

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

ОТЧЕТ
по учебной практике
по теме «Генетические алгоритмы»
Тема: Задача о кратчайшем пути.

Студенты гр. 0382

Преподаватель

Злобин А.С.

Андрющенко К.С.

Жангиров Т. Р.

Санкт-Петербург

2022

Цель работы

Разработать и реализовать программу, решающую одну из оптимизационных задач (файл “Варианты”) с использованием генетических алгоритмов (ГА), а также визуализирующая работу алгоритма.

Задача

Вариант 9

Задача о кратчайшем пути (задача поиска самого короткого пути (цепи) между двумя точками (вершинами) на графе, в которой минимизируется сумма весов рёбер, составляющих путь).

Необходимо реализовать вариант о нахождении кратчайшего пути между всеми парами вершин в графе, т.е. вывести матрицу кратчайших путей

Входные данные:

- Список вершин
- Список ребер

Выполнение работы

Используемый язык программирования: Python 3.

Выполнение работы

1. Скetch GUI. Взаимодействие пользователя с программой.

Для реализации графического интерфейса будет использоваться фреймворк Kivy.

Предварительный набросок интерфейса программы:

- Основное окно программы

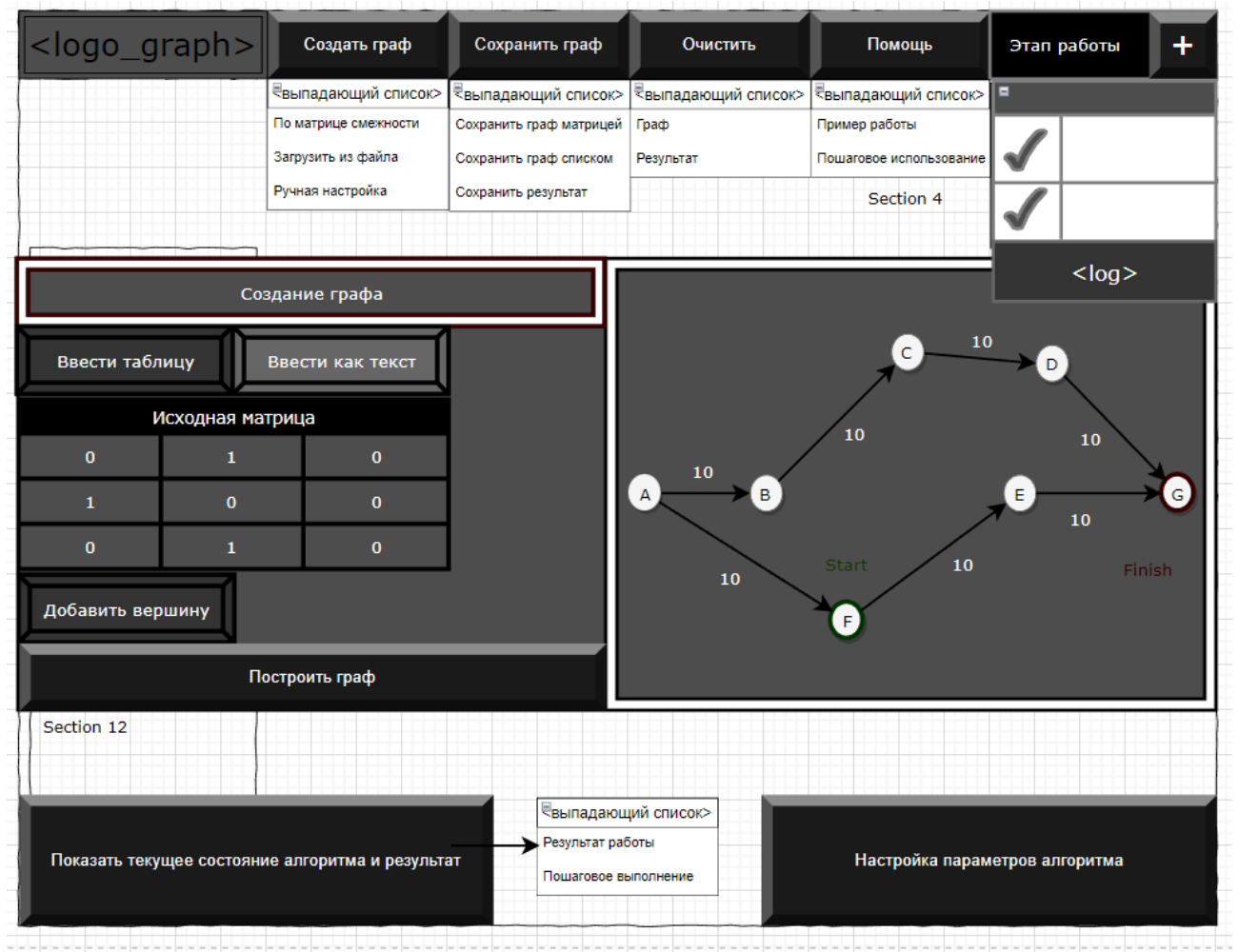


Рисунок 1 - Основное окно программы

- Меню настроек

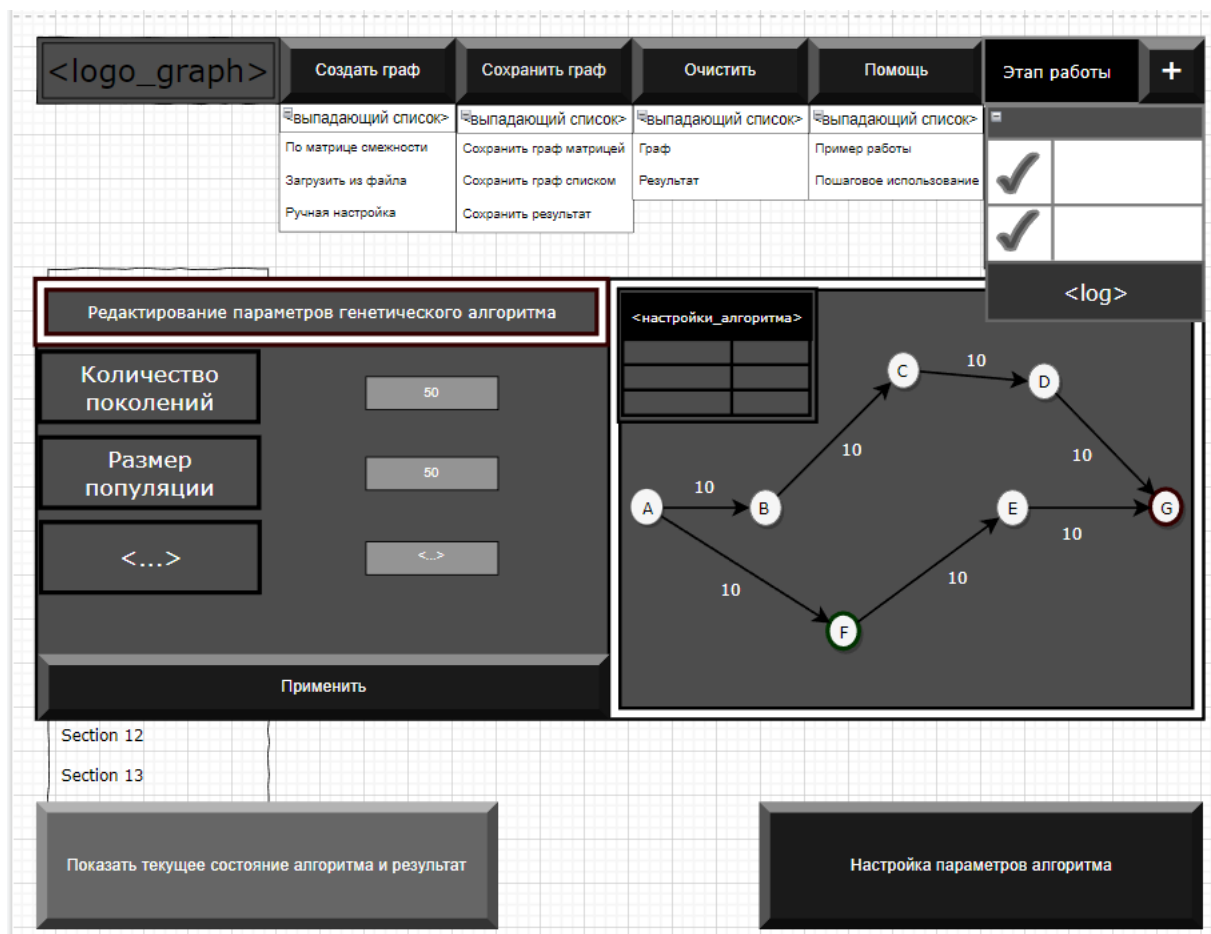


Рисунок 2 - Меню настроек параметров генетического алгоритма

- Визуализация работы алгоритма

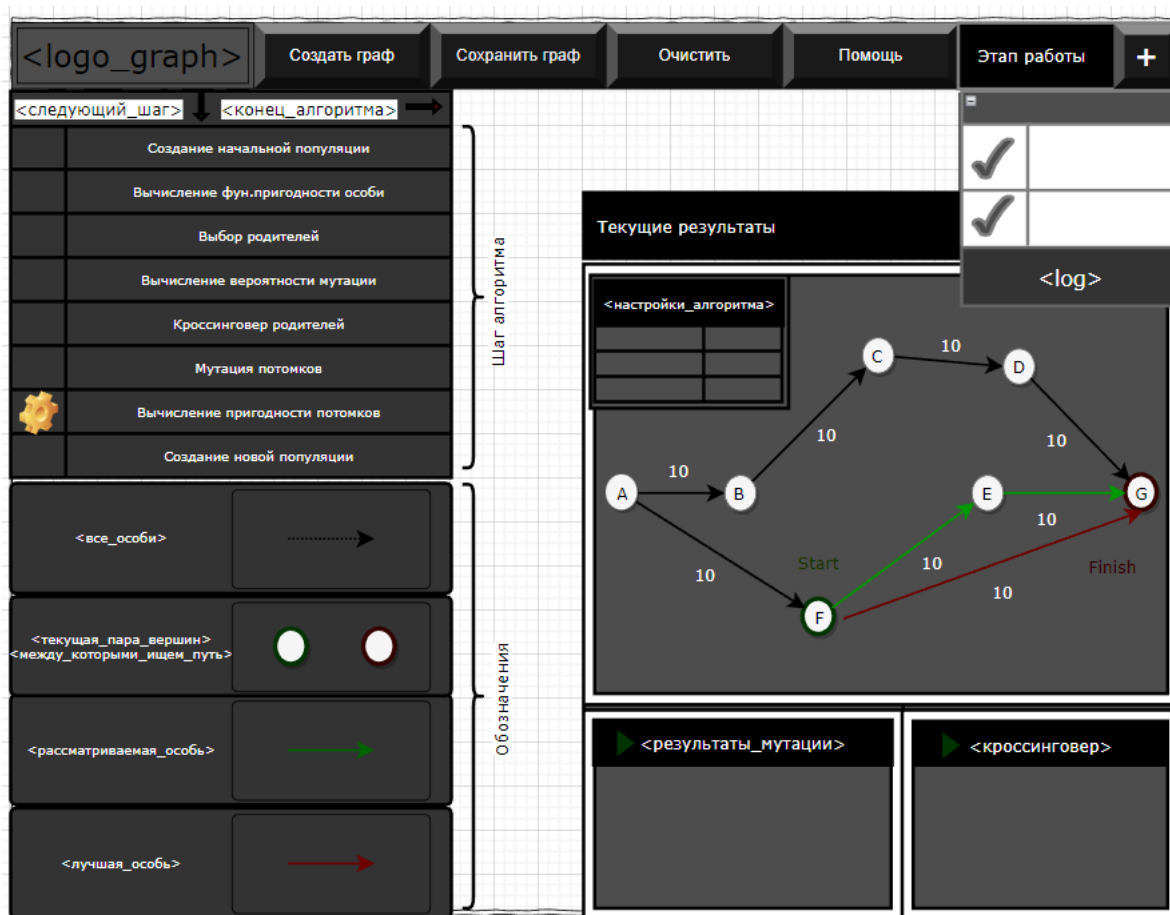


Рисунок 3 - Визуализация работы алгоритма

Описание сценария взаимодействия пользователь с программой.

«Основное окно программы»

При запуске программы открывается основное окно.

«Меню настроек»

В этом окне будут присутствовать область настроек параметров алгоритма (см. Рисунок 2), такие как размер популяции, количество поколений, вероятность кроссингвера и т. д. Эти параметры пользователю нужно будет указать (либо оставить значения по умолчанию) перед запуском алгоритма.

«Визуализация работы алгоритма»

Демонстрация работы алгоритма запускается по нажатию кнопки «показать текущее состояние алгоритма и результат». В правом окне отображаются пути между двумя вершинами, которые представляют собой

особей поколения. Ниже этого окна будут отображаться результаты мутаций и кроссинговера для текущего поколения. В левом окне находятся две кнопки для управления визуализацией алгоритма. Кнопка «Следующий шаг» переносит нас на следующий шаг алгоритма, список которых перечислен ниже. Текущий шаг отмечен соответствующим символом. Кнопка «Конец алгоритма» пропускает визуализацию и переносит нас сразу к результату выполнения алгоритма. Ниже списка шагов приведены условные обозначения, используемые при визуализации особей поколения.

«Ввод данных»

В качестве входных данных используется список вершин и список рёбер. Пользователь будет иметь возможность задать граф, для которого будут вычисляться кратчайшие пути, следующими способами:

- С помощью матрицы смежности

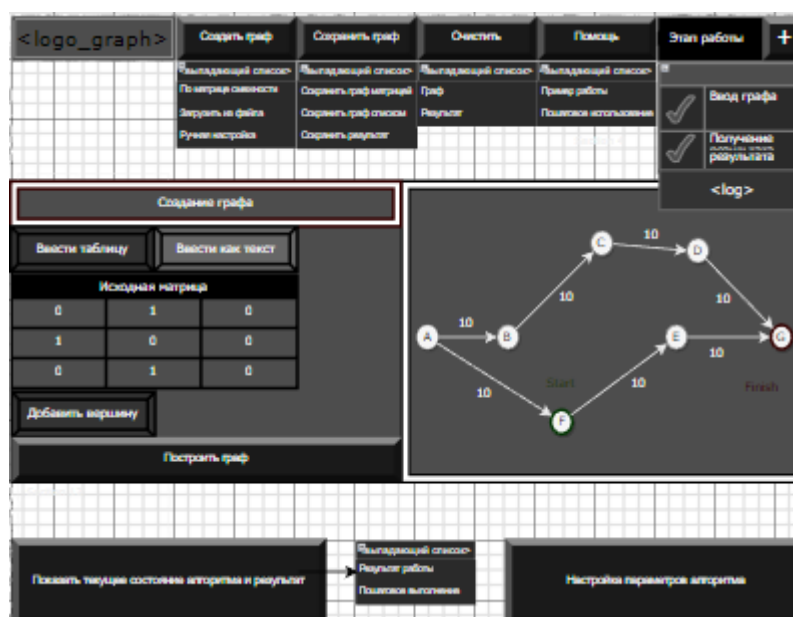


Рисунок 4 – Ввод графа с помощью матрицы смежности заполняя таблицу

Пользователь заполняет ячейки таблицы указывая вес ребра (ячейка пуста или 0 – нет пути между вершинами), в окне справа отрисовывается текущий граф, после нажатия кнопки «Построить граф». В случае нехватки ячеек, пользователь может добавить вершину через соответствующую кнопку

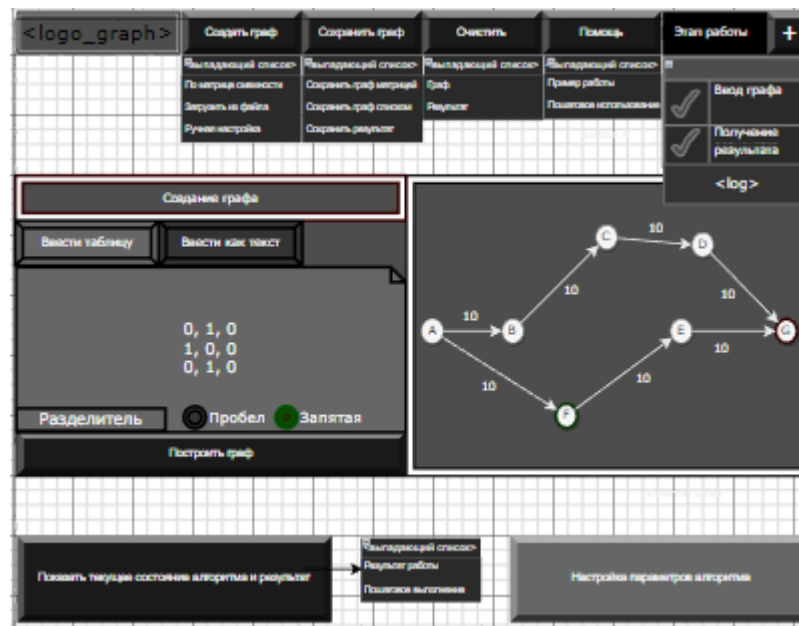


Рисунок 5 – Текстовый ввод графа с помощью матрицы смежности

Пользователь вводит матрицу в строку, указав один из предложенных разделителей между элементами матрицы.

- «Рисуя» граф в представленном окне с помощью инструментов «создать вершину», «создать направленное ребро» и «создать ненаправленное ребро».

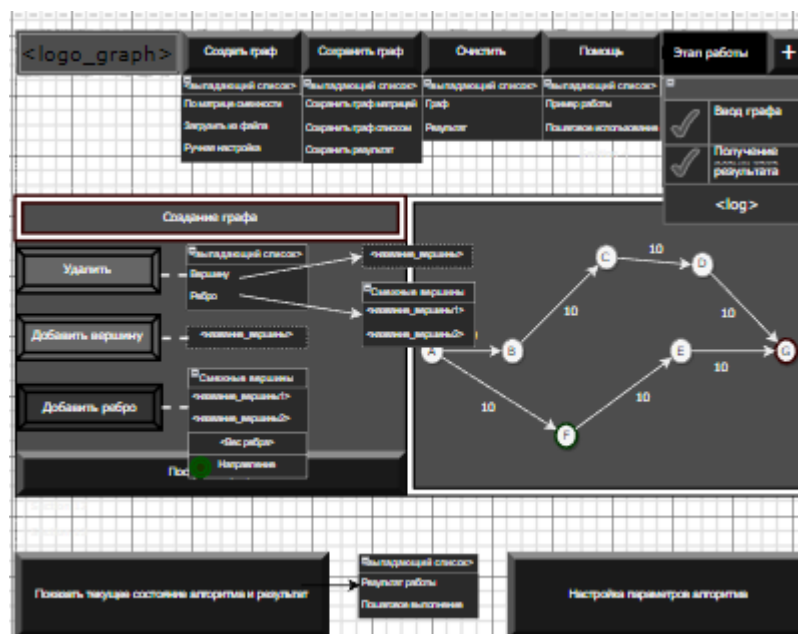
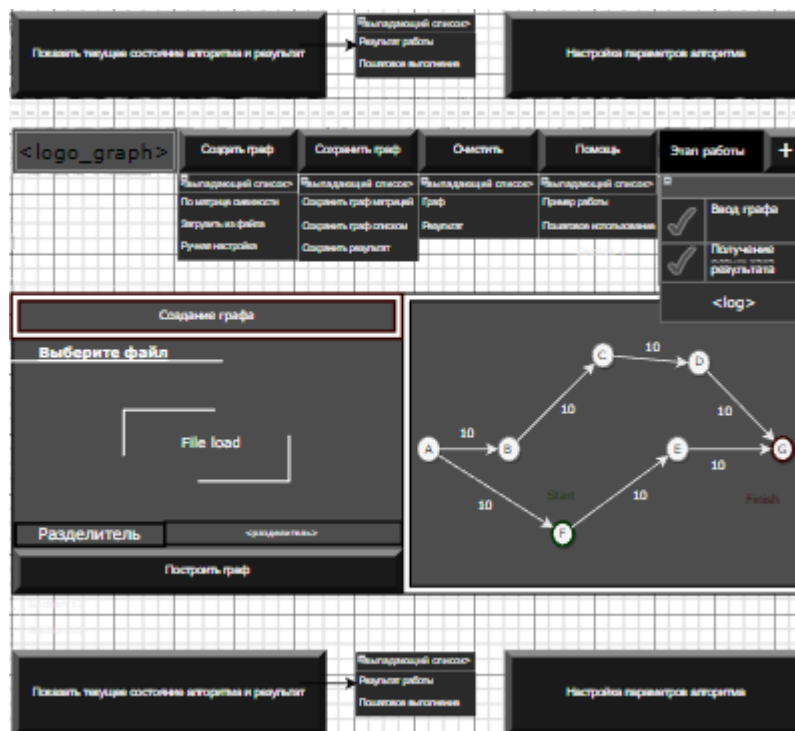


Рисунок 6 – Ручная настройка графа

После нажатия кнопки удалить появляется выпадающий список с возможными для удаления элементами. В случае удаления вершины пользователь должен ввести ее имя. При удалении ребра необходимо ввести вершину начало и вершину конец. В случае ненаправленного ребра повторить данный пункт 2 раза, изменяя порядок вершин.



Пользователю необходимо выбрать нужный файл.

В меню с выпадающими списками вверху окна можно выбрать параметры очистки полей, ввода и сохранения графа. Под кнопкой помощь можно выбрать из списка вариантов, в котором все поля будут заполнены и показан результат (пример работы), или получение набора инструкций по работе с GUI.

2. Описание генетического алгоритма.

Первым делом нам нужно формализовать задачу и определиться со способом хранения информации в хромосомах. Здесь могут быть самые разные варианты, однако остановимся на следующем. Одна хромосома описывает все возможные маршруты от исходной точки до всех остальных вершин графа.

Узлы, для удобства, пронумерованы от 0 до $n - 1$ (всего n узлов). Каждый маршрут представлен отдельным списком. Так как длина пути изначально неизвестна, то он заканчивается тогда, когда встречается вершина с номером назначения.

Для хранения графа будем использовать матрицу смежности, которая на пересечении строк и столбцов содержит значения весов дуг между соответствующими вершинами. Те клетки, что не содержат значений, означают отсутствие связей. По идее, здесь можно прописать бесконечные величины, означающие бесконечно длинные маршруты.

Следующим шагом в программе нужно определить, как вычислять приспособленность отдельных особей, как выполнять скрещивание и мутации. Начнем с функции приспособленности. Она должна возвращать кортеж значений, которые являются суммой длин путей до вершин графа.

В функции мутации перебираем списки маршрутов особи и для каждого выполняем перемешивание индексов с некоторой заданной вероятностью.

В качестве функции скрещивания будем использовать алгоритм упорядоченного скрещивания.