Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Кафедра «Информатика»
кафедра

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 4

Метод Нелдера-Мида
Тема

Преподаватель			В. В. Тынченко
		Подпись, дата	Инициалы, Фамилия
Студент	КИ19-17/1Б, №031939174		А. К. Никитин
	Номер группы, зачетной книжки	Подпись, дата	Инициалы, Фамилия

Красноярск 2021

1 Постановка задачи

Разработать программу, реализующую метод Нелдера-Мида.

Найти безусловный экстремум функции, выбранной в соответствии с заданием, с использованием разработанной программы.

Функция: $(x_2 + x_1 - 1)^2 + 2(x_1 - 2)^2 \to min$

2 Описание метода

Метод Нелдера—Мида предназначен для минимизации функции n действительных переменных с использованием лишь вычисляемых на каждом шаге значений минимизируемой функции (метод нулевого порядка). Метод Нелдера—Мида на каждом шаге итеративного процесса хранит невырожденный симплекс – геометрический объект в n-мерном пространстве ненулевого объема, являющийся выпуклой оболочкой, натянутой на n+1 вершину.

Каждая итерация прямого симплекс-метода поиска минимума начинается с построения симплекса, который задается своими n+1-ой вершинами и вычисляемыми в этих вершинах значениями функции. Затем многогранник дополняется одной либо несколькими точками вместе со значениями функции в них. Одна или несколько вершин после этого отбраковывается. Итерационный процесс завершается тогда, когда вершины симплекса и вычисленные в них значения функции при сравнении с предыдущей итерацией удовлетворяют некоторым условиям сходимости.

3 Исходные тексты программ

На листинге 1 представлен код программы, реализующий задание.

Листинг 1 – Метод Нелдера-Мида

```
import numpy as np
from const import f
from functools import reduce
import math
```

Окончание листинга 1

```
def nelder_mead_algorithm(vertexes, alpha=1, beta=0.5, gamma=2, epsilon=0.01):
    n = len(vertexes) - 1

    while True:
        vertexes_y = list(map(f, vertexes))

        vertexes = np.array(sorted(vertexes, key=f))

        min_vert, premin_vert = vertexes[:2]
        worst_vert = vertexes[-1]

        mass_center = sum(vertexes[:-1]) / n

        sigma = math.sqrt(reduce(lambda prev, y: prev + (y - f(mass_center)) **
2, vertexes_y) / (n + 1))

        if sigma <= epsilon:
            return min_vert

        reflected_vert = mass_center + alpha * (mass_center - worst_vert)

        if f(reflected_vert) <= f(min_vert):
            stretched_vert = mass_center + gamma * (reflected_vert -
mass_center)
            vertexes[-1] = stretched_vert if f(stretched_vert) < f(min_vert)
else reflected_vert

        elif f(premin_vert) < f(reflected_vert) <= f(worst_vert):
            compressed_vert = mass_center + beta * (worst_vert - mass_center)
            vertexes[-1] = compressed_vert

        else:
            vertexes = min_vert + (vertexes - min_vert) / 2


if __name__ == '__main__':
    print(nelder_mead_algorithm(((-10, -10), (10, -10), (0, 10)), alpha=1,
beta=0.5, gamma=2, epsilon=0.05))
```

4 Исследование влияния параметров метода на точность и скорость нахождения решения

Из рисунка 1 можно заключить, что чем меньше параметр эпсилон, тем точнее решение, однако тем больше шагов необходимо алгоритму для нахождения ответа.

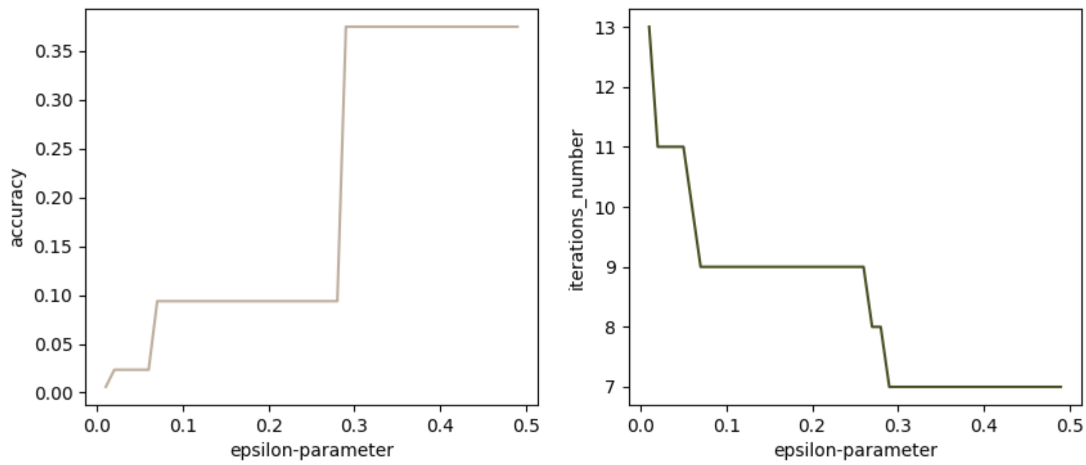

Рисунок 1 – Влияние параметра ε на точность и производительность

Из рисунка 2 можно заключить, что параметр α, с увеличением повышает количество итераций, однако не имеет очевидной зависимости между параметром и точностью.

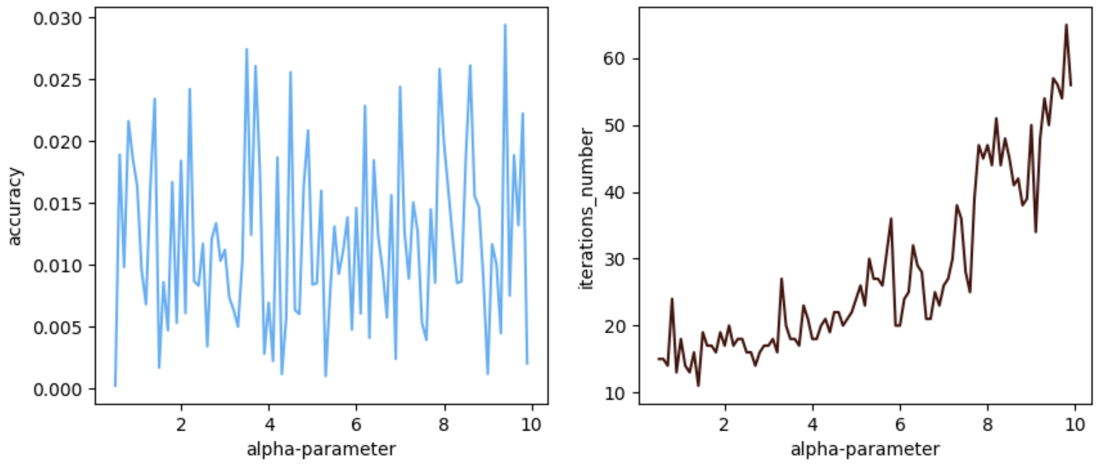

Рисунок 2 - Влияние параметра α на точность и производительность

Из рисунка 3 можно заключить, что величина сжатия в данном случае никак не повлияла на работу алгоритма.

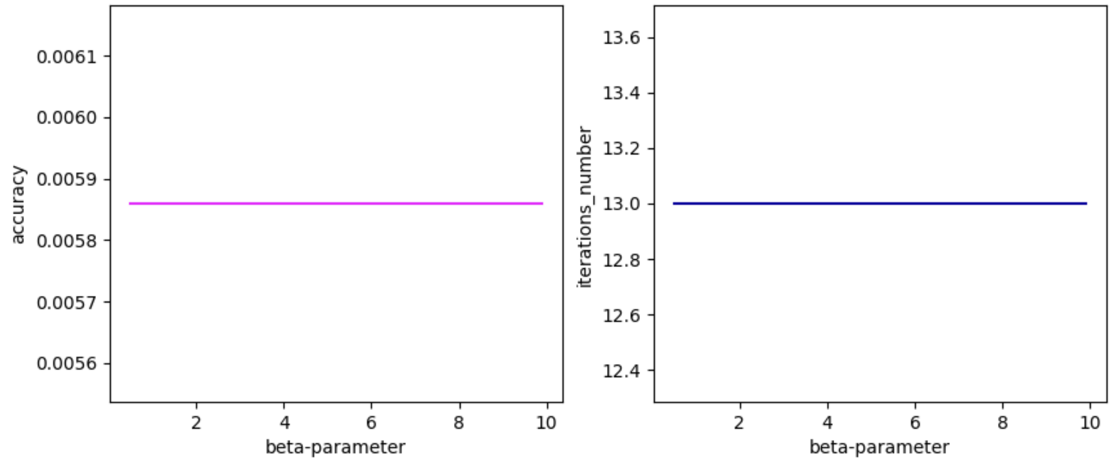

Рисунок 3 - Влияние параметра β на точность и производительность

Из рисунка 4 можно заключить, что параметр γ перестает оказывать влияния после γ=2.

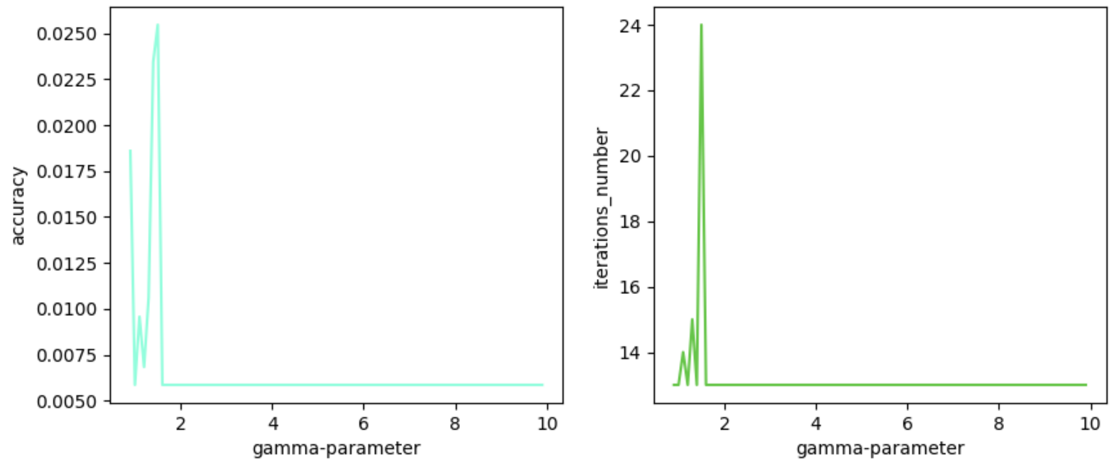

Рисунок 4 – Влияние параметра γ на точность и производительность

5 Вывод

В результате данной работы был реализован и проанализирован метод Нелдера-Мида для поиска локального минимума многомерной функции. Также были проанализированы гиперпараметры метода и их влияние на работу функции.