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

Институт космических и информационных технологий
институт

Кафедра «Информатика»
кафедра

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 14

Сравнительный анализ эффективности численных методов первого порядка для поиска безусловного экстремума

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

На основании результатов выполнения практических работ модуля "Численные методы первого порядка для поиска безусловного экстремума" сравнить реализованные алгоритмы по точности и скорости решения задач оптимизации, варьируя параметры алгоритмов. Для проведения вычислительных экспериментов самостоятельно выбрать 3 целевые функции и интервалы неопределенности, интересные с точки зрения исследования. Результаты вычислительных экспериментов представить в табличном виде, прокомментировать их и сделать обоснованный вывод об особенностях работы исследуемых алгоритмов и их эффективности на различных целевых функциях.

2 Функции для исследования

Ниже представлен список функций нескольких переменных.

1. $f(x_1, x_2) = 3x_1^2 + x_1 x_2 + 2x_2^2 - x_1 - 4 x_2$
2. $f(x_1, x_2) = (x_1 - 4)^2 + (x_2 - 1)^2$
3. $f(x_1, x_2) = 2x_1^2 -2x_1 + x_1x_2 - x_2 + x_2^2$

3 Исходные тексты программ

На листинге 1 представлен код программы, проводящий анализ функций.

Листинг 1 – Анализ функций нескольких переменных

```
import numpy as np
import matplotlib.pyplot as plt
from const import exact_extremum, assess_func
from fletcher_reeves import fletcher_reeves_method


def check_params_format(*args):
    if any(map(lambda arg: arg is None, args)):
        return False

    are_params_list = list(map(lambda x: isinstance(x, list), args))
    if any(are_params_list) and not all(are_params_list):
        return False
```

```
    if all(are_params_list) and len(set(map(len, args))) != 1:
        return False
    return True



def to_list(var):
    if not isinstance(var, list):
        return [var]
    return var



def assess(method, param_name, param_start, param_end, param_step, **kwargs):
    if not check_params_format(param_start, param_end, param_step):
        raise Exception('Wrong parameter data type!')

    param_start, param_end, param_step = to_list(param_start),
to_list(param_end), to_list(param_step)

    deltas = []
    parameters = []
    iteration_numbers = []

    for params in zip(
            *[np.arange(param_start[j], param_end[j], param_step[j]) for j in
range(len(param_step))]
    ):

        params = list(params) if len(params) != 1 else params[0]

        parameter_val = {param_name: params}

        estimated_extremum, iteration_num = method(**parameter_val, **kwargs)
        delta = sum([abs(exact_extremum[i] - estimated_extremum[i]) for i in
range(len(estimated_extremum))])

        deltas.append(delta)
        parameters.append(params)
        iteration_numbers.append(iteration_num)

    plt.figure(figsize=(10, 4))
    plt.subplot(1, 2, 1)
```

```
    plt.plot(parameters, deltas, c=np.random.rand(3,))
    plt.xlabel(param_name + '-parameter'),
    plt.ylabel('accuracy')
    plt.subplot(1, 2, 2)
    plt.plot(parameters, iteration_numbers, c=np.random.rand(3, ))
    plt.xlabel(param_name + '-parameter'),
    plt.ylabel('iterations_number')
    plt.show()


if __name__ == '__main__':
    assess(fletcher_reeves_method, 'M', 1, 15, 1, f=assess_func, x0=(-10, 10))
```

4 Сравнительный анализ методов

На рисунках 1, 2, 3 представлены графики изменения точности и производительности методов Хука-Дживса, Пауэлла, Розенброка, Нелдера-Мида от параметра epsilon.

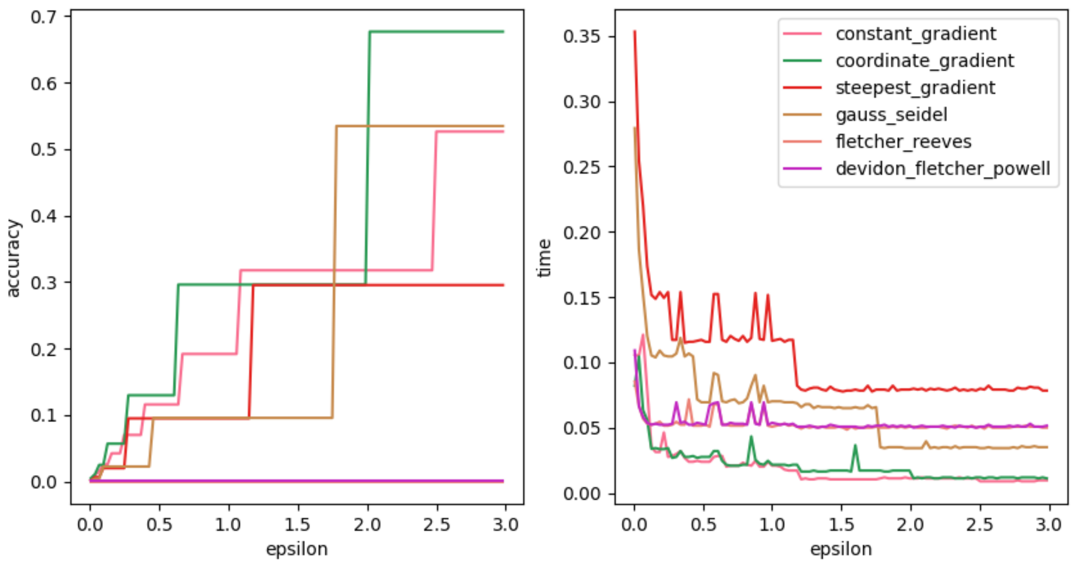

Рисунок 1 – Изменение точности алгоритмов от параметра epsilon функции $f(x_1, x_2) = 3x_1^2 + x_1 x_2 + 2x_2^2 - x_1 - 4 x_2$

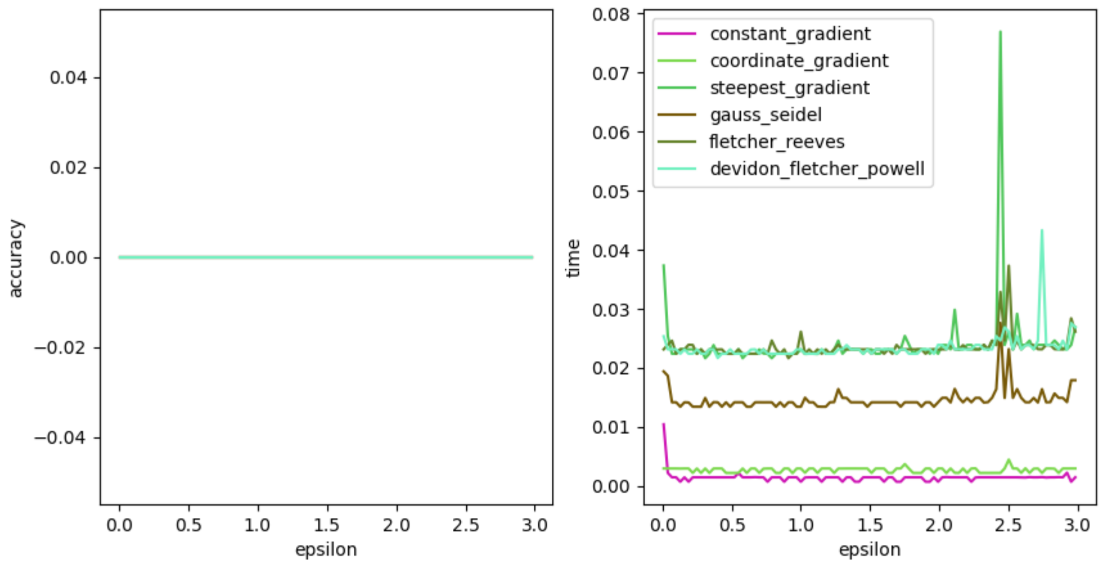

Рисунок 2 – Изменение точности алгоритмов от параметра epsilon функции $f(x_1, x_2) = (x_1 - 4)^2 + (x_2 - 1)^2$

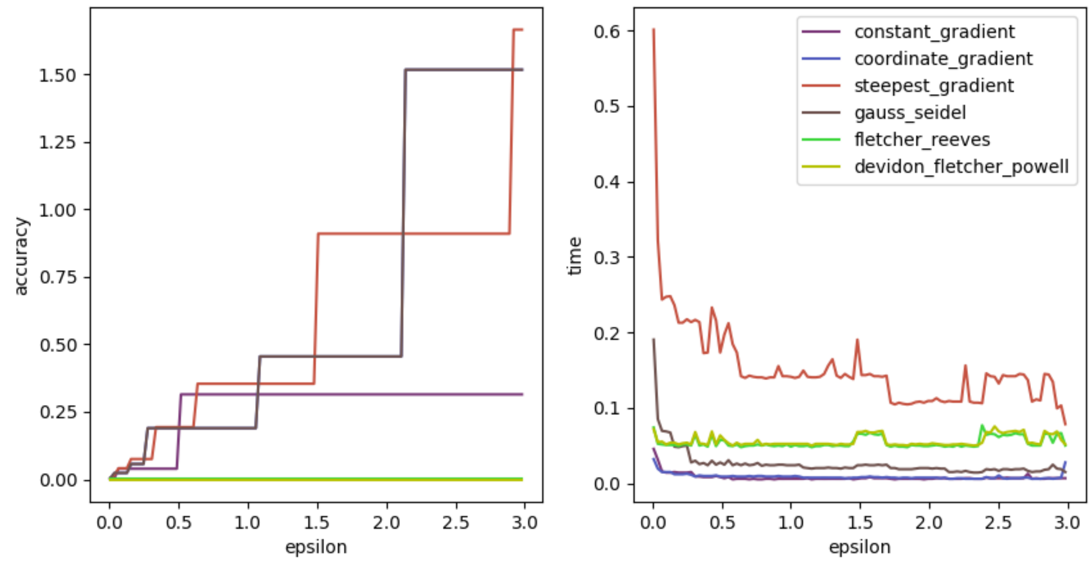

Рисунок 3 – Изменение точности алгоритмов от параметра epsilon функции $f(x_1, x_2) = 2x_1^2 - 2x_1 + x_1x_2 - x_2 + x_2^2$

5 Вывод

Таким образом, для минимизации функций нескольких переменных переменной наиболее эффективными по точности оказались методы Флетчера-Ривса и Дэвидона-Флетчера-Пауэлла; по времени методы постоянного и координатного спуска..