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»

Виконала:

Студентка групи КН-36а

Сушко В. В.

Перевірив:

Голоскоков О. Є.

Харків – 2019

**Тема:** Рішення оптимізаційної задачі управління методами генетичних алгоритмів (GA)

**Мета:** Вирішити оптимізаційну задачу управління методами Genetic Algorithm.

**Хід роботи**

1. **Генетичний алгоритм**

Генетичний алгоритм – це метод розв’язання задач оптимізації на основі природного добору, аналогічно тому, як це відбувається в процесі біологічної еволюції. У генетичному алгоритмі відбувається багаторазова модифікація сімейства окремих розв’язків. На кожному кроці проводиться відбір вибраних навмання суб'єктів з отриманого поточного розв’язання, що називається батьківським і яке використовується для генерації наступного дочірнього покоління. За допомогою послідовного відбору поколінь відбувається "еволюція" у напрямку до оптимального розв’язання. Генетичний алгоритм можна застосовувати до різноманітних задач оптимізації, які недостатньо чітко вписуються в стандартні оптимізаційні алгоритми. На кожному кроці для генерації наступного покоління в генетичному алгоритмі використовуються наступні три основні правила:

− правило відбору – відбирає об'єкти, що називаються батьківськими та які становлять основу наступного покоління з поточного розв’язання;

− правило схрещування, за яким відбувається вибір із двох батьківських об'єктів дочірнього для наступного покоління;

− правило мутації, за яким на основі ймовірнісних алгоритмів з батьківських об'єктів формуються дочірні.

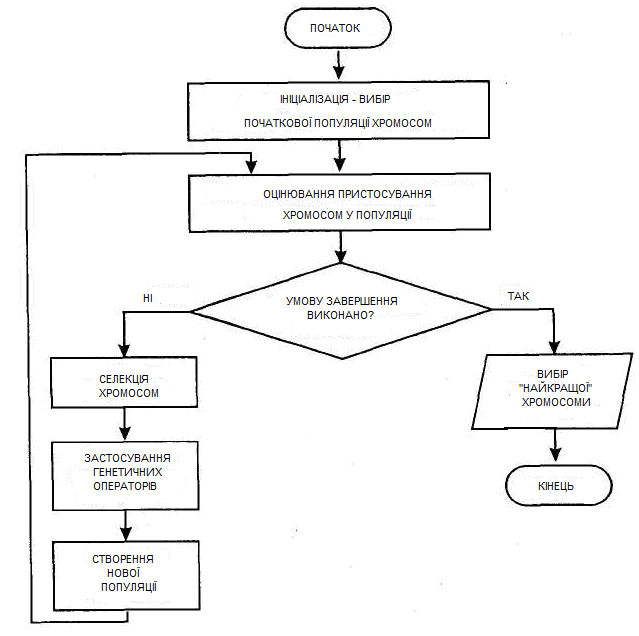


Рис. 1. Блок-схема генетичного алгоритму

Генетичний алгоритм відрізняється від стандартних методів оптимізації наступними особливостями (табл. 1):

Таблиця 1 – Порівняння стандартного й генетичного алгоритмів

|  |  |
| --- | --- |
| Стандартний алгоритм | Генетичний алгоритм |
| На кожній ітерації генерується одна єдина точка. Отримана послідовність точок сходиться до оптимального рішення. | На кожній ітерації генерується сімейство точок. Отримане сімейство сходиться до оптимального розв’язання. |
| Вибір наступної точки здійснюється на основі детермінованих розрахунків. | Вибір наступного сімейства точок здійснюється на основі стохастичних розрахунків. |

Для запуску пакету Genetic Algorithm Tool треба у командному рядку MATLAB виконати команду gatool. Після цього запуститься пакет генетичних алгоритмів і на екрані з’явиться основне вікно утиліти (рис. 2).

У полі Fitness function вказується функція, що оптимізується, у вигляді @fitnessfun, де fitnessfun.m – назва M-файлу, в якому заздалегідь слід описати функцію, що оптимізується. Для виконання генетичного алгоритму слід клікнути мишкою на кнопку Start. Далі в панелі Status and Results здійснюється відображення результатів оптимізації.

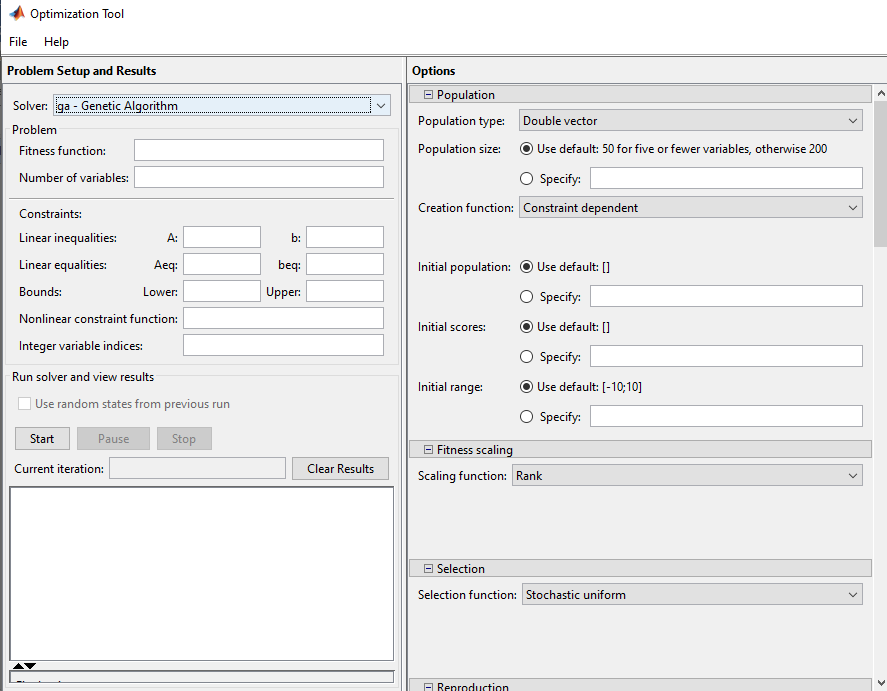


Рис.2 **–** Запуск пакету Genetic Algorithm Tool

1. **Оптимізація задачі за допомогою стандартного методу MATLAB.**
2. Задаємо тестову функцію на прикладі задання безумовного оптимуму.

y = **x4+y4-2x2+4xy-2y2+1**

Координати початкової точки пошуку [1.0, -1.0].

1. Для початку побудуємо графік функції від двох змінних - для цього створимо новий скрипт і пропишемо там цей код:

[x y] = meshgrid(-2:0.1:2, -2:0.1:2);

z = x.^4 + y.^4 - 2\*x.^2 + 4 \* x.\*y - 2\*y.^2 + 1;

surf(x,y,z);

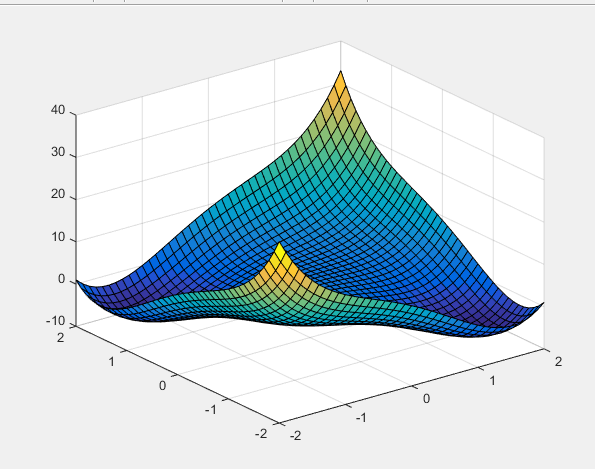


Рис.3 **–** Графік функції від двох змінних

1. Як видно з графіку, є дві ділянки, де присутній локальний мінімум (темно-сині ділянки), і наше завдання знайти координати і значення двох цих точок. Скористаємося стандартними інструментами Matlab і створимо новий скрипт з ім'ям second.m, в якому і пропишемо код:

function fun=second(x)

fun = x(1)^4 + x(2)^4 - 2\*x(1)^2 + 4 \* x(1)\*x(2) - 2\*x(2)^2 + 1;

end

1. Після цього, в командному рядку, як і для першого завдання, прописуємо стандартну функцію Matlab:

>> [z,f,exitflag,output] = fminsearch(@second, 1.0,-1.0, optimset('TolX',1e-5))

1. Отримуємо результат:

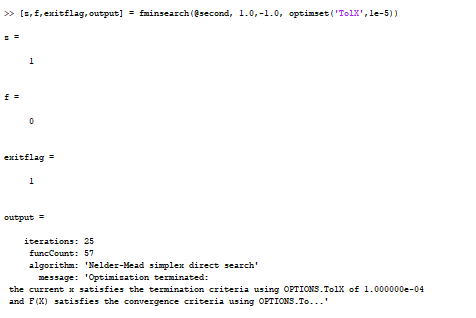


Рис.4 **–** Результат оптимізації звичайним методом

1. **Оптимізація задачі за допомогою Genetic Algorithm MATLAB.**
2. Задаємо тестову функцію на прикладі задання безумовного оптимуму.

y = **x4+y4-2x2+4xy-2y2+1**

Координати початкової точки пошуку [1.0, -1.0].

1. Для запуску пакету Genetic Algorithm Tool треба у командному рядку MATLAB виконати команду gatool. Заповнюємо наступні поля: Fitness function, Number of variables, Hibrid function.

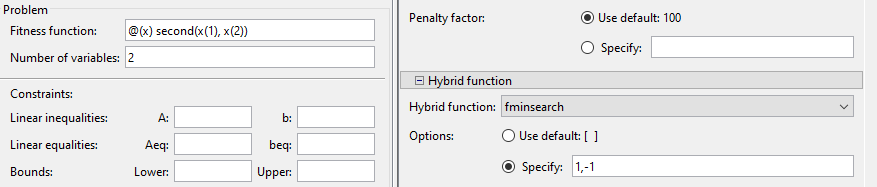


Рис.5 **–** Заповнені поля для оптимізації задачі за допомогою генетичного алгоритму

1. Отримуємо результат:

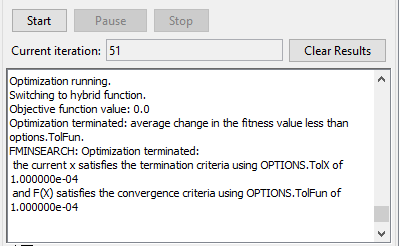


Рис.6 **–** Результат оптимізації за допомогою генетичного алгоритму

1. **Порівняння отриманих результатів за ступенем точності обчислення.**

Таблиця 2 – Порівняння стандартного й генетичного алгоритмів

|  |  |  |
| --- | --- | --- |
|  | Стандартний алгоритм | Генетичний алгоритм |
| Кількість ітерацій | 25 | 51(завжди різна) |
| Значення цільової функції | 0 | 0 |
| Зупинка оптимізації | Середня зміна значення менша, ніж опції.TolFun.  Поточний х задовольняє критеріям припинення за допомогою OPTIONS.TolX 1.000000e-04  F (X) задовольняє критеріям конвергенції за допомогою OPTIONS.TolFun 1000000e-04 | Середня зміна значення менша, ніж опції.TolFun.  Поточний х задовольняє критеріям припинення за допомогою OPTIONS.TolX 1.000000e-04  F (X) задовольняє критеріям конвергенції за допомогою OPTIONS.TolFun 1000000e-04 |

**Висновок:** на цій лабораторній роботі було досліджено програмний пакет MATLAB, вирішено оптимізаційну задачу управління за допомогою Genetic Algorithm та стандартного методу MATLAB. Порівнюючи результати обох алгоритмів, можна зробити висновок, що результати оптимізації співпадають. Але кількість ітерацій відрізняються, до того ж, використовуючи генетичний алгоритм, кількість ітерацій завжди різна.