**Операторний алгоритм у просторі .**

Задача пошуку мінімуму функції  заданої у прямокутній області , з заданою величиною похибки .

1. *Формування початкової множини розв’язків.*

Нехай ,  вершини прямокутника, тоді іншу пару вершин отримаємо наступним чином:

Оператор .

.

Звідси маємо , .

1. *Побудова нової області пошуку (першої популяції векторів-потомків).*

Нехай  ( - пара батьківських хромосом). Матричні оператори  мають вигляд.



,

де , деякі **випадкові** числа з інтервалу (0,1).

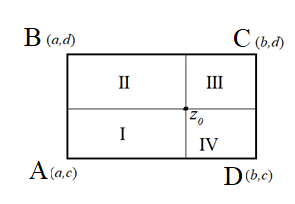


Дію цих операторів на пару векторів-хромосом з області пошуку називаємо узагальненим кросовером.

Область пошуку є прямокутник ABCD (Рис.1), вершини якого розіб’ємо на пари протилежних вершин,  та , . Ці чотири точки утворюють початкову популяцію векторів хромосом, , яка складається всього з двох векторів. Застосуємо до них оператори узагальненого кросоверу. Отримаємо пари векторів-потомків:

 , ,

 , .

Компоненти отриманих векторів є точками області пошуку, в яких обчислюються значення фітнес-функції ,, . Позначимо  точку області пошуку, для якої функція має найменше з отриманих значень, тобто: Рис.1



Точка  розбиває область пошуку на чотири прямокутники, для яких вона буде однією з вершин (Рис.1). Протилежними до неї вершинами будуть вершини початкової області пошуку.

Для кожного з прямокутників I, II, III, IV застосуємо попередню процедуру. Наприклад, для прямокутника І, оператори застосовуються до векторів , , в результаті чого отримаємо точку , де фітнес-функція приймає мінімальне значення. Аналогічно визначаємо точки .

Таким чином отримано п’ять точок ,, (Рис.2).

Изображение выглядит как стол

Автоматически созданное описаниеСеред цих точок вибираємо дві кращих, тобто ті для яких значення функції є найменшими. Припустимо, що це будуть наступні точки  і , для яких має місце нерівність . Тоді точка  буде вершиною нового прямокутника Рис.2

(позначимо її ), а точка  його центром.

Розглянемо пару точок  та . Будуємо вектор . Застосовуючи до нього оператор  при , отримаємо вектор

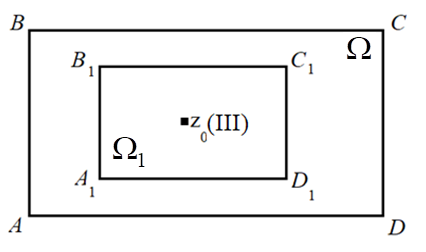
.

Якщо , то розглянемо пару точок  і . Для цієї пари точок має місце рівність .

Тоді точка  є найкращим наближенням розв’язку на даному ітераційному кроці. В даному випадку вершина прямокутника протилежна до точки  визначається за формулою.

Якщо , то розглянемо пару точок  і . Для цієї пари точок має місце рівність  Тоді точка  є найкращим наближенням розв’язку на даному ітераційному кроці. В даному випадку вершина прямокутника протилежна до точки  визначається за формулою.

Таким чином отримано дві вершини нової області пошуку . Зрозуміло, що після застосування оператора  інші дві вершини області пошуку матимуть вигляд: .

Таким чином, отримуємо новий прямокутник, який міститься всередині початкового (Рис. 3). Отриманий прямокутник утворює нову область пошуку, для якої повторюється наведена вище процедура. Рис.3

Необхідно зауважити, що починаючи з третього кроку нова область пошуку (новий прямокутник) може не міститися у попередній, але їх перетин завжди буде непорожнім.

Продовжуючи цей процес, отримаємо послідовність прямокутників  з центрами . Послідовність точок  збігається до шуканого розв’язку. Тобто для довільної точності обчислень (похибки ) матимемо починаючи з певного номеру .

1. Узагальнена мутація.

Розглянемо випадок, коли і при цьому прямокутник не вироджується у відрізок прямої або точку. Застосовуємо процедуру «ігри хаосу». В цьому випадку це процес побудови «килима» Серпінського.

Спочатку побудуємо прямокутник, для якого будуватиметься «килим» Серпінського. Застосовуючи оператор  при  до вектора побудуємо прямокутник , . Його вершини  знайдемо наступним чином:





.

Відрізок  в три рази довший ніж ,тобто .

Як відомо, вершини прямокутника  і середини його сторін є аттракторами.

Координати середніх точок  сторін прямокутника отримуємо наступним чином:



.

Вибравши точку  і випадковим чином обираємо один з аттракторів ,, наприклад . Розглянемо вектор . Подіємо на цей вектор оператором , де , отримаємо дві точки *М* та,

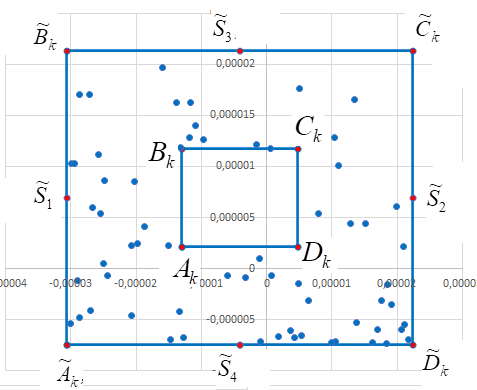
.

Саме точка  не належить прямокутнику . Продовжуючи процес, обираєм випадковим чином один з аттракторів наприклад , отримаємо дві точки

,

з яких  не належить .

Продовжуючи цей випадковий процес, отримаємо множину точок, що розташовані в середині прямокутника , але не належать прямокутнику  (Рис.4). Тобто відбувається процес розсіювання точок в околі прямокутника , що дає змогу контролювати поведінку послідовності наближень  шуканого розв’язку, а також попадання її в окіл хибного екстремуму (якщо

функція має крім глобального мінімуму певну кількість локальних). Кількість таких точок *m* визначається, в залежності від складності функції . Процес «ігор хаосу» можна запускати кілька разів, доки не з’явиться точка така, що . Якщо після певної кількості запусків Рис. 4

процесу «ігри хаосу» точка  не з’явилася, то це означає що збільшення значення функції було випадковим . Після чого ітераційний пошук розв’язку продовжується згідно п.2.

б) Якщо виникає ситуація злипання прямокутника у відрізок або в точку, то необхідно повернутись до попереднього ітераційного кроку і застосувати описану в п.а) процедуру узагальненої мутації.

Розглянемо детально алгоритм для однієї з тестових функцій, а саме для сферичної функції або функції Де Йонга 1.

1. Сферична функція або функція Де Йонга 1(Sphere model, De Jong’s function 1)*:* .

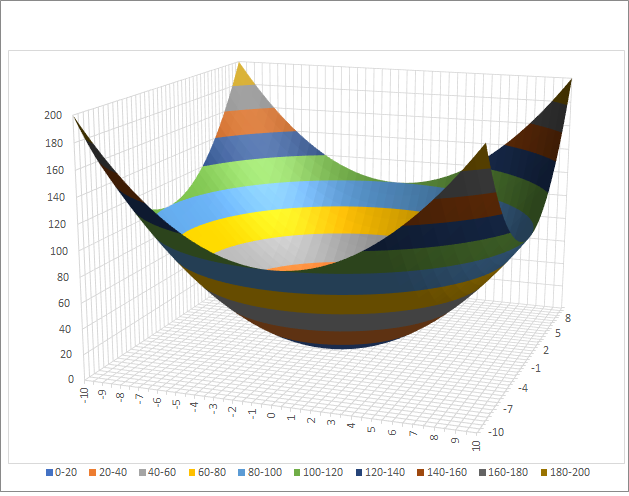


Рис.5

Для *n*=2: . (Рис.5)

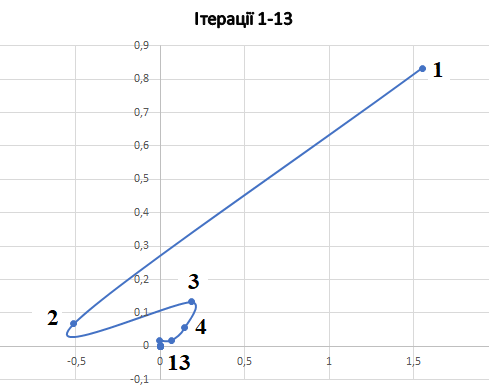
(похибка для ) , нав’язка для послідовності наближень ).

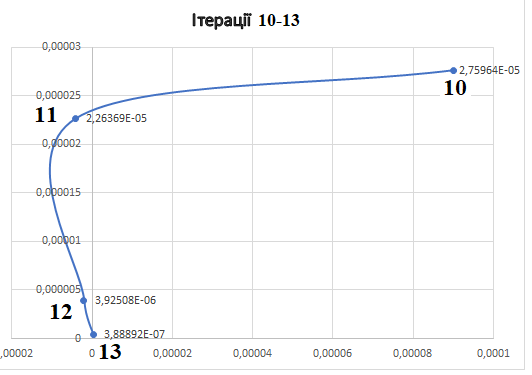
Таблиця 2

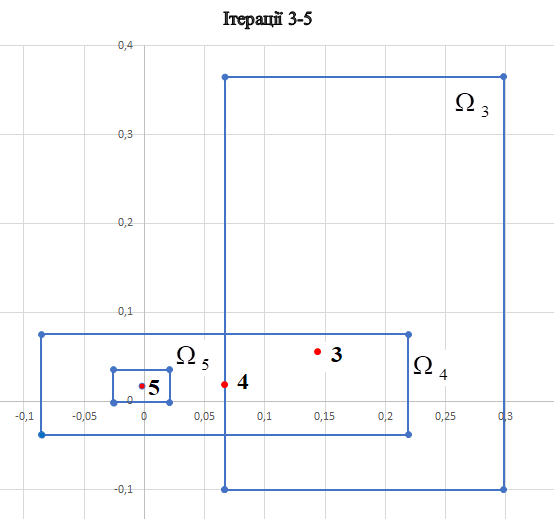
Результат роботи алгоритму для функції Де Йонга

(програмна реалізація Excel)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № ітер. | Значення  *F*(*x,y*) | Значення  *х* | Значення  y | Кільк. ітер. | Кільк. мут. | Випадкові коефіцієнти операторів кросоверу | | | |
|  |  |  |  |
| 1 | 2,223E-13 | 2,666E-07 | 3,888E-07 | 13 | 0 | 0,35 | 0,68 | 0,03 | 0,04 |
| 2 | 1,292E-14 | -4,89E-08 | -1,81E-07 | 14 | 0 | 0,46 | 0,58 | 0,23 | 0,14 |
| 3 | 4,475E-14 | 1,57E-07 | 1,41E-07 | 13 | 1 | 0,87 | 0,31 | 0,96 | 0,85 |
| 4 | 3,791E-13 | 5,221E-07 | 3,262E-07 | 19 | 0 | 0,93 | 0,85 | 0,37 | 0,72 |
| 5 | 8,370E-11 | 8,493E-06 | 3,399E-06 | 16 | 0 | 0,69 | 0,62 | 0,67 | 0,42 |
| Середнє | 1,68E-11 | 1,878E-06 | 8,150E-07 | 15 | 0 |  |  |  |  |







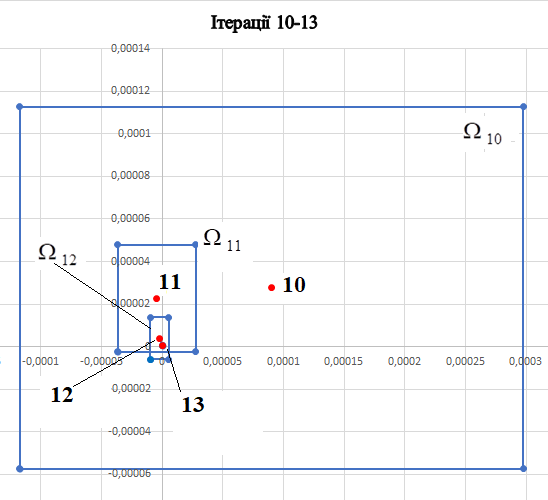
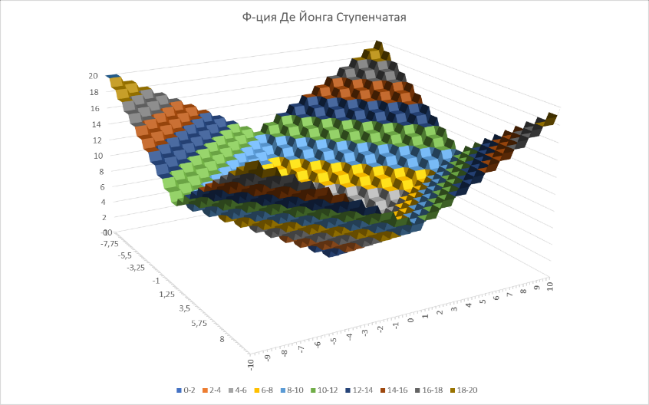


Рис.6

1. Сходинкова функція Де Йонга (Step function (De Jong)):

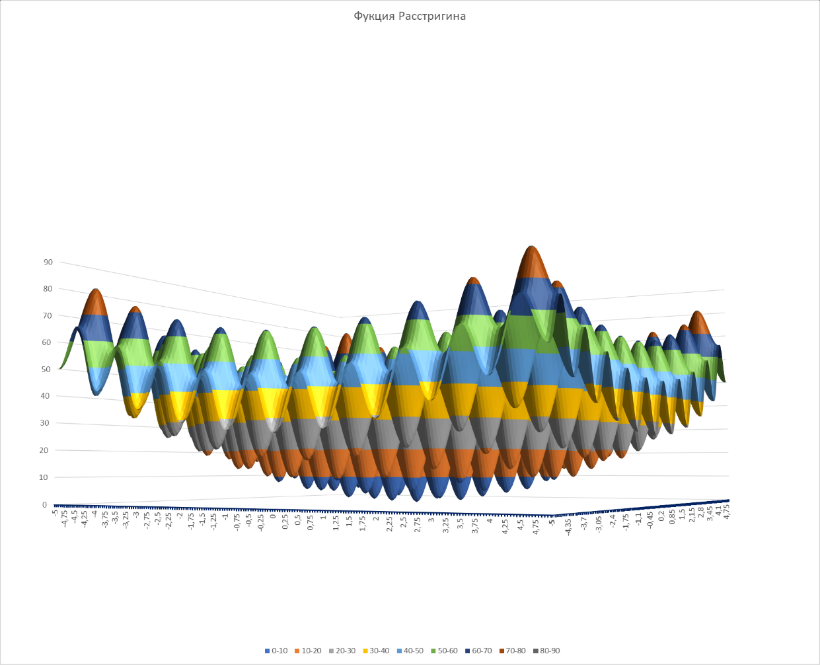
.

Глобальний мінімум .



1. Функція Растрігіна (Rastrigin’s function):

.

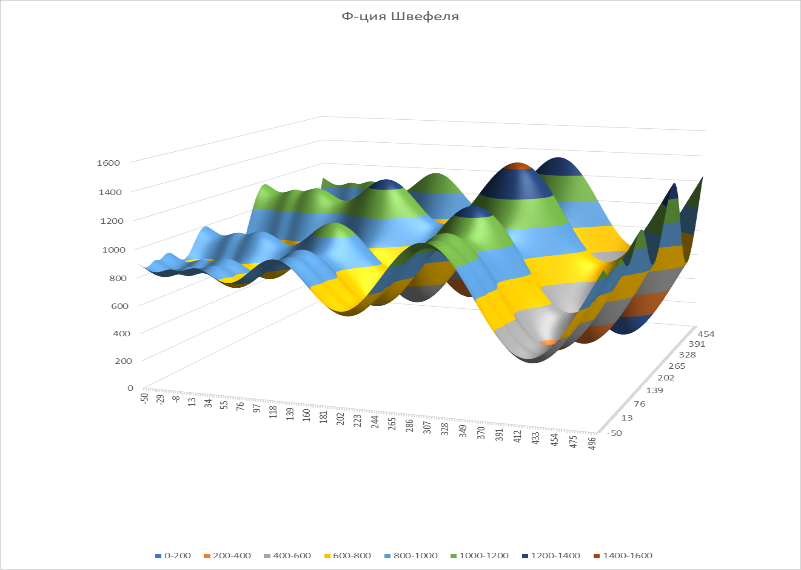
Глобальний мінімум . Локальних мінімумів злічена кількість.

4. Функція Швефеля (Schwefel’s (Sine root) function)

.



Глобальний мінімум .



Блок-схема операторного алгоритму