МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Програмна інженерія та інформаційні технології управління»

Звіт з лабораторної роботи №2

з дисципліни «Основи управління складними системами»

Виконав:

Студент групи КН-36а

Кулик В.В.

Перевірив:

Голоскоков О. Є.

Харків – 2018

**Тема:** Вирішення оптимізаційної задачі за допомогою генетичних алгоритмів та “optimization tool”.

**Ціль:** Навчитися вирішувати оптимізаційні задачі.

**Хід виконання роботи:**

Задана функція:

y = (10\*x^2 + 3\*x^2+x+5)^2;

За допомогою оптимізаційної задачі ми знаходимо екстремум функції, а саме точку мінімуму.

1. Оптимізація за допомогою генетичного алгоритму.

Для того, щоб здійснити таку оптимізацію, потрібно перейти у розділ “apps” та вибрати пункт “optimization” , це показано на рисунку 1. Інтерфейс цього пункту показаний на рисунку 2. У ньому потрібно обрати пункт “ga” та увести необхідні дані, а саме функцію для оптимізації, та кількість змінних у ній.

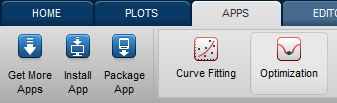


Рисунок 1 – Пункт оптимізації у розділі додатків

Далі потрібно натиснути кнопку старт і у розділі “Final point” з’явиться точка мінімума .

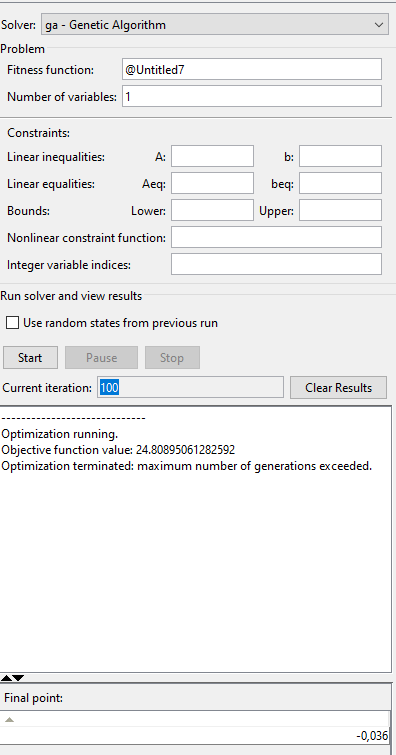


Рисунок 2 – Інтерфейс пункту оптимізації, та введені дані для початку розв’язку задачі оптимізації генетичним алгоритмом

Отримуємо наступні вихідні значення.

**Аргумент функції** x = -0,036

**Значення цільової функції**: 24,8089

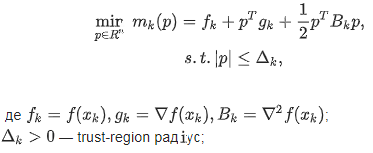
1. Оптимізація за допомогою методом нелінійної оптимізації довірчої області.

Цей алгоритм є одним з найбільш важливих чисельних методів оптимізації в рішенні проблем нелінійного програмування . Метод базується на визначенні регіону навколо кращого рішення, в якому квадратична модель апроксимує цільову функцію.

**Розглянемо даний алгоритм:**

**Крок №1**

Метод використовує квадратичну модель. На кожній ітерації крок обчислюється шляхом вирішення наступної квадратичної задачі (subproblem):



Таким чином trust-region вимагає послідовного обчислення аппроксимаций квадратичної моделі в яких цільова функція і умова (яке може бути записано ) теж квадратично. Коли гессіан () позитивно визначений і рішення легко визначити - це безумовний мінімум. В даному випадку називають повним кроком. Рішення не так очевидно в інших випадках, однак пошук його не варто занадто дорого. Нам необхідно знайти лише приблизне рішення, щоб отримати достатню збіжність і хорошу поведінку алгоритму на практиці.

**Крок №2**

Перша проблема, яка виникає при визначенні trust-region алгоритму - це вибір стратегії для пошуку оптимального trust-region радіусу на кожній ітерації. Вибір грунтується на схожості функції і цільової функції на попередній ітерації. Після знаходження ми визначаємо наступне співвідношення:



Знаменник завжди повинен бути невід’ємним, оскільки крок виходить шляхом мінімізації квадратичної моделі по регіону, який включає в себе крок. Відношення використовується для визначення наступності кроку і подальшого оновлення trust-region радіусу.

Якщо чи ми зменшуємо розмір trust-region області.

Якщо , тоді модель добре відповідає цільовій функції. В даному випадку слід розширити trust-region на наступній ітерації.

В інших випадках trust-region залишається незмінним.

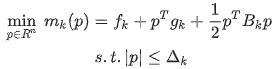
**Крок №3**

Наступний алгоритм описує процес:

Визначаємо стартову точку , максимальний trust-region радіус , початковий trust-region радіус і константу

for k = 0, 1, 2, ... поки не оптимум.

Вирішуємо:



де – рішення.

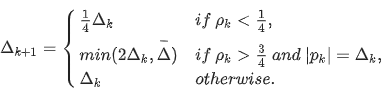
Вичислюємо відношення:



Оновлюємо поточну точку:



Оновлюємо trust-region радіус:



Для вирішення задачі оптимізації методом нелінійної оптимізації, потрібно обрати метод “fminunc” у розділі “solver”, та увести функцію, початкову точку і натиснути кнопку старту, введені дані та результат показані на рисунку 3.

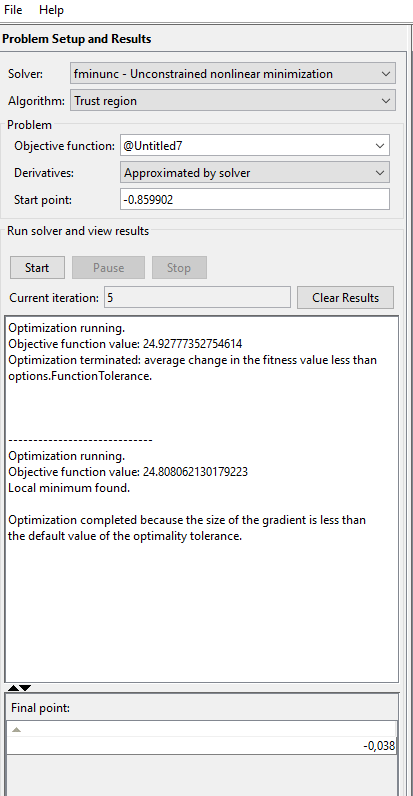


Рисунок 3 – результат роботи алгоритму метода нелінійної оптимізації

Отримуємо наступні вихідні значення.

**Аргумент функції** x = -0,038

**Значення цільової функції**: 24,9277

**Аналіз:**

Виконавши оптимізацію двома алгоритмами (генетичним та нелінійної оптимізації) можна помітити, що значення аргументів функції та значення цільової функції приблизно на рівні похибки однакові. Також видно, що алгоритм нелінійної оптимізації значно ефективніший за генетичний алгоритм (5 ітерацій проти 100).

**Висновки:**

У даній лабораторній роботі було вирішено оптимізаційну задачу за допомогою генетичного алгоритму, та методом нелінійної оптимізації довірчої області за допомогою додатку “optimization tool”.