|  |  |
| --- | --- |
|  | Московский Государственный Технический Университет им. Н.Э.Баумана  Факультет ИУ «Информатика и системы управления»  Кафедра ИУ8 «Информационная Безопасность» |

Отчёт по лабораторной работе №2

«Исследование метода случайного поиска экстремума функции одного переменного»

по дисциплине

«Теория систем и системный анализ»

Выполнил: Т.А. Орбелян

Проверил: И.С. Строганов

Группа: ИУ8-11М

Вариант: 6

Оценка:

Москва 2021

**Цель работы:**

Изучение метода случайного поиска экстремума на примере унимодальной и мультимодальной функций одного переменного.

**Постановка задачи:**

1. На интервале [-3.5, 3.5] задана унимодальная функция одного переменного f(x) = (1 - x)­­2 + ex. Используя метод случайного поиска, осуществить поиск минимума f(x) с заданной вероятностью попадания в окрестность экстремума P при допустимой длине интервала неопределенности ε . Определить необходимое число испытаний N. Численный эксперимент выполнить для значений P = 0.90, 0.91, ..., 0.99 и значений ε = (b - a) q, где q = 0.005, 0.010, ..., 0.100.
2. При аналогичных исходных условиях осуществить поиск минимума f(x), модулированной сигналом sin5x, т.е. мультимодальной функции.

**Практическая часть:**

Программы написаны на языке Python версии 3.8 в интегрированной среде разработки Pycharm версии 2021.2 (Professional Edition). Файлы доступны для скачивания: <https://github.com/RedCactusInn/IU8/tree/master/TSiSA/lab_2/src>

В файле function.py, который находится в директории “../..”, реализованы унимодальная и мультимодальная функции (рисунок 1).

import math  
  
  
def function(x: float) -> float:  
 function\_value = (1 - x) \* (1 - x) + math.exp(x)  
 return function\_value  
  
  
def function\_multimodal(x: float) -> float:  
 return function(x) \* math.sin(5 \* x)

Рисунок 1. Содержимое файла function.py

В файле random\_search.py реализован алгоритм случайного поиска (рисунок 2)

import random  
import math  
  
  
def get\_number\_of\_estimations(p, q) -> int:  
 number\_of\_estimations = math.ceil(math.log(1 - p) / math.log(1 - q))  
 return number\_of\_estimations  
  
  
def random\_search(func, q, p, a\_border, b\_border) -> tuple:  
 a\_border, b\_border = (a\_border, b\_border) if b\_border > a\_border else (b\_border, a\_border)  
 area = b\_border - a\_border  
 number\_of\_estimations = get\_number\_of\_estimations(p, q)  
 dots = []  
 for estimation in range(number\_of\_estimations):  
 x = (random.random()) \* area + a\_border  
 y = func(x)  
 dots.append((x, y))  
 min\_dot = min(dots, key=lambda elem: elem[1])  
 return min\_dot, dots

Рисунок 2. Содержимое файла random\_search.py

В файле lab2.py (рисунок 3) реализовано сравнение работы алгоритма случайного поиска при разных значениях q и P для мультимодальной и для унимодальной функции.

import random\_search  
import function  
import numpy  
import matplotlib.pyplot as plt  
  
  
def plot\_dots(dots):  
 plt.scatter(\*zip(\*dots))  
 plt.show()  
  
  
function\_unimodal = function.function  
function\_multimodal = function.function\_multimodal  
functions = (function\_unimodal, function\_multimodal)  
a\_border = -3.5  
b\_border = 3.5  
  
  
def get\_template\_array(shape):  
 return numpy.zeros(shape=shape)  
  
  
def main():  
 q\_from, q\_to, q\_step = (0.005, 0.101, 0.005)  
 q\_values = numpy.arange(q\_from, q\_to, q\_step)  
 p\_from, p\_to, p\_step = (0.90, 0.991, 0.01)  
 p\_values = numpy.arange(p\_from, p\_to, p\_step)  
  
 shape = (len(q\_values), len(p\_values))  
 x\_min\_unimodal = get\_template\_array(shape)  
 f\_min\_unimodal = get\_template\_array(shape)  
 number\_of\_estimations\_unimodal = get\_template\_array(shape)  
 x\_min\_multimodal = get\_template\_array(shape)  
 f\_min\_multimodal = get\_template\_array(shape)  
 number\_of\_estimations\_multimodal = get\_template\_array(shape)  
  
 dots\_total\_unimodal = []  
 dots\_total\_multimodal = []  
  
 for i in range(len(q\_values)):  
 q = q\_values[i]  
 for j in range(len(p\_values)):  
 p = p\_values[j]  
  
 min\_dot\_unimodal, dots\_unimodal = random\_search.\  
 random\_search(function\_unimodal, q, p, a\_border, b\_border)  
 x\_min\_unimodal[i][j] = min\_dot\_unimodal[0]  
 f\_min\_unimodal[i][j] = min\_dot\_unimodal[1]  
 number\_of\_estimations\_unimodal[i][j] = len(dots\_unimodal)  
 dots\_total\_unimodal.extend(dots\_unimodal)  
  
 min\_dot\_multimodal, dots\_multimodal = random\_search.\  
 random\_search(function\_multimodal, q, p, a\_border, b\_border)  
 x\_min\_multimodal[i][j] = min\_dot\_multimodal[0]  
 f\_min\_multimodal[i][j] = min\_dot\_multimodal[1]  
 number\_of\_estimations\_multimodal[i][j] = len(dots\_multimodal)  
 dots\_total\_multimodal.extend(dots\_multimodal)  
  
 plot\_dots(dots\_total\_multimodal)  
 plot\_dots(dots\_total\_unimodal)  
 return  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Рисунок 3. Содержимое файла lab2.py

Скрипт lab2.py запущен в режиме отладки, точка останова установлена на 61 строке (return). Полученные графики функций изображены на рисунках 4 и 5.

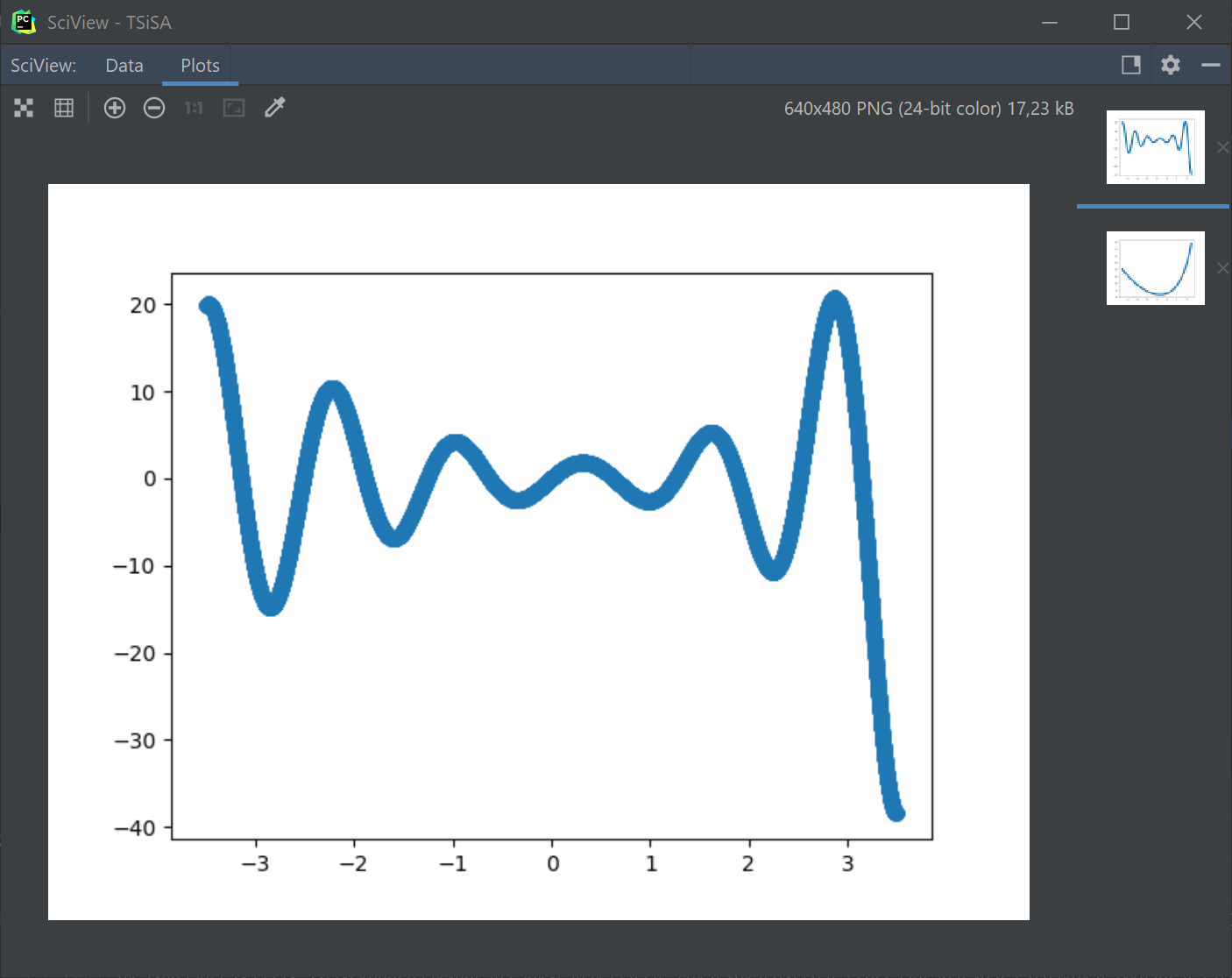


Рисунок 4. График мультимодальной функции.

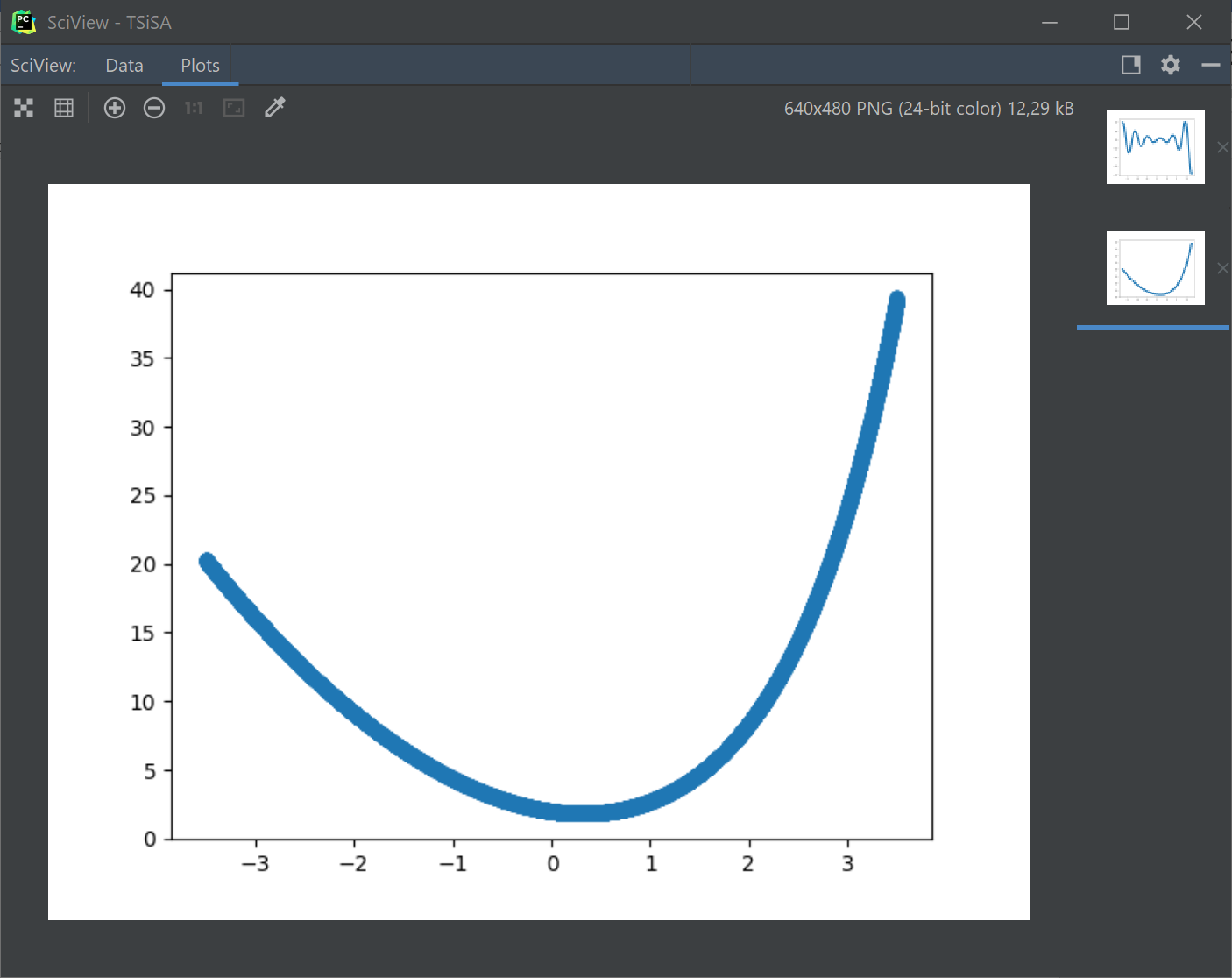


Рисунок 5. График унимодальной функции.

Результаты сравнения представлены в виде матриц, где по горизонтальной оси значения P (рисунок 6), а по вертикальной – значения q (рисунок 7).

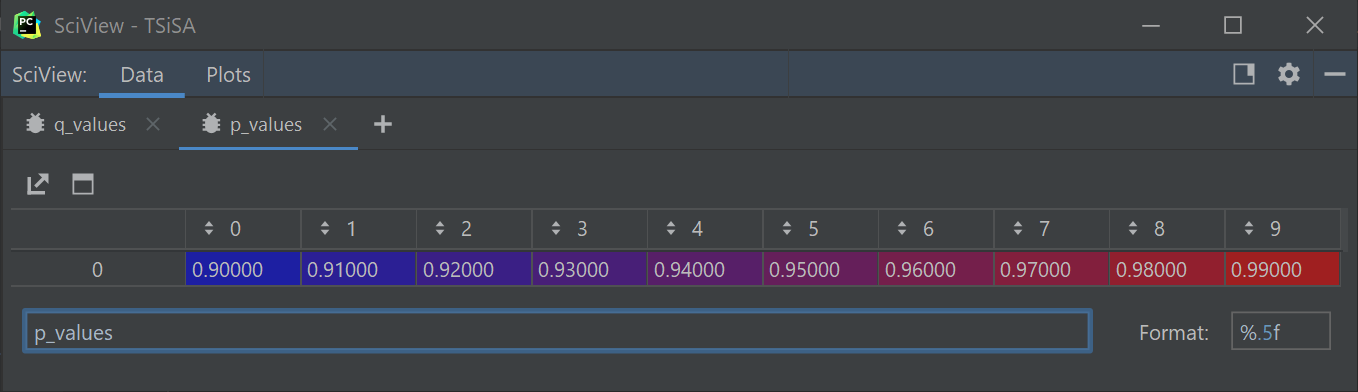


Рисунок 6. Значения P.

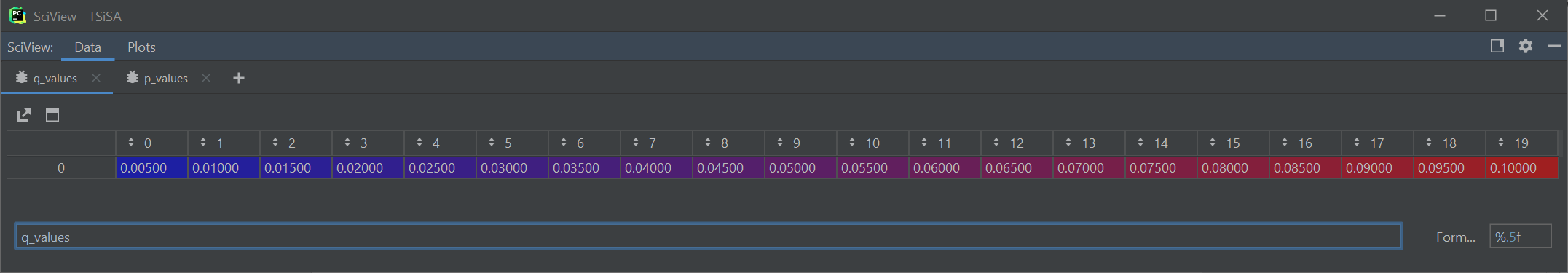


Рисунок 7. Значения q.

Полученные матрицы со значениями найденных минимумов (f\_min\_multimodal), координат найденных точек минимума (x\_min\_multimodal) и количества точек измерения (number\_of\_estimations\_multimodal) для мультимодальной функции (рисунок 4) изображены на рисунках 8, 9 и 10 соответственно.

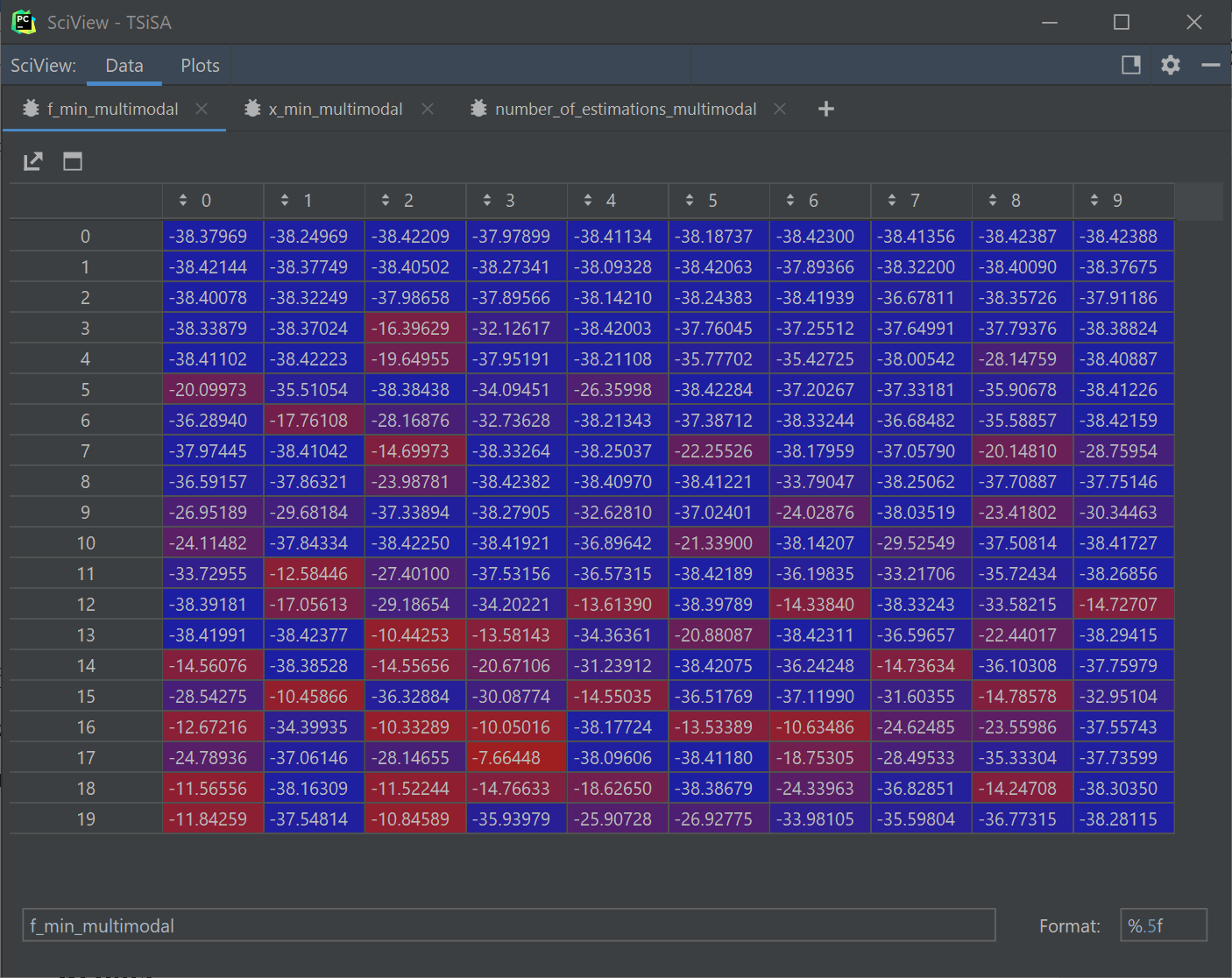


Рисунок 8. Матрица со значениями найденных минимумов.

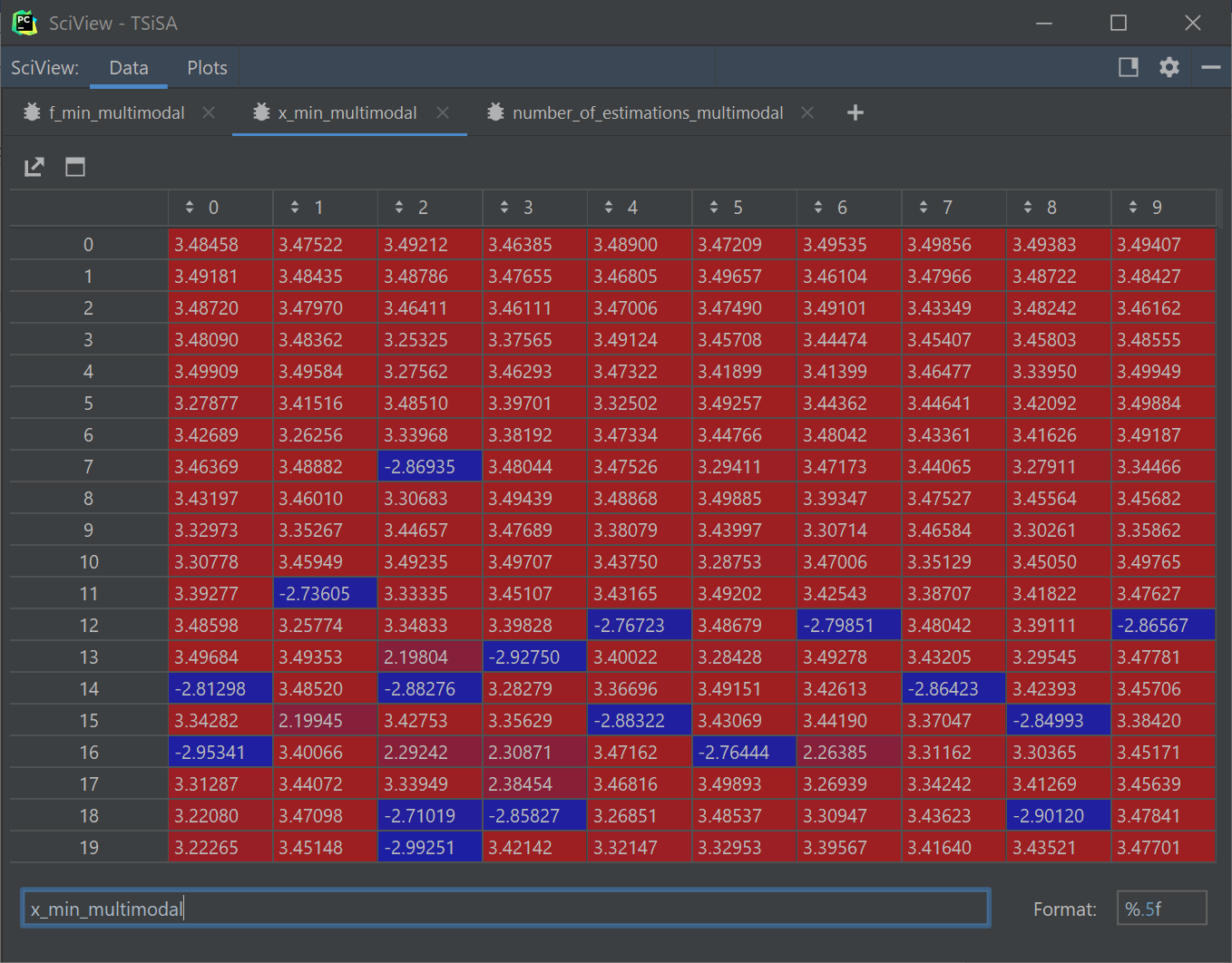


Рисунок 9. Матрица со значениями координат найденных минимумов.

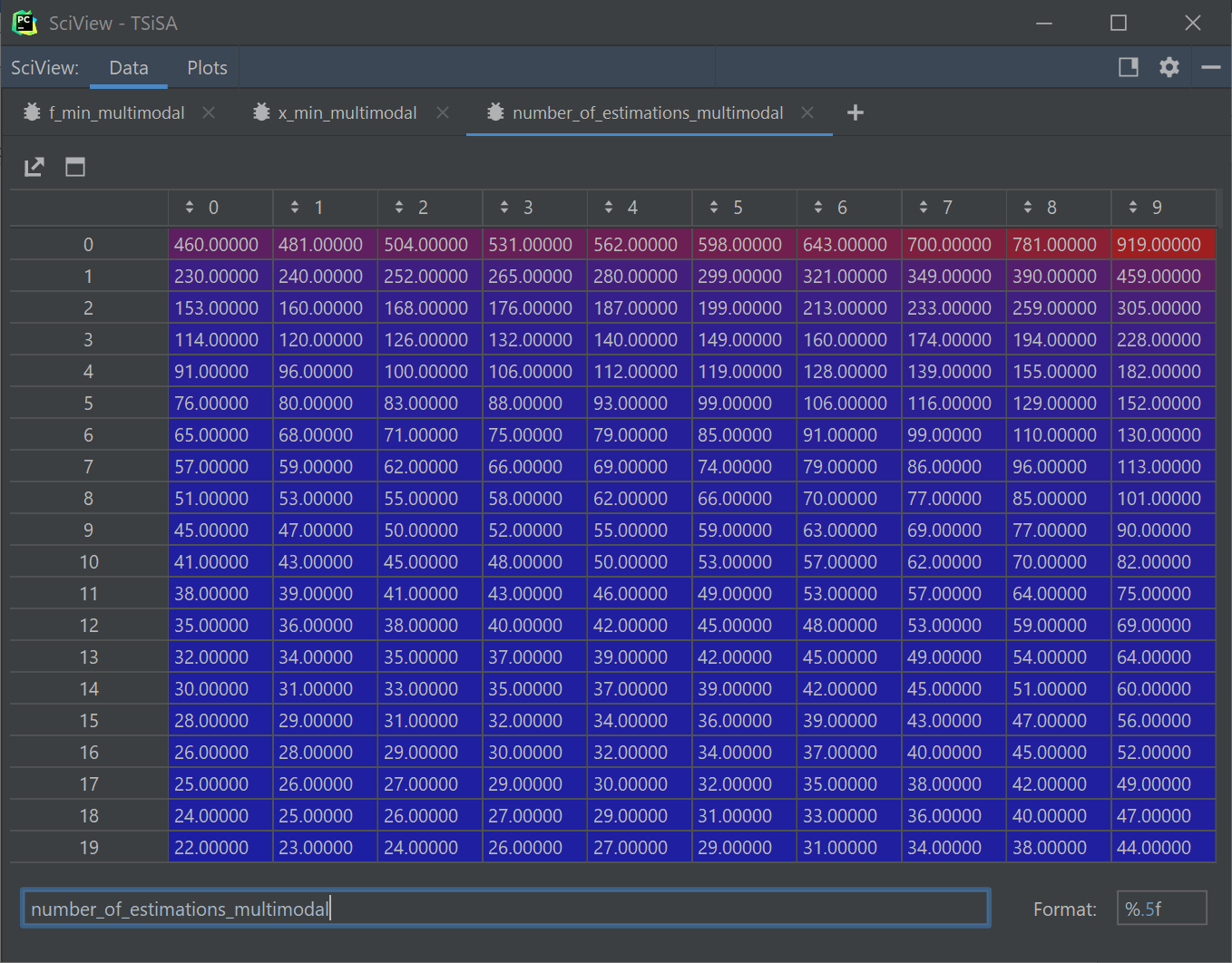


Рисунок 10. Матрица со значениями количества измерений.

Полученные матрицы со значениями найденных минимумов (f\_min\_unimodal), координат найденных точек минимума (x\_min\_unimodal) и количества точек измерения (number\_of\_estimations\_unimodal) для унимодальной функции (рисунок 5) изображены на рисунках 11, 12 и 13 соответственно.

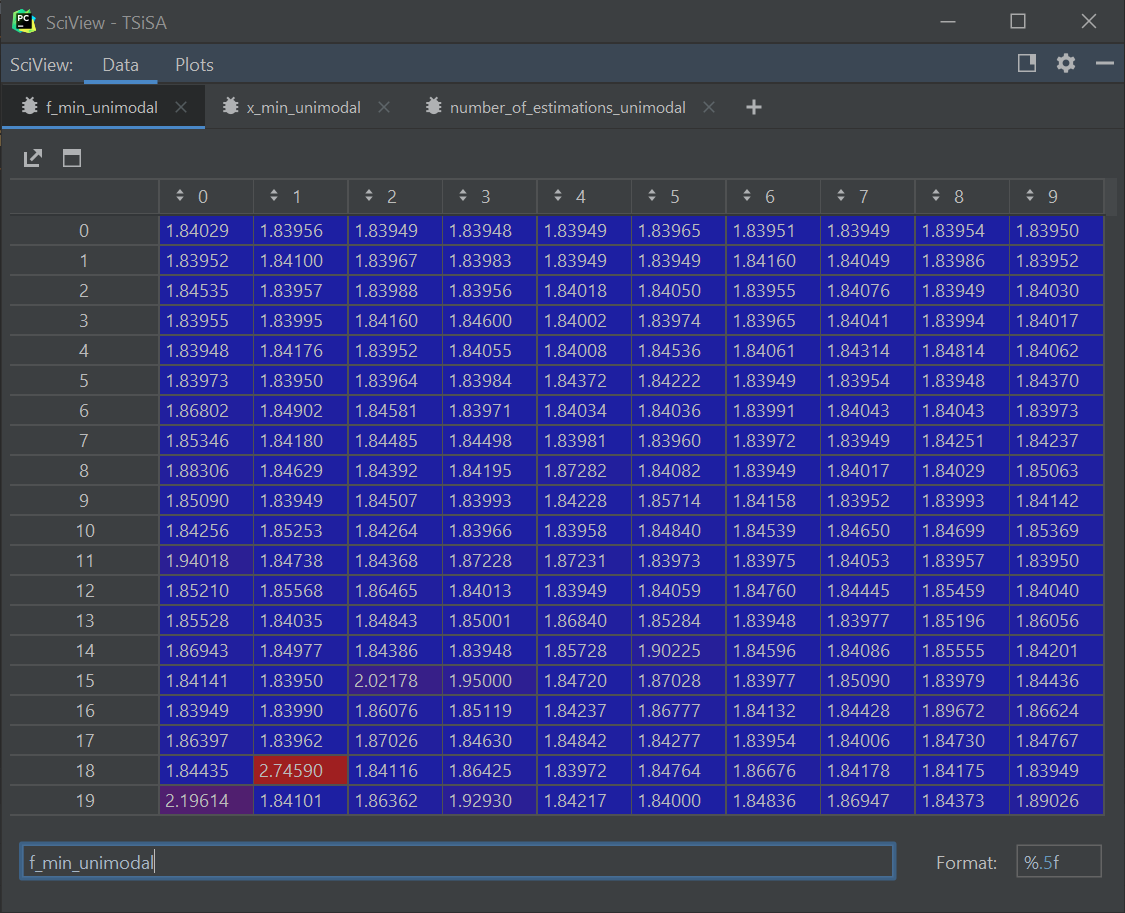


Рисунок 11. Матрица со значениями найденных минимумов.

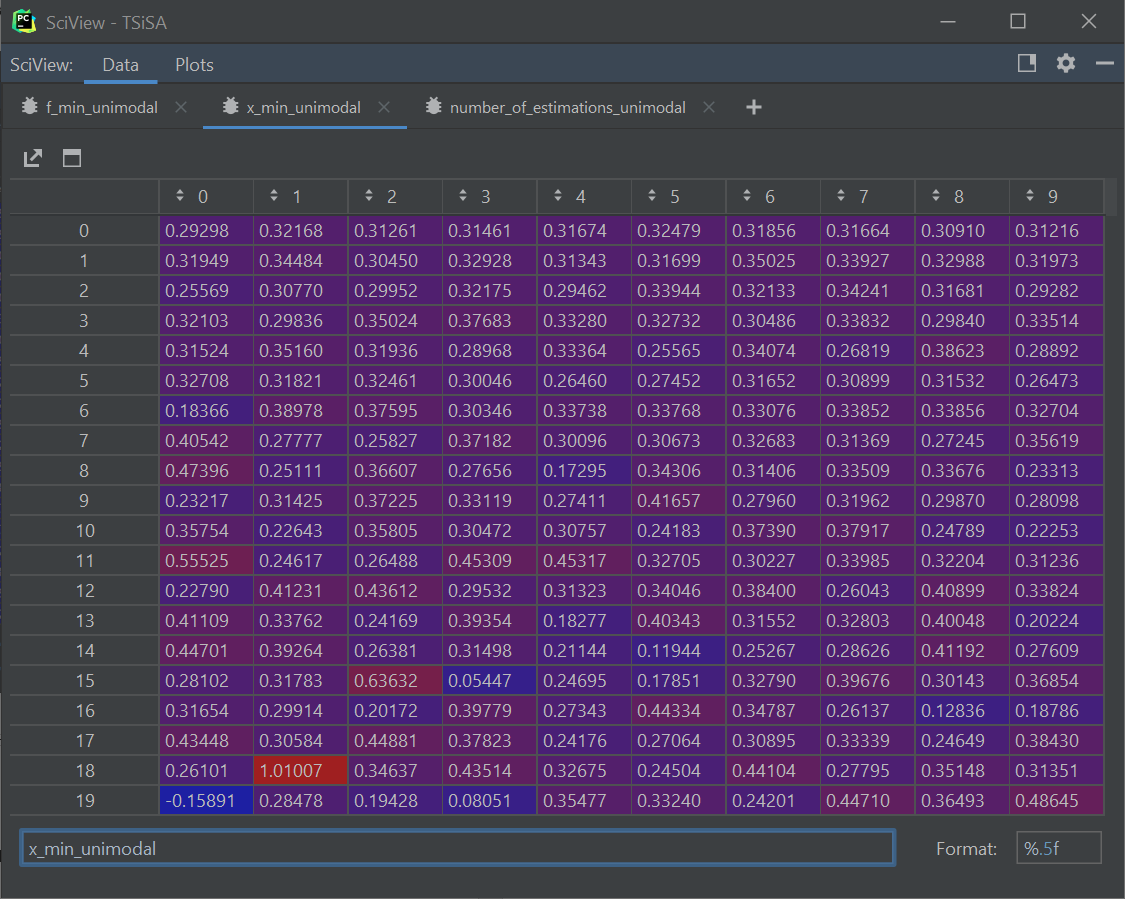


Рисунок 12. Матрица со значениями координат найденных минимумов.

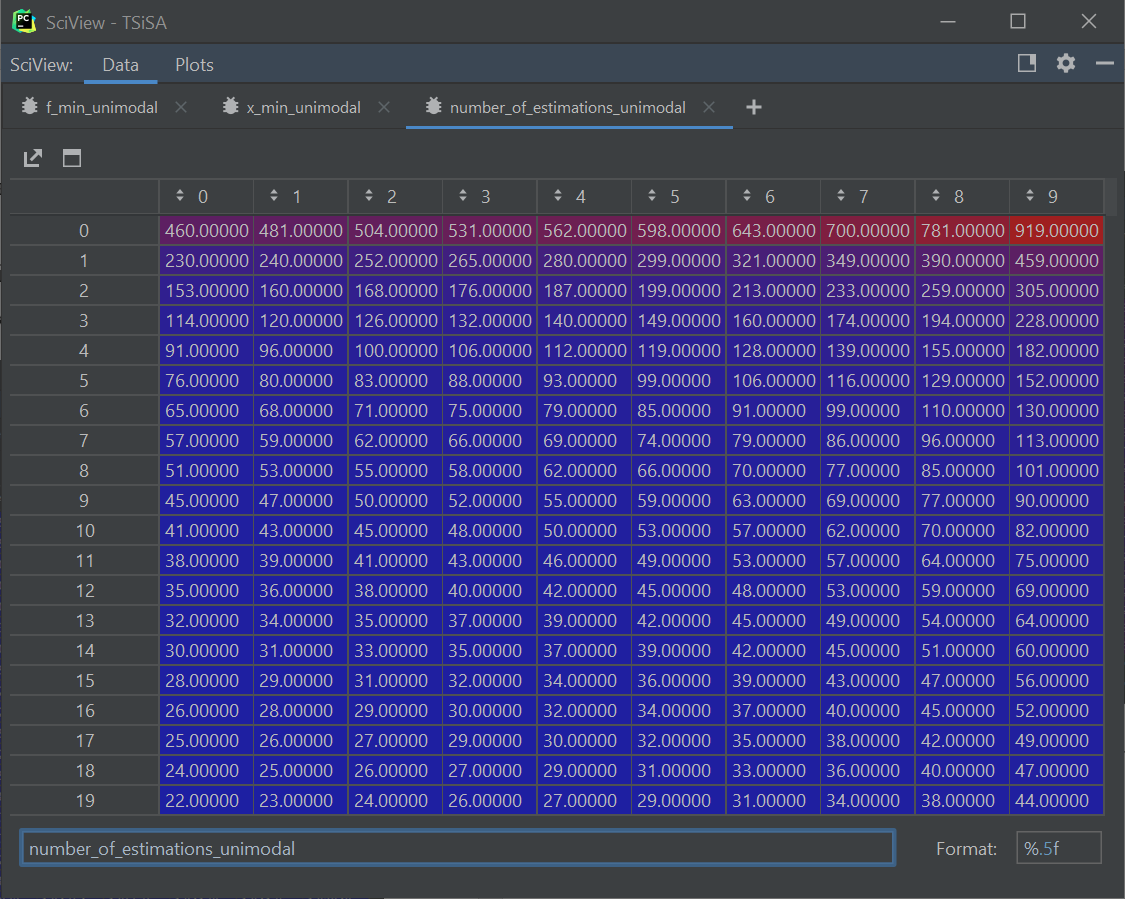


Рисунок 13. Матрица со значениями количества измерений.

**Вывод:**

Изучен метода случайного поиска экстремума на примере унимодальной и мультимодальной функций одного переменного.

Анализируя результаты поиска экстремума мультимодальной функции методом случайного поиска при разных количествах измерений, можно заметить, что при малом количестве измерений за минимум функции может быть принят локальный минимум. Однако, с увеличением количества испытаний, вероятность этого снижается.

Для унимодальной функции метод работает более стабильно.