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 «Інформаційні технології управління в умовах ризику»

на тему: «Вирішення оптимізаційної задачі за допомогою генетичних алгоритмів та tool-box optimization»

Виконав:

Студент групи КН-416а

Рубан Ю. Д.

Перевірив:

Голоскоков О. Є.

Харків – 2019

**Ціль:** Навчитися вирішувати оптимізаційні задачі.

**Хід виконання роботи:**

Задана функція:

y = 4\*(x1-5).^2 +(x2-6).^2;

Опис генетичного алгоритму.

Генетиичний алгоритм — це [еволюційний алгоритм](https://uk.wikipedia.org/wiki/%D0%95%D0%B2%D0%BE%D0%BB%D1%8E%D1%86%D1%96%D0%B9%D0%BD%D0%B8%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) пошуку, що використовується для вирішення задач оптимізації і моделювання шляхом послідовного підбору, комбінування і варіації шуканих параметрів з використанням механізмів, що нагадують [біологічну](https://uk.wikipedia.org/wiki/%D0%91%D1%96%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%8F) [еволюцію](https://uk.wikipedia.org/wiki/%D0%95%D0%B2%D0%BE%D0%BB%D1%8E%D1%86%D1%96%D1%8F).

Задача кодується таким чином, щоб її вирішення могло бути представлено в вигляді масиву подібного до інформації складу хромосоми. Цей масив часто називають саме так «хромосома». Випадковим чином в масиві створюється деяка кількість початкових елементів «осіб», або початкова популяція. Особи оцінюються з використанням функції пристосування, в результаті якої кожній особі присвоюється певне значення пристосованості, яке визначає можливість виживання особи. Після цього з використанням отриманих значень пристосованості вибираються особи, допущені до схрещення (*селекція*). До осіб застосовується «генетичні оператори» (в більшості випадків це оператор схрещення (crossover) і оператор мутації (mutation)), створюючи таким чином наступне покоління осіб. Особи наступного покоління також оцінюються застосуванням генетичних операторів і виконується селекція і мутація. Так моделюється еволюційний процес, що продовжується декілька життєвих циклів (*поколінь*), поки не буде виконано критерій зупинки алгоритму. Таким критерієм може бути:

* знаходження глобального, або надоптимального вирішення;
* вичерпання числа поколінь, що відпущені на еволюцію;
* вичерпання часу, відпущеного на еволюцію.

Генетичні алгоритми можуть використати для пошуку рішень в дуже великих і важких просторах пошуку.

Етапи генетичного алгоритму:

1. Створення початкової популяції:
2. Обчислення функції пристосованості для осіб популяції (оцінювання)
3. Повторювання до виконання критерію зупинки алгоритму:
   1. Вибір індивідів із поточної популяції (селекція)
   2. Схрещення або/та мутація
   3. Обчислення функції пристосовуваності для всіх осіб
   4. Формування нового покоління

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.

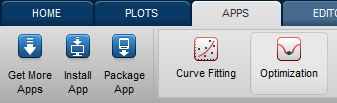
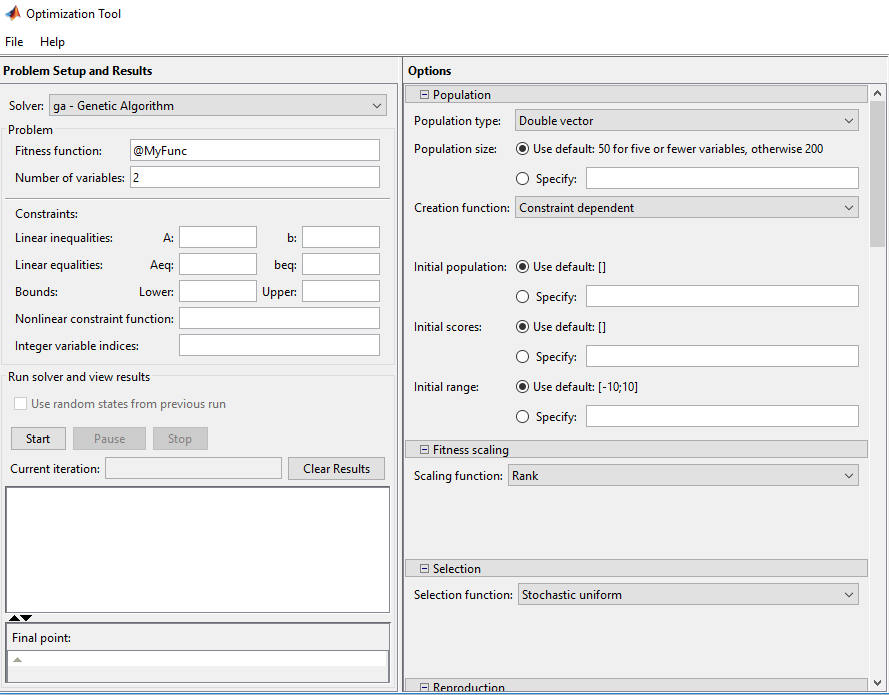


Рисунок 1 – Пункт оптимізації у розділі додатків

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

Інтерфейс Optimization Tool поділяється на 4 розділи:

1. Вибір методу оптимізації та необхідні початкові значення
2. Розділ обмежень у який вводиться матриця коефіцієнтів А, вектор обмежень b. Linear inequalities задає обмеження виду Ax < b, Linear equalities задає обмеження виду Ax = b. Lower та Upper bound задають границі в яких знаходиться вектор х. Nonlinear constraint function задає нелінійну обмежувальну функцію.
3. Вікно результатів у якому відображаються деякі відомості про процес оптимізації та результати обчислень.
4. Вікно опції у якому можна налаштувати обраний метод.

Для генетичного алгоритму список опцій показаний на рисунку 3.

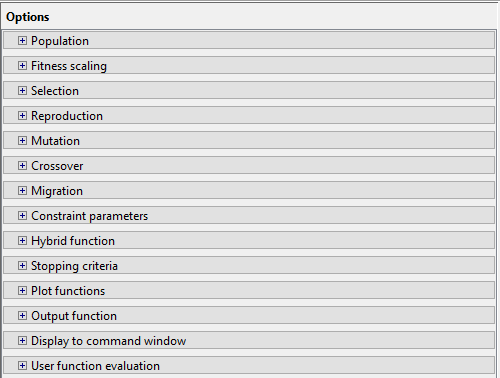


Рисунок 3 – опції для генетичного алгоритму

1) популяція (вкладка Population);

2) оператор відбору (вкладка Selection);

3) оператор репродукції (вкладка Reproduction);

4) оператор мутації (вкладка Mutation);

5) оператор схрещування (вкладка Crossover);

6) перенесення особин між популяціями (вкладка Migration);

7) багатокритеріальні спеціальні параметри (вкладка Multiobjective problem settings);

8) завдання гібридної функції (вкладка Hybrid function);

9) завдання критерію зупинки алгоритму (вкладка Stopping criteria);

10) висновок різної додаткової інформації по ходу роботи генетичного алгоритму (вкладка Plot Functions);

11) висновок результатів роботи алгоритму у вигляді нової функції (вкладка Output function);

12) задання набору інформації для виведення в командне вікно (вкладка Display to command window);

13) спосіб обчислення значень оптимізуючої та обмежувальної функцій (вкладка User function evaluation).

Для того, щоб знайти точку мінімума для заданої функції потрібно натиснути на кнопку Start, дочекатися завершення разрахунків і отримати результат. Результат показаний на рисунку 4.

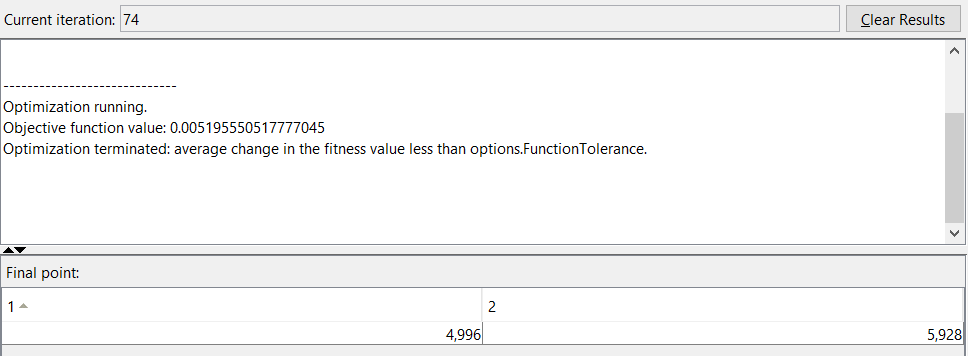


Рисунок 4 – Рішення задачі оптимізації методом генетичного алгоритму.

Іншим способом мінімум функції можна знайти за допомогою вирішувача “fminunc” для якого потрібно указати функцію для якої потрібно знайти мінімум, обрати метод знаходження похідних та указати початкову точку.

Для вирішення задачі оптимізації методом нелінійної оптимізації, потрібно натиснути кнопку старту. Введені початкові дані та результат показані на рисунку 5.

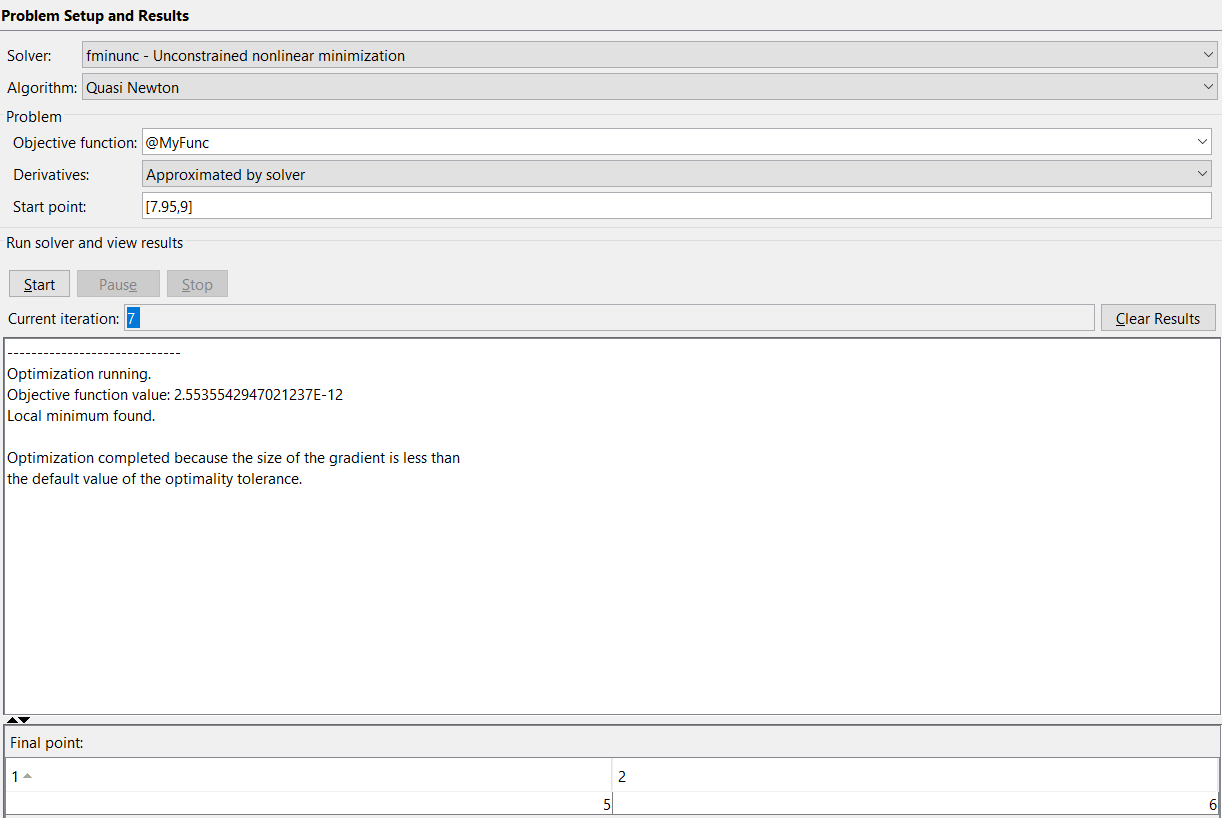


Рисунок 5 – результат роботи алгоритму метода нелінійної оптимізації

Вирішувач “fminunc” має ряд деяких опції, які показані на рисунку 6.

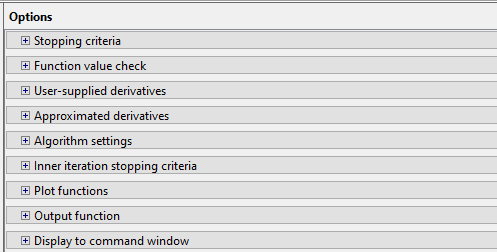


Рисунок 6 – опції для fminfunc

1. Критерій зупинки алгоритму
2. Перевірка на некоректні дані, такі як NaN, inf
3. Інструменти знаходження похідних задані користувачем
4. Спосіб пошуку похідних
5. Опції алгоритму
6. Критерії зупинки внутрішнього циклу

7) Висновок різної додаткової інформації по ходу роботи генетичного алгоритму

8) Висновок результатів роботи алгоритму у вигляді нової функції

9) Задання набору інформації для виведення в командне вікно

Таблиця порівняння результатів показана у таблиці 1.1.

Таблиця 1.1. – Порівняння роботи генетичного алгоритму та солверу fminfunc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Алгоритм |  |  |  | Кількість ітерацій |
| Генетичний алгоритм | 4.996 | 5.928 | 0.0051 | 74 |
| Алгоритм мінімізації методом Quasi Newton | 5 | 6 | 0 | 7 |

З таблиці 1.1 видно, що для даної задачі генетичний алгоритм показує результаті гірші, аніж звичайний метод оптимізації.

**Висновки:**

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