**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Программа бакалавриата**

**“Большие данные и распределенная цифровая платформа”**

**ОТЧЕТ**

**по лабораторной работе №2**

**по дисциплине «Алгоритмы и структуры данных»**

**на тему Генетический алгоритм для оптимизации функции**

**Студент гр. 22Б16-пу \_\_\_\_\_\_\_\_\_\_\_\_\_ Шарабарин М.С.**

**Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_ Дик А.Г.**

**Санкт-Петербург**

**2024 г.**

# Оглавление

# Цель работы

Целью лабораторной работы является разработка и реализация программы, использующей генетический алгоритм для минимизации заданной целевой функции. Основное внимание уделяется изучению механизмов мутации, кроссовера и селекции в контексте оптимизации.

**Описание задачи (формализация задачи)**

* **Задача заключается в минимизации функции: [ f(x\_1, x\_2) = 4(x\_1 - 5)^2 + (x\_2 - 6)^2 ] Где ( x\_1 ) и ( x\_2 ) находятся в пределах заданных границ (-10, 10). Программа должна находить значения этих переменных, которые минимизируют целевую функцию.**

## 

**Теоретическая часть**

Генетические алгоритмы (ГА) являются эвристическими методами поиска, вдохновлёнными процессами естественного отбора. Основные компоненты генетических алгоритмов включают:

* **Инициализация**: Генерация начальной популяции возможных решений.
* **Отбор**: Выбор родителей для создания нового поколения на основе их приспособленности.
* **Кроссовер**: Комбинация характеристик двух родителей для создания потомства.
* **Мутация**: Случайная модификация потомству для поддержания генетического разнообразия.

## **Основные шаги программы**

1. **Генерация начальной популяции**: Создание множества потенциальных решений.
2. **Отбор родителей**: Использование турниров для выбора лучших индивидов.
3. **Кроссовер**: Генерация новых решений на основе выбранных родителей.
4. **Мутация**: Модификация потомства с помощью случайных изменений.
5. **Итерации**: Повторение отборов, кроссовера и мутаций на протяжении нескольких поколений.
6. **Поиск наилучшего решения**: Определение индивида с минимальным значением функции завершает выполнение алгоритма.

**Ограничения**:

* Данные должны быть в формате csv

**Основные шаги программы**

1. **Анонимизация данных**: Замена реальных значений на анонимизированные, включая ФИО, номера паспортов и карт.
2. **Обобществление номеров и цен**: Применение маскирования и обобщения.
3. **Обработка временных меток**: Добавление случайных значений для анонимизации временных данных.
4. **Выделение уникальных строк и вычисление K-анонимити**: Определение групп записей и их количества для защиты данных.
5. **Представление результатов**: Вывод результатов анализа, таких как размер исходного и анонимизированного наборов данных.

**Блок схема программы**

****

Рис 1. Блок-схема основной программы

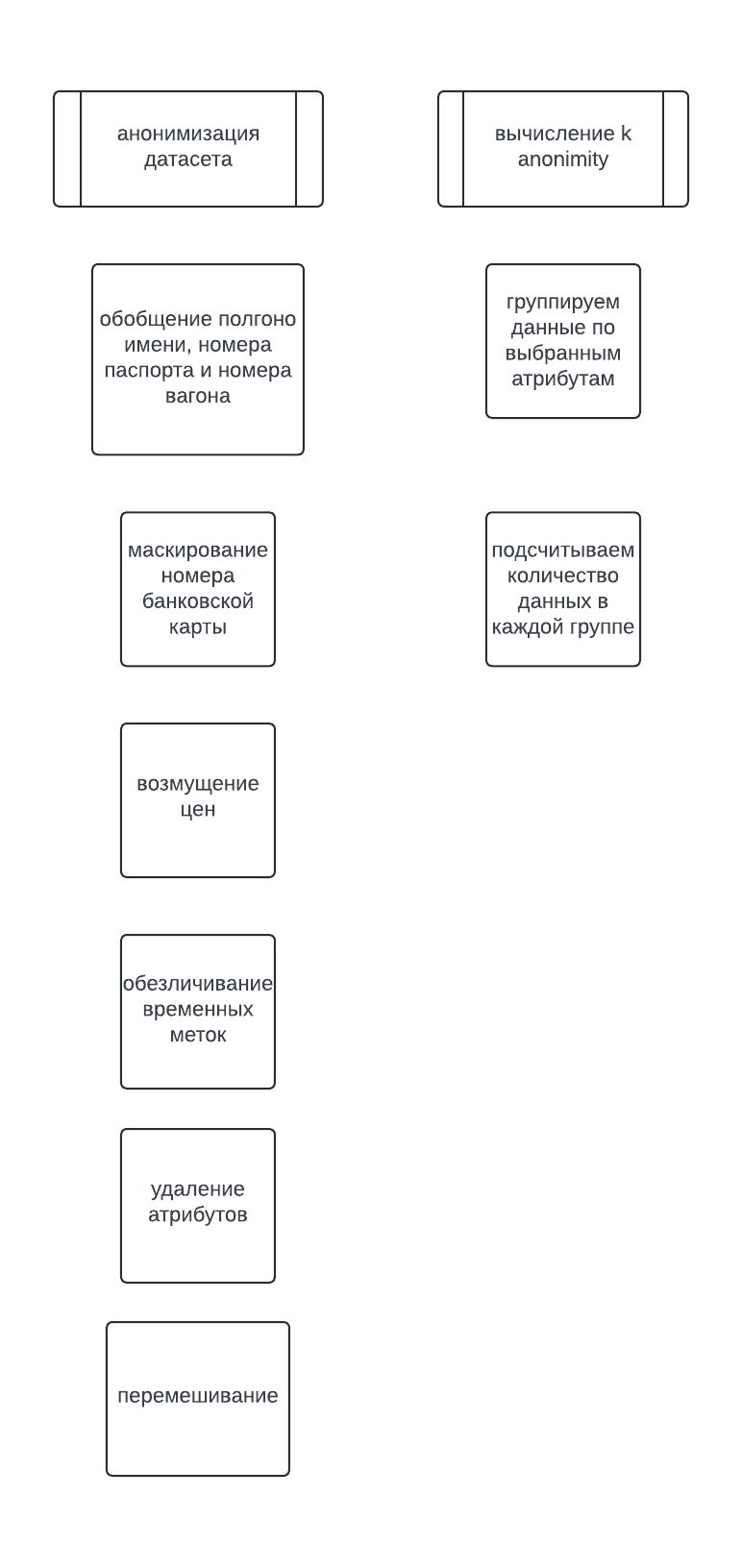


Рис 2. Блок-схема подпрограмм

**Описание программы**

Программа реализована на Python и состоит из нескольких основных функций:

* fitness\_function: Вычисляет значение целевой функции.
* generate\_initial\_population: Генерирует начальную популяцию решений.
* select\_parents: Выбирает лучших индивидов из популяции для скрещивания.
* crossover: Выполняет операции кроссовера.
* mutate: Применяет мутацию к потомству.
* genetic\_algorithm: Управляет основным процессом ГА.

### **Рекомендации для пользователя**

При использовании программы рекомендуется:

* Убедиться, что входные параметры правильно заданы.
* Изучить документацию для понимания настройки гиперпараметров (размер популяции, коэффициенты кроссовера и мутации).

### **Рекомендации для программиста**

* Обеспечить комментарии к функциям и основному коду для облегчения чтения.
* Реализовать обработку исключений для увеличения устойчивости программы.
* Провести тестирование с разными значениями гиперпараметров для анализа эффективности.

**Исходный код программы**

**<https://colab.research.google.com/drive/1IWGOe2Rj0q4tXwp-jfXuLU0an_3KVzuF#scrollTo=8YXla8aoeP4I>**

**Контрольный пример**

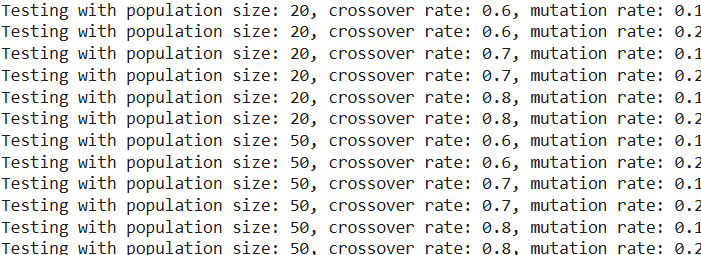
**

Рис 3.Подбор гиперпараметров



Рис 4.Вывод локального минимума

**Вывод**

В данной лабораторной работе реализована программа для анонимизации и обезличивания данных в заданном наборе. Успешно достигнуты цели по обеспечению конфиденциальности личной информации при анализе больших данных.

**Источники**

1. Pandas библиотека: <https://pandas.pydata.org/>
2. Numpy библиотека: https://numpy.org/
3. CSV модуль: <https://docs.python.org/3/library/csv.html>
4. Concurrent Futures модуль: <https://docs.python.org/3/library/concurrent.futures.html>