

**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

Ильин Д.А.  
Морева Е.С.  
Смирнов И.А.  
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 10**

Задача о раскраске графа (задача поиска раскраски графа с наименьшим кол-вом цветов).

Необходимо реализовать программу, которая для заданного графа выдаёт наименьшее кол-во цветов, в которые можно покрасить весь граф.

Входные данные:

- Список вершин
- Список ребер

## **Выполнение работы**

Используемый язык программирования: Python 3.

## **Выполнение работы(Итерация 2)**

### **1. GUI**

На данный момент реализована основа приложения, сделаны нужные кнопки, так же реализовано поле результата и рабочая область, в которой будет находиться граф(Рисунок 1).

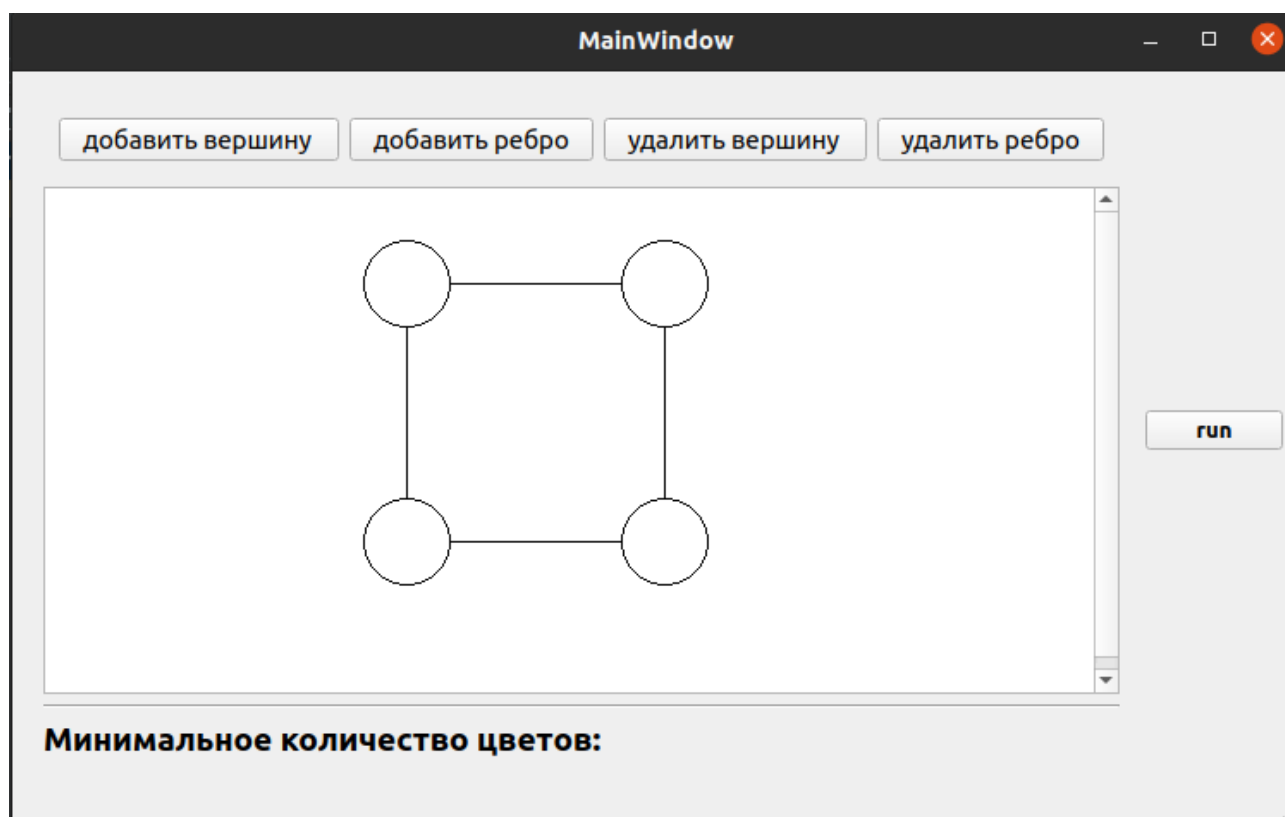


Рисунок 1.

## 2. Генетический Алгоритм

Реализован класс графа, для хранения заданных вершин и рёбер. Класс раскраски( Coloring) — отвечающий за определённую раскраску, элемент данного класса является хромосомой относительно реализуемого ГА, если говорить в терминах ГА. В классе раскраски создана функция проверяющая возможность данной раскраски на графе.

Выбрана островная модель ГА, которая является наиболее распространенной моделью параллельного ГА. Ее суть заключается в том, что популяция, как правило состоящая из очень большого числа особей, разбивается на одинаковые по размеру подпопуляции. Каждая подпопуляция обрабатывается отдельным процессором с помощью одной из разновидностей непараллельного ГА. Изредка, например, через пять поколений, подпопуляции будут обмениваться несколькими особями. Такие миграции позволяют подпопуляциям совместно использовать генетический материал.

Данная модель позволяет уменьшить время вычисления нужной хромосомы для большого количества вариаций хромосом, что как раз происходит в нашей ситуации.

Реализованы две функции:

- `create_coloring` — создаёт одну хромосому(раскраску в данном случае), элемент класса `Coloring`.
- `create_population` — создаёт популяцию, состоящую из хромосом. Нужна для начала работы алгоритма.

Методом отбора было решено выбрать комбинацию элитарного и турнирного отборов в процентном соотношении 10% и 90% соответственно.

После чего написана основа ГА. Далее будет описание его работы.

Алгоритм работает в цикле для различного кол-ва цветов, изначально кол-во цветов берётся по количеству вершин. Для каждого кол-ва цветов ищется нужная хромосома(такая, что раскраска возможна), посредством выбранного отбора, мутации и скрещивания создаются новые потомки. Если нужная хромосома найдена, то переходим к количеству меньшему на единицу нежели то, что было до этого, иначе цикл заканчивается при появлении слишком большого кол-ва потомков и возвращается последнее число, для которого была найдена раскраска.

3. Инструкция по сборке и запуску находится в файле на репозитории в папке `/doc/`, под названием - «Инструкция по сборке и запуску».