

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

ОТЧЕТ
по учебной практике
Тема: Генетические алгоритмы и ПСА

Студент гр. 3344

Клюкин А.В.

Студентка гр. 3344

Якимова Ю.А.

Руководитель

Жангиров Т.Р.

Санкт-Петербург

2025

ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студент Ключкин А.В., Студентка Якимова Ю.А.

Группа 3344

Тема практики: Генетические алгоритмы и ПСА

Задание на практику:

1) Задача о назначениях

Пусть имеется N работ и N кандидатов на выполнение этих работ, причем назначение j -й работы i -му кандидату требует затрат $c_{ij} > 0$. Необходимо назначить каждому кандидату по работе, чтобы минимизировать суммарные затраты. Причем каждый кандидат может быть назначен на одну работу, а каждая работа может выполняться только одним кандидатом.

Сроки прохождения практики: 25.06.2025 – 07.07.2025

Дата сдачи отчета:

Дата защиты отчета:

Студентка		Якимова Ю.А
Студент		Ключкин А.В.
Руководитель		Жангиров Т.Р.

АННОТАЦИЯ

Практика направлена на изучение и применение генетических алгоритмов для решения оптимизационных задач. Требуется разработать программу с графическим интерфейсом, которая будет находить оптимальное распределение работ между кандидатами, минимизируя суммарные затраты. Основные этапы включают:

- Реализацию ГА без использования специализированных библиотек.
- Интеграцию GUI для ввода данных различными способами, настройки параметров ГА и визуализации результатов.
- Обеспечение пошагового выполнения алгоритма с отображением лучшего решения, его стоимости и средних показателей поколения.
- Построение графика изменения приспособленности в процессе работы алгоритма.
- Демонстрацию возможности повторного запуска с новыми параметрами и данными.

СОДЕРЖАНИЕ

	Введение	5
1.	Анализ и проектирование	6
1.1.	Постановка задачи	6
1.2.	Представление решения	6
1.3.	Структура проекта	6
1.4.	Прототип GUI	8
1.5.	Порядок реализации, роли и сроки	9

ВВЕДЕНИЕ

Целью данной практики является изучение принципов работы ГА и применение их для решения задачи на оптимизацию. Разработка программы с графическим интерфейсом для телеметрии и демонстрации пошаговой работы алгоритма для анализа результатов.

Задачи:

- Изучить теоретические основы работы ГА
- Разработать алгоритм, решающий задачу с помощью ГА
- Реализовать графический интерфейс
- Связать графический интерфейс с основным алгоритмом
- Протестировать работу программы на различных входных данных

1. АНАЛИЗ И ПРОЕКТИРОВАНИЕ

1.1. Постановка задачи

Задача заключается в оптимальном разделении N работ между N кандидатами, где каждое значение (i, j) имеет стоимость $c[i][j]$. Требуется найти такое распределение, что каждый кандидат выполняет ровно одну работу и суммарная стоимость назначений минимальна.

1.2. Представление решения

Кодирование решения - хромосома. Будет представляться в виде массива, длиной N , где индексом элемента является номер кандидата, а значение элемента - номер работы, которую он выполняет.

Функция приспособленности вычисляется как сумма стоимости назначений из данной матрицы. Из хромосомы берется индекс кандидата и номер работы. Чем меньше сумма, тем лучше решение.

Генерация начальной популяции создается случайным образом с проверкой на валидность.

Скрещивание будет реализовано с помощью метода “Упорядоченное скрещивание”. Метод равномерного или k -точечного скрещивания не подходит, тк возможно нарушение валидности хромосомы.

Мутация будет реализована через метод обмена и обращения.

Отбор подчиняется правилу рулетки.

1.3. Структура проекта

Проект разбит на две части - реализация генетического алгоритма и работа с графической составляющей. С точки зрения структуры, будет разделено на несколько модулей. Основные компоненты:

Главный модуль - `main.py`. Является точкой запуска программы. Инициализирует GUI и управляет запуском ГА.

Модуль генетического алгоритма - `genetic_solver.py`. Реализация ГА, содержащий методы скрещивания, мутации, отбора.

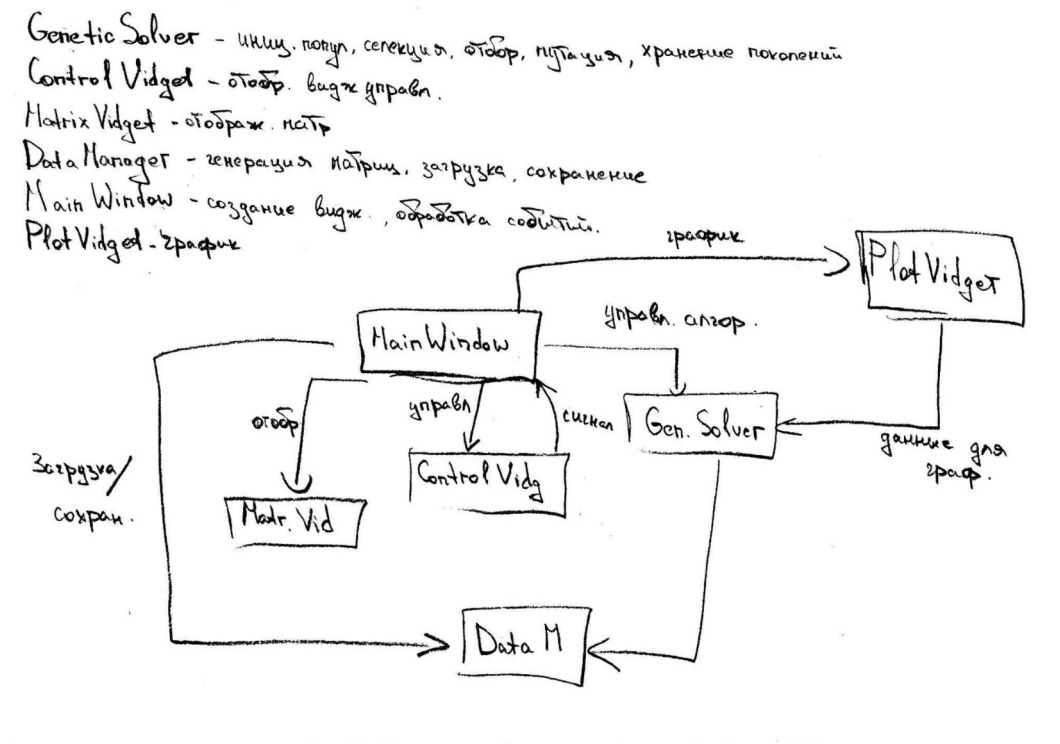
Модуль управления данными - data_manager.py. Работа со входными данными, генерацией, хранением для графического представления.

Модули графического интерфейса - matrix_widget, control_widget, plot_widget

Различные вспомогательные функции - utils.py

Схема взаимодействия модулей

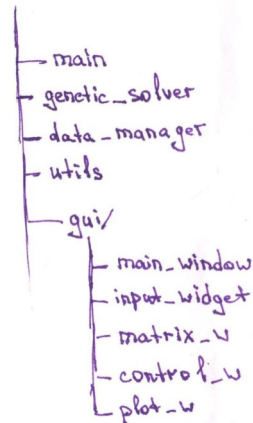
1. Пользователь запускает программу через main.py.
2. main_window.py загружает интерфейс и предоставляет варианты ввода данных (файл, генерация, ручной ввод).
3. data_manager обрабатывает данные и передает матрицу стоимостей в genetic_solver.
4. genetic_solver выполняет алгоритм, обновляя информацию о поколениях.
5. plot_widget и matrix_widget отображают прогресс и позволяют управлять параметрами.
6. Результаты выводятся в главном окне.



(Рис. 1)

8 gna GUI - PyQt + Matplotlib

Структура



(Рис. 2)

1.4. Прототип GUI

Для создания GUI было решено использовать фреймворк PyQt6. Это решение обусловлено простотой, популярностью фреймворка а также возможностью интеграции Matplotlib в виджеты.

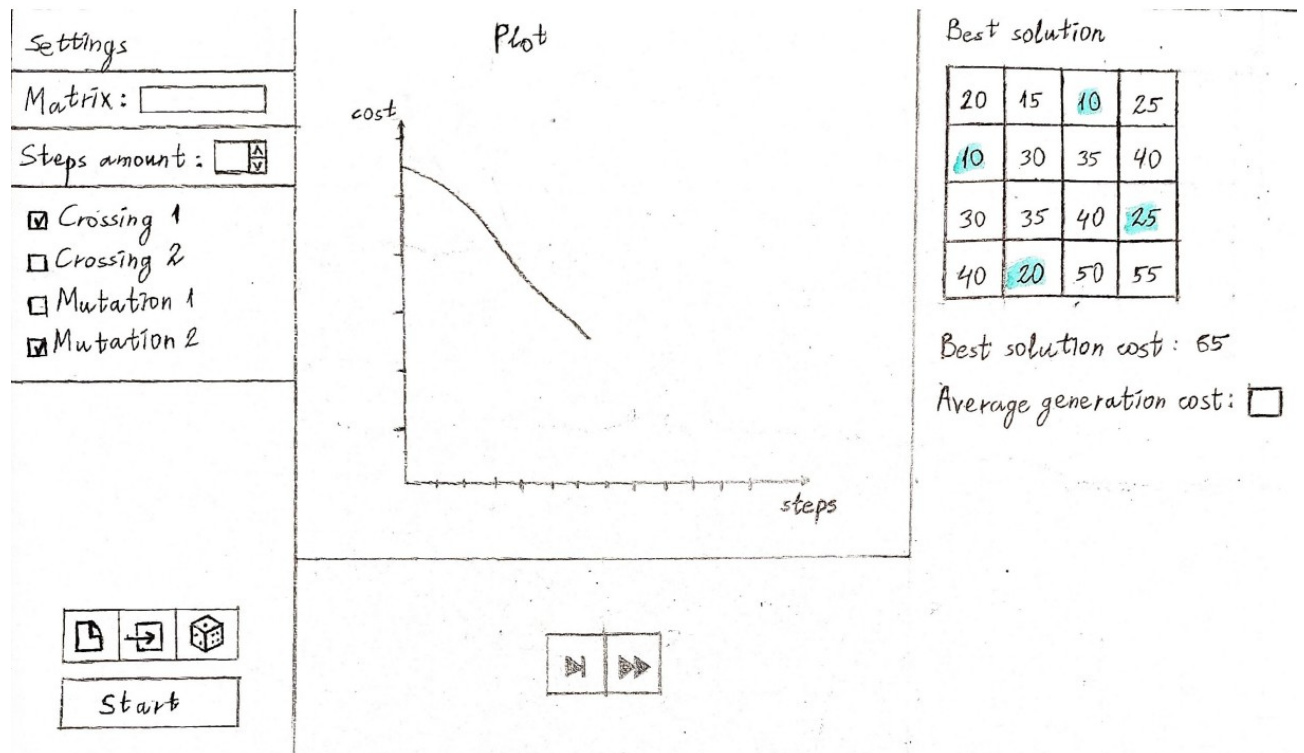
Для начала будет создано главное окно приложения (`main_window`). Оно создает другие виджеты и управляет ими (`input_w`, `matrix_w`, `plot_w`, `control_w`), а также обрабатывает основные события.

`Input_w` предоставляет интерфейс для выбора источника входных данных (из файла, ввод через GUI, случайная генерация) и для их ввода/настройки. Обрабатывает ввод пользовательских данных и предоставляет их в структурированном формате.

`Matrix_w` отображает матрицу затрат.

`Plot_w` отображает график изменения приспособленности (лучшей и средней) в зависимости от поколения в процессе работы генетического алгоритма.

Control_w предоставляет элементы управления для управления выполнением генетического алгоритма.



1.5. Порядок реализации, роли и сроки

- Первоначально будут реализованы модуль хранения данных и работа ГА. Параллельно разработан GUI. (02.07)
- Затем ГА будет доработан, а интерфейс связан с данными алгоритма. (04.07)
- Тестирование и демонстрация полного проекта (06.07)

За создание графического интерфейса, взаимодействие с пользователем, визуализацию данных и управление пользовательским вводом отвечает Якимов Ю.А.

За реализацию генетического алгоритма, обработку данных, расчет целевой функции отвечает Клюкин А.В.