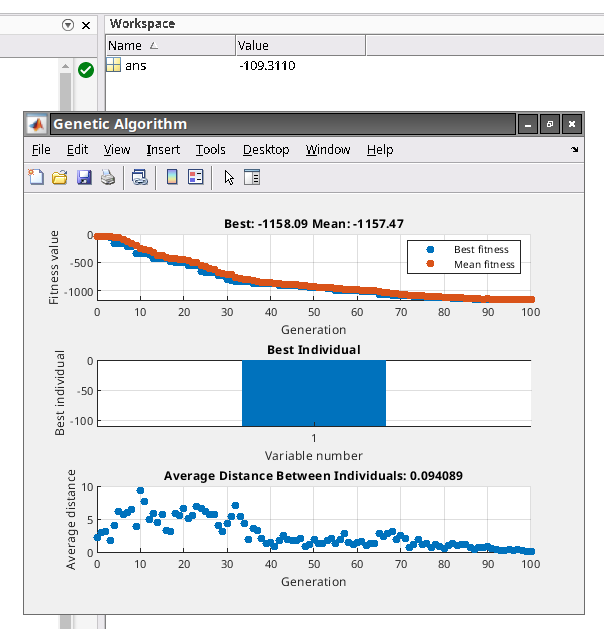


Результати оптимізації

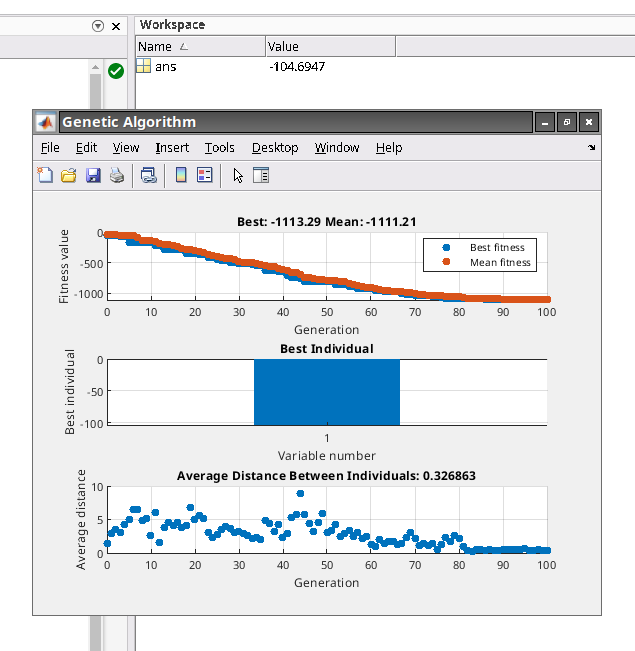
1-ша оптимізація



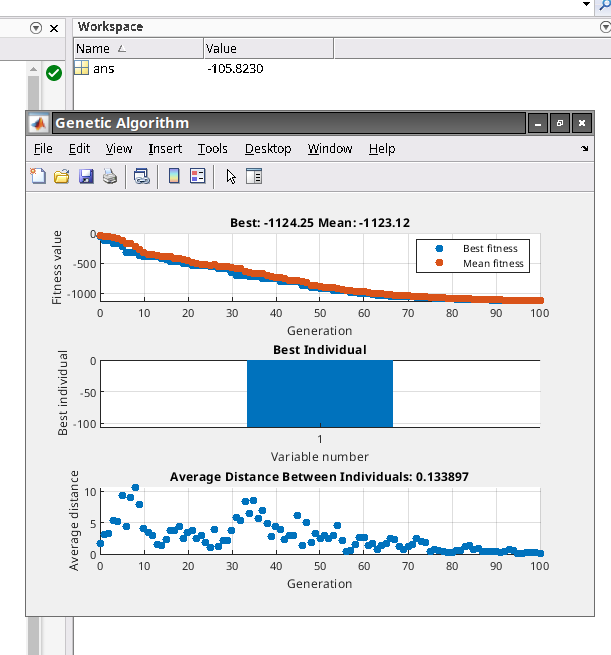
2-а оптимізація



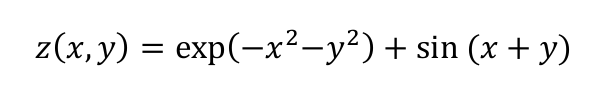
3-я оптимізація



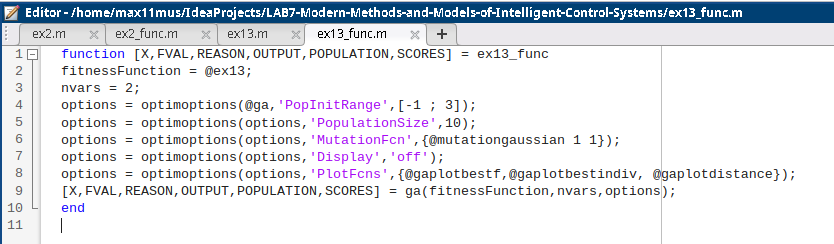
4-а оптимізація



Необхідно мінімізувати функцію двох змінних:



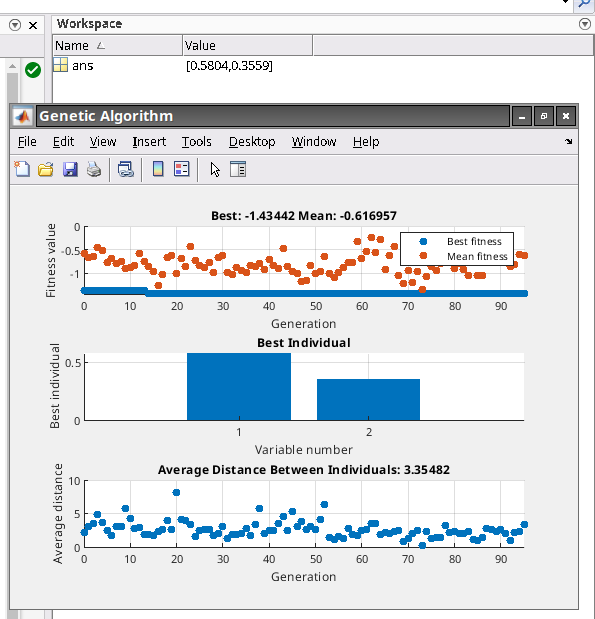
Для цього, у середовищі *MATLAB®* необхідно створити два скрипти ex13.m та ex13\_func.m.



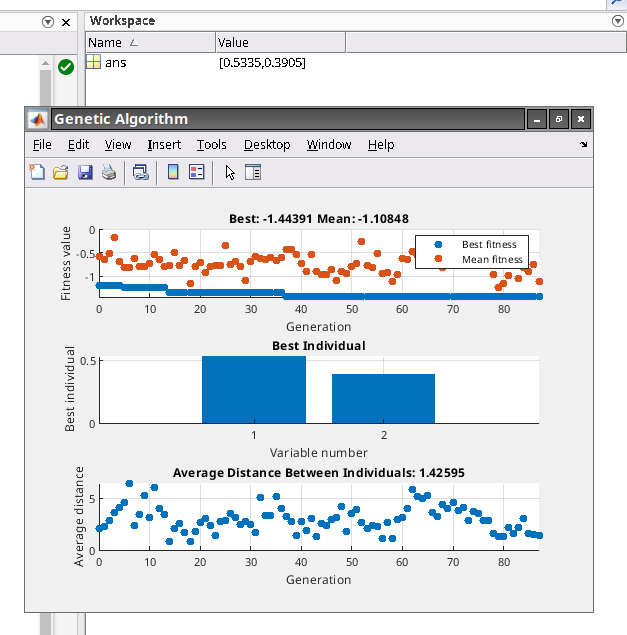
Запускаємо декілька разів оптимізацію функції

Результати оптимізації:

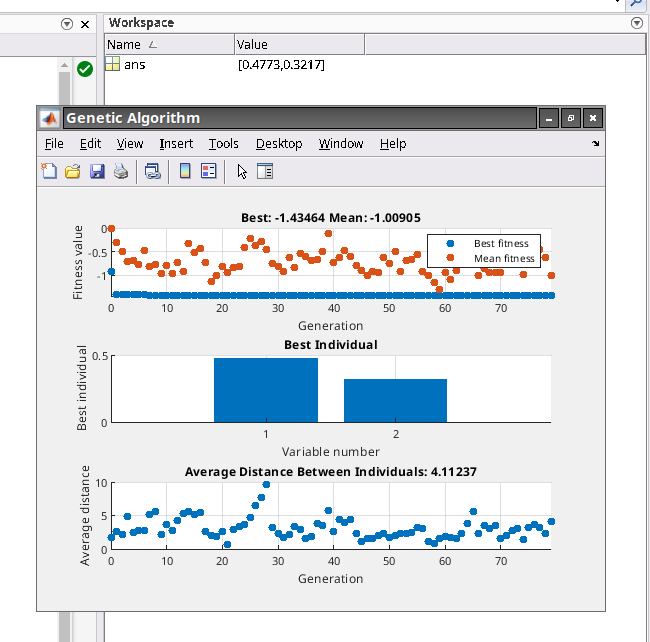
1-ша оптимізація



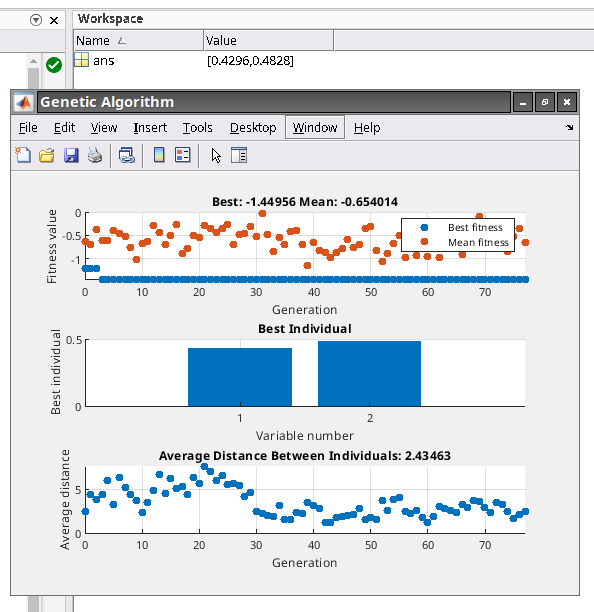
2-а оптимізація



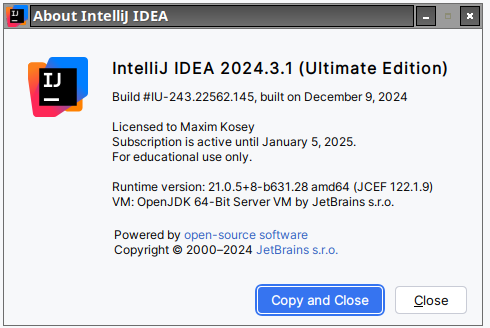
3-я оптимізація



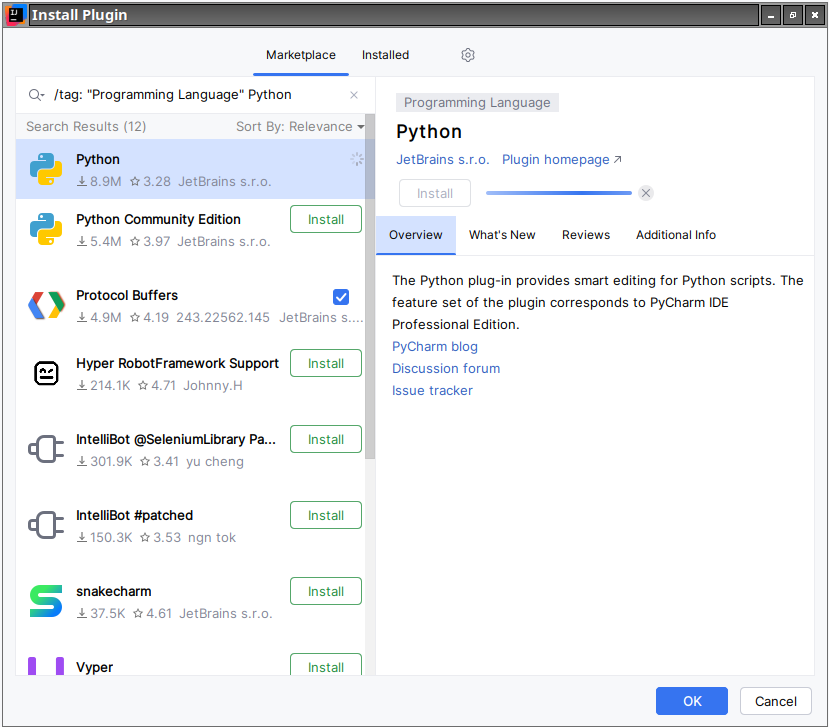
4-а оптимізація



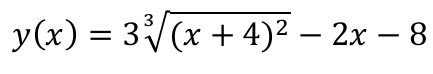
Зробимо оптимізацію інших функцій за допомогою мови програмування **Python** та **IDE InteliJ IDEA**



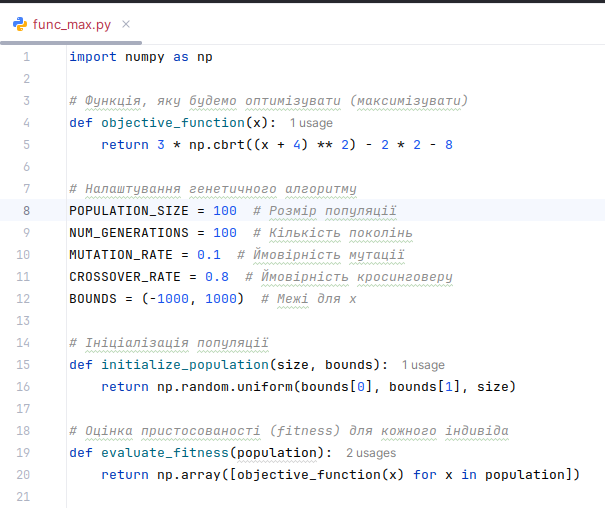
Встановлюємо модуль Python

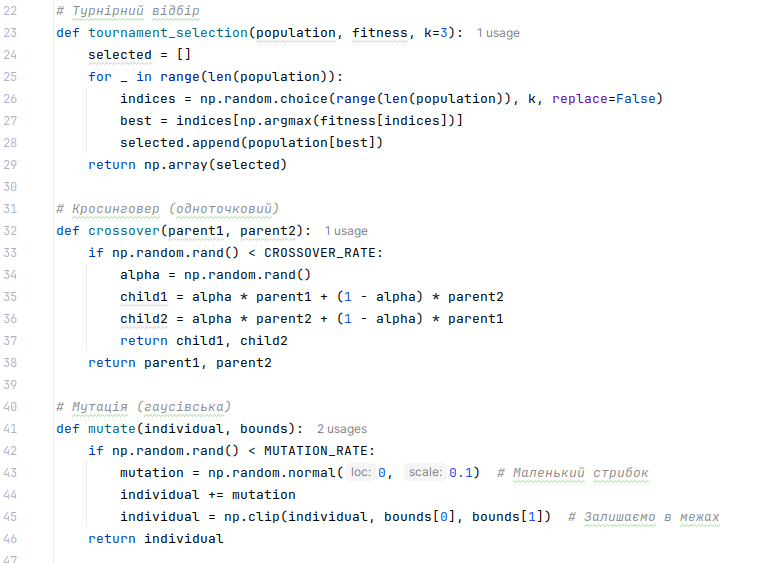


Необхідно максимізувати функцію однієї змінної:



Реалізація генетичного алгоритму на мові Python



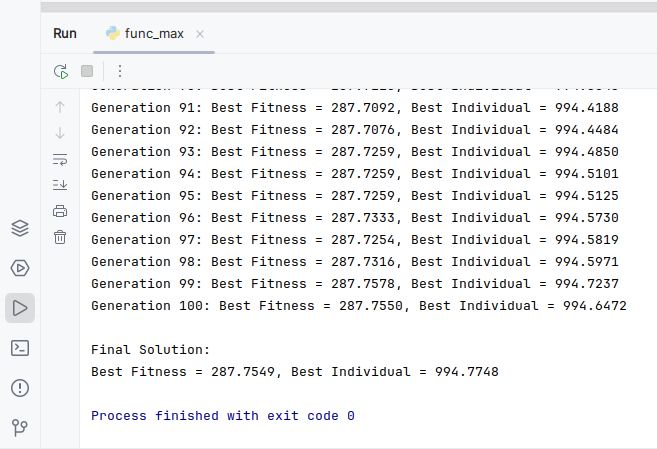




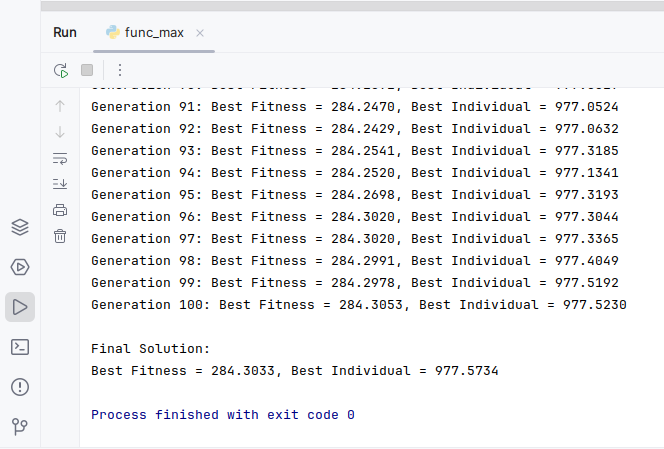
Запускаємо декілька разів оптимізацію функції

Результати оптимізації:

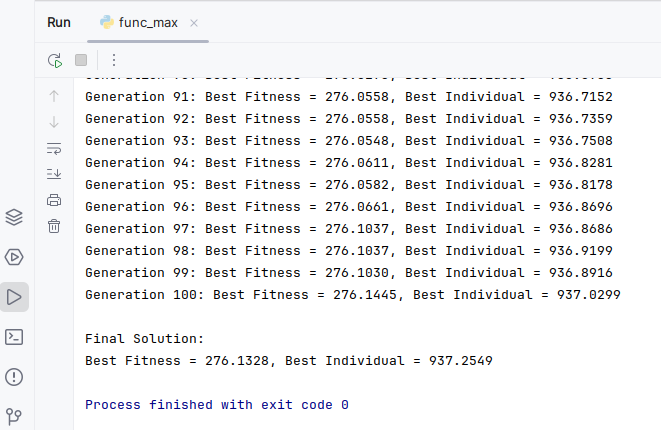
1-ша оптимізація



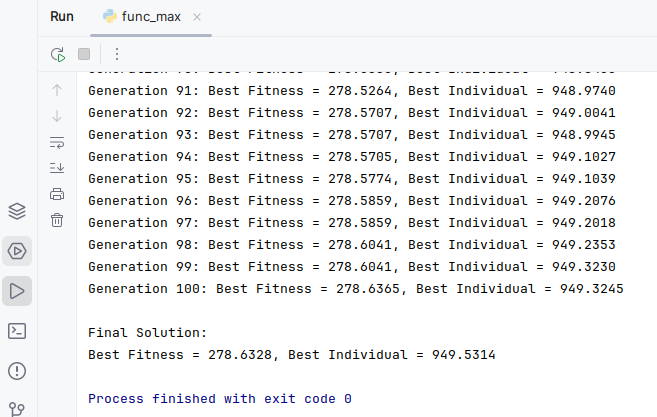
2-а оптимізація



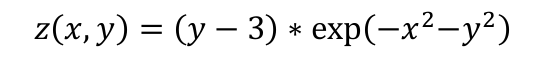
3-я оптимізація



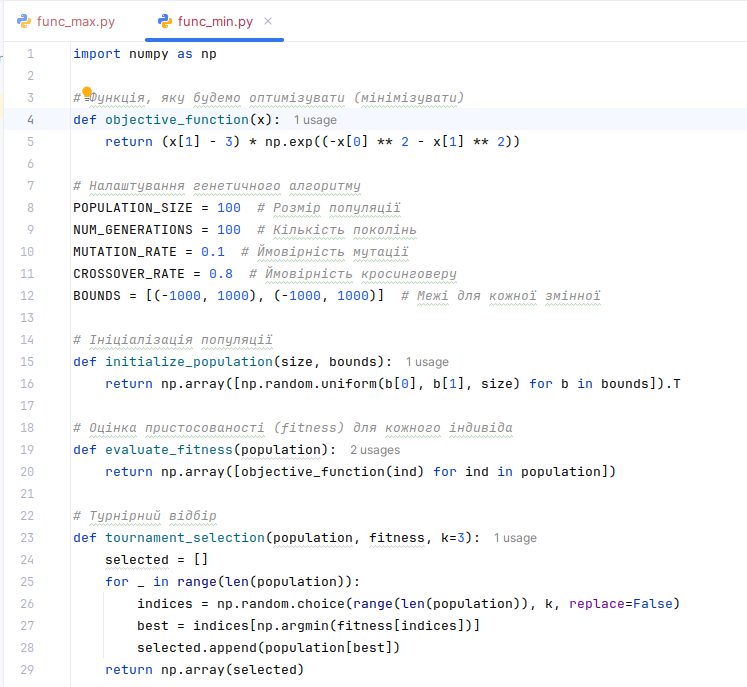
4-а оптимізація

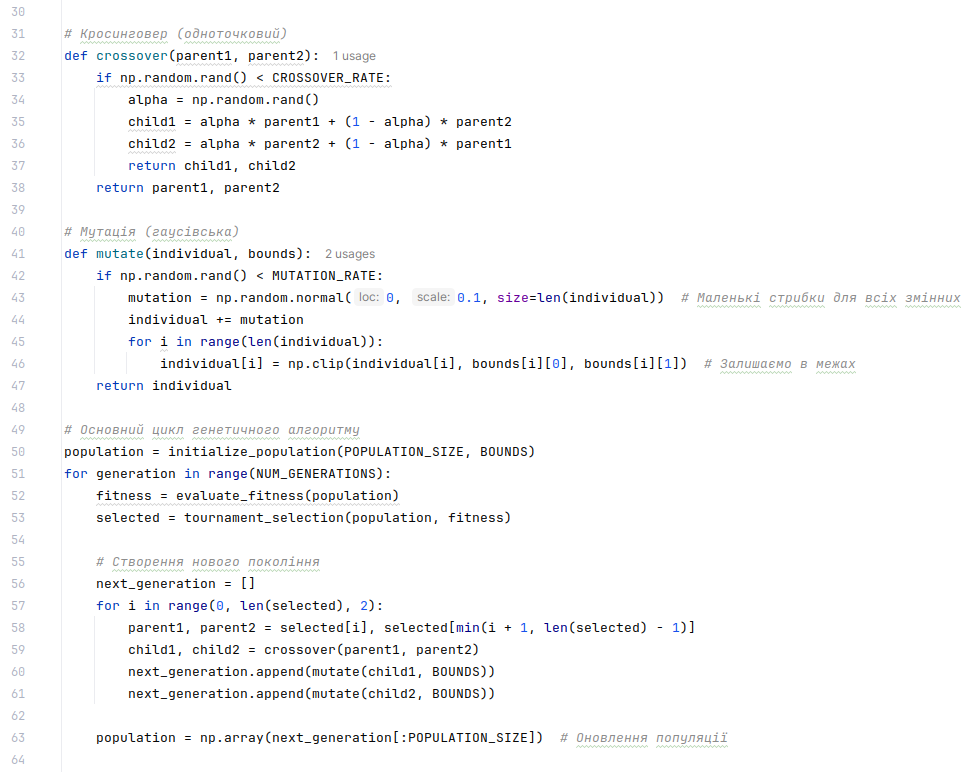


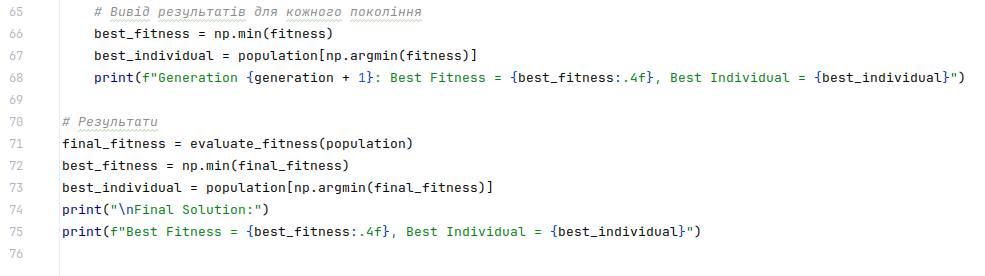
Мінімізуємо функцію



Реалізація генетичного алгоритму на мові Python

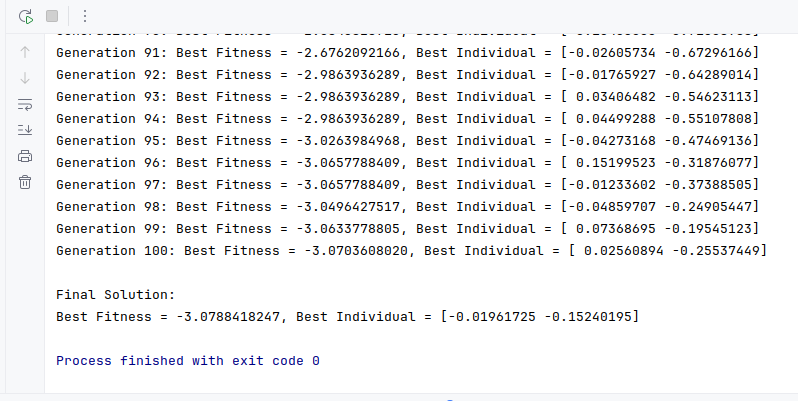


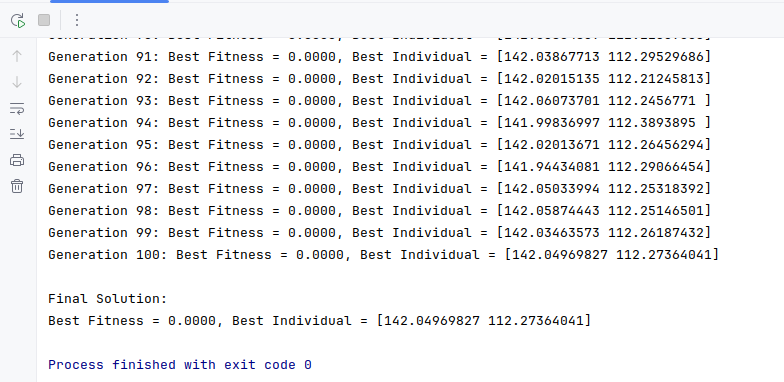


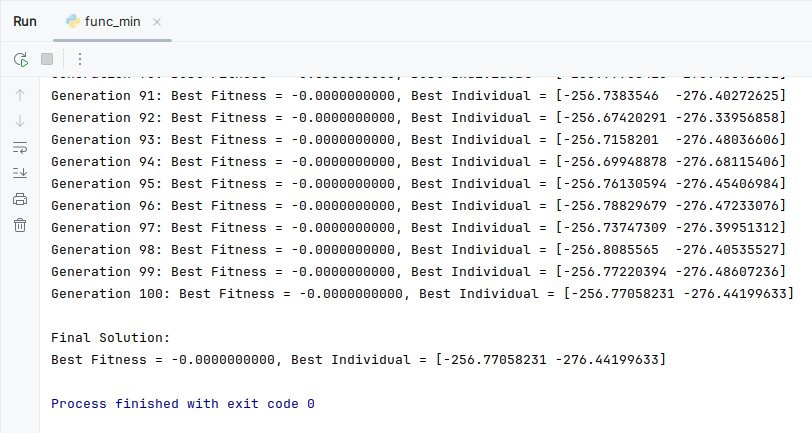


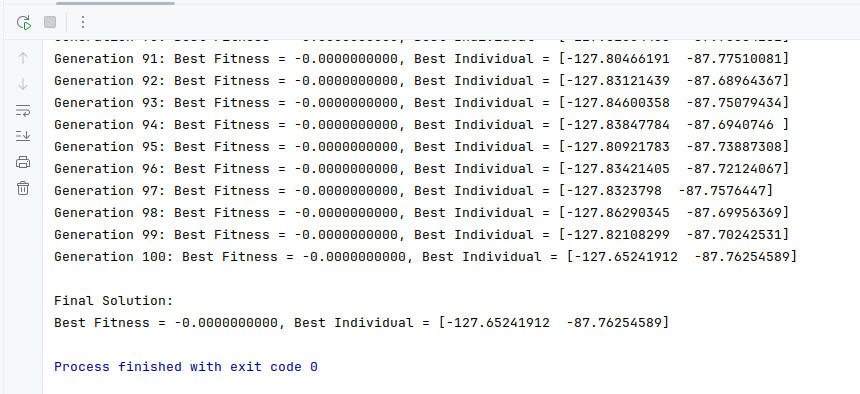
Запускаємо 4-ри рази оптимізацію функції

Результати оптимізації:









**ВИСНОВКИ**

**В результаті виконаної лабораторної роботи за допомого пакета Matlab 2024a та мови програмування Python була досліджена оптимізація функцій за допомогою генетичних алгоритмів.**

**Усі матеріали викладенні у репозіторії GitHub, за посиланням** <https://github.com/Max11mus/LAB7-Modern-Methods-and-Models-of-Intelligent-Control-Systems>**.**