

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по учебной практике
Тема: Поиск глобального минимума на криволинейной поверхности

Студентка гр. 1381	_____	Рымарь М.И.
Студентка гр. 1384	_____	Мухачёва П.Р.
Студент гр. 1384	_____	Белокобыльский И.В.
Преподаватель	_____	Жангиров Т.Р.

Санкт-Петербург
2023

ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студентка Рымарь М.И. группы 1381

Студентка Мухачёва П.Р. группы 1384

Студент Белокобыльский И.В. группы 1384

Тема практики: Поиск глобального минимума на криволинейной поверхности

Задание на практику:

Задача для алгоритма роя частиц. Поиск глобального минимума на криволинейной поверхности задаваемой уравнением $z=f(x,y)$. Формула должна вводиться через GUI. Для проверки корректности можно использовать функцию Растринга.

Сроки прохождения практики: 30.06.2023 – 13.07.2023

Дата сдачи отчета: 03.07.2023

Дата защиты отчета: 03.07.2023

Студентка	_____	Рымарь М.И.
Студентка	_____	Мухачёва П.Р.
Студент	_____	Белокобыльский И.В.
Руководитель	_____	Жангиров Т.Р.

АННОТАЦИЯ

Данная работа посвящена поиску глобального минимума на криволинейной поверхности, задаваемой уравнением $z=f(x,y)$, с использованием алгоритма роя частиц. Проект реализован на языке программирования Python с использованием библиотек Tkinter, Matplotlib и NumPy для создания графического интерфейса пользователя (GUI) и эффективной работы с численными массивами.

SUMMARY

This project is devoted to finding a global minimum on a curved surface given by the equation $z=f(x,y)$, using particle swarm algorithm. The project is implemented in the Python programming language using the Tkinter, Matplotlib and NumPy libraries to create a graphical user interface (GUI) and work with numerical arrays efficiently.

СОДЕРЖАНИЕ

	Введение	5
1.	Первая итерация	6
1.1.	Прототип GUI	6
1.2.	Выбор модификаций, метрик и настраиваемых параметров	7
1.3.	Стек технологий и сторонние библиотеки	8
1.4.	Распределение ролей в бригаде	8
	Заключение	9
	Список использованных источников	10

ВВЕДЕНИЕ

Цель данного проекта - разработка программы для поиска глобального минимума на криволинейной поверхности.

Задачи проекта:

1. Реализовать GUI с помощью библиотеки Tkinter для удобного ввода формулы криволинейной поверхности, а также необходимых параметров, а также визуализацию криволинейной поверхности с помощью библиотеки
2. Разработать эффективный алгоритм поиска глобального минимума на криволинейной поверхности, основываясь на методе роя частиц, используя библиотеку NumPy
3. Визуализировать результаты на графике для удобного анализа и сравнения полученных минимумов при помощи библиотеки Matplotlib.
4. Протестировать полученный результат используя в качестве примера функцию Растринга для проверки корректности результатов поиска минимума.

1. ПЕРВАЯ ИТЕРАЦИЯ

1.1. Прототип GUI

Прототип GUI разработан с использованием Figma. Он состоит из двух окон. Первое окно является стандартным и предназначено для ввода функции, параметров и количества итераций. В этом окне также присутствуют кнопки для построения конечного графика, пошагового построения графика (перехода на следующий шаг) и сброса данных. Кроме того, в окне есть кнопка, при нажатии на которую открывается дополнительное окно с графиком изменения функции.

Дополнительное окно содержит график изменения функции в зависимости от количества итераций. Этот график позволяет анализировать и сравнивать полученные минимумы.

Прототип GUI разработан с учетом удобства использования и позволяет пользователю легко вводить необходимые параметры, а также анализировать результаты поиска глобального минимума на криволинейной поверхности.

Макеты стандартного и дополнительного окна представлены на рисунках 1 и 2, соответственно.

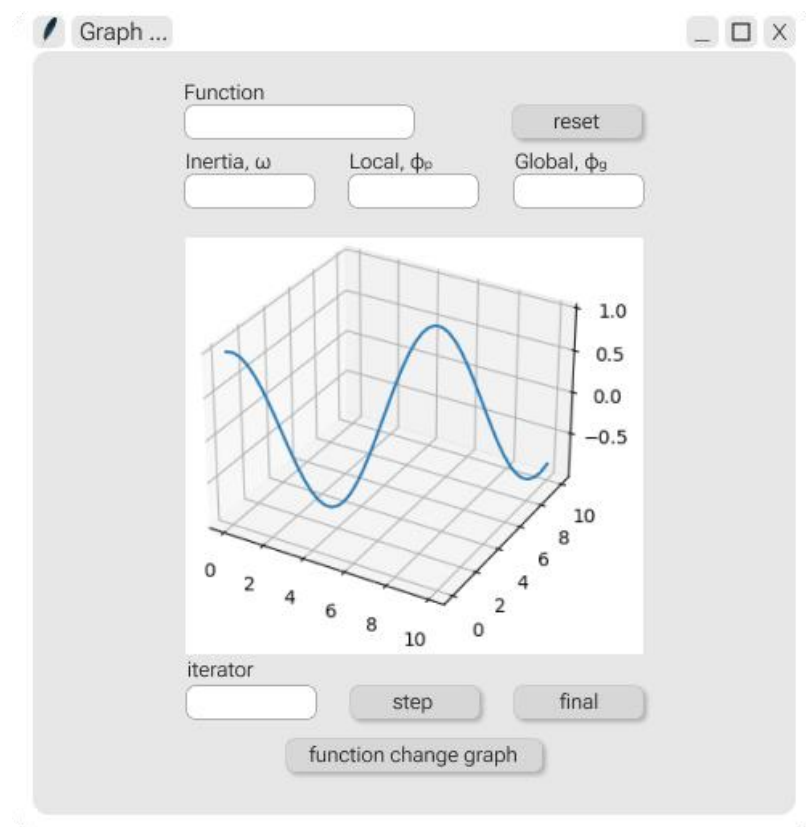


Рисунок 1 – Макет стандартного окна

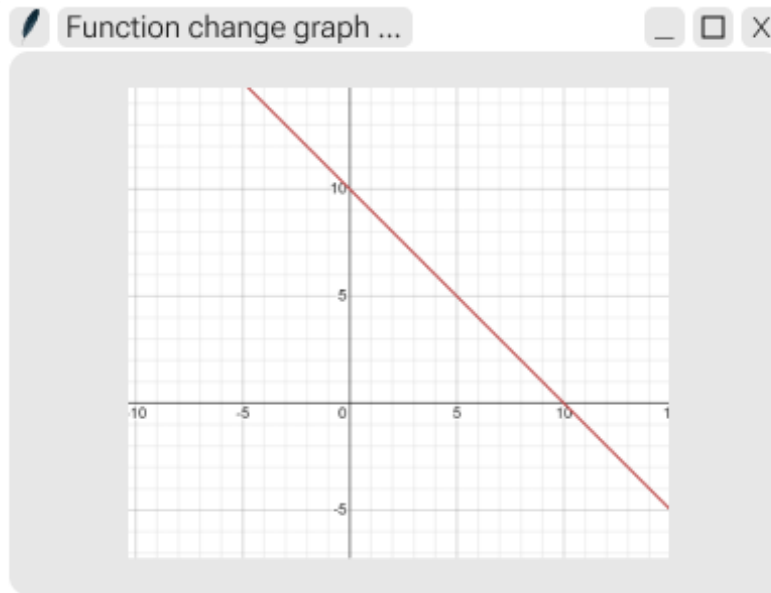


Рисунок 2 – Макет дополнительного окна

1.2. Выбор модификаций, метрик, настраиваемых параметров

Суть алгоритма заключается в изменении положения частиц в пространстве, задаваемой функцией $f(x, y)$. Сама функция выглядит следующим образом:

$$\mathbf{v}_i \leftarrow \omega \mathbf{v}_i + \phi_p \mathbf{r}_p \cdot (\mathbf{p}_i - \mathbf{x}_i) + \phi_g \mathbf{r}_g \cdot (\mathbf{g} - \mathbf{x}_i)$$

Где ω – инерция, ϕ_p – коэффициент локального оптимума, ϕ_g – коэффициент глобального оптимума, \mathbf{v}_i – скорость частицы, \mathbf{r}_p , \mathbf{r}_g – случайные векторы из равномерного распределения на отрезке $[0, 1]$, \mathbf{p}_i – локальный оптимум для i -й частицы, \mathbf{g} – глобальный оптимум, \mathbf{x}_i – положение частицы в пространстве. В качестве операции умножения имеется в виду покомпонентное умножение векторов.

Первые три из описанных выше параметров доступны для ввода пользователем. Также, помимо этого, пользователь может повлиять на количество итераций алгоритма, так как сам метод роя частиц является стохастическим, следовательно, не имеет единого для всех алгоритмов критерия останова.

Было проведено сравнение двух модификаций алгоритма – с инерцией и без нее. На тестовых данных вариант с инерцией показал себя лучше, поэтому

было принято решение оставить его. К тому же, чтобы убрать влияние инерции на вычисления, достаточно приравнять ее к 1.

1.3. Стек технологий и сторонние библиотеки

В качестве языка программирования был выбран Python, так как на нем присутствует наиболее удобные на взгляд участников бригады решения для визуализации – библиотеки Matplotlib для составления графиков и Tkinter для создания GUI.

Так как язык Python предоставляет не самый эффективный способ работы с числовыми массивами, в дополнение к представленному выше списку была взята библиотека NumPy.

1.4. Распределение ролей в бригаде

Мухачёва Полина – Написание самого алгоритма роя частиц;

Рымарь Мария – Разработка пользовательского интерфейса, анализ проделанной всей бригадой работы;

Белокобыльский Илья – Проектирование архитектуры приложения, выбор стека технологий, тестирование.

ЗАКЛЮЧЕНИЕ

В рамках первой итерации проекта выполнены следующие задачи:

- Разработан прототип GUI, который интуитивно понятен пользователю;
- Проведены исследования и выбраны модификации, которые будут использоваться в алгоритме глобальной оптимизации. Также было составлено описание настраиваемых параметров алгоритма, чтобы пользователи могли настроить его под свои потребности и задачи;
- Представлен стек технологий, который будет использоваться в проекте. Стек включает в себя использование таких сторонних библиотек, как NumPy, Matplotlib и Tkinter;
- Распределены роли между участниками бригады для продуктивной дальнейшей работы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Jenyay.net. (2021). Particle Swarm Optimization (Метод роя частиц) [Online]. Available: <https://jenyay.net/Programming/ParticleSwarm>
2. Wikipedia.org. (2021). Метод роя частиц [Online]. Available: https://ru.wikipedia.org/wiki/Метод_роя_частиц