

**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 1.**

1. Выбор метода решения задачи.

Будем начинать с раскраски в  $k$  цветов ( $k$  - количество вершин), наши хромосомы (представители) - вектора длиной  $k$  с числами от 1 до проверяемого числа цветов (собственно раскраска, на 3 месте стоит 4 - вершина с индексом 3 покрашена в четвёртый цвет) которые мы скрещиваем и мутируем какое-то количество поколений и если получаем сходимость 0 (граф раскрашен), то переходим к числу цветов меньшего, нежели предыдущее на 1. Если спустя некоторое кол-во поколений сходимость не получили, то делаем вывод, что последнее значение, для которого сходимость нашли, является минимальным.

Первично выбранные параметры ГА:

- вероятность кроссинговера - 90%
- вероятность мутации - 0.5%
- размер популяции - 30 особей
- способ отбора - совмещение элитарного отбора и отбора с усечением
- кодирование особей - список с номерами цветов для вершин с номером, равным индексу цвета в списке, цвета в списке имеют номера порядка, в котором встречаются (то есть каждая раскраска задана однозначно, обмен цветов номерами не учитывается)
- значение приспособленности - количество правильно окрашенных ребер, при приспособленности равной числу рёбер достигнута правильная раскраска
- модель ГА — островная

## 2. Формирование прототипа GUI.

На рисунке 1 представлен вид программы после запуска. Пользователь может выбрать одну из следующих команд:

- 1) Создать вершину — при выборе этой команды, наведя на рабочую область(под панелью инструментов, далее — рабочая область) можно создать вершину, она появится в месте нажатия.
- 2) Удалить вершину — удаляет вершину, которую выберет пользователь, посредством нажатия на неё.
- 3) Создать ребро — создаёт ребро между двумя выбранными вершинами.
- 4) Удалить ребро — работает аналогично удалению вершины.

Результат выполнения команд выше представлен на рисунке 2.

Так же рабочую область можно приблизить или отдалить, для этого служат две кнопки с лупами, с плюсом приближает, с минусом отдаляет.

Зелёный треугольник запускает ГА для заданного на данный момент графа, находящего в рабочей области.

Зелёный жучок является кнопкой запуска с более подробными переходами. Данная кнопка вызывает окно(Рисунок 3), в котором выбирается шаг, с которым будет выводиться раскраска(количество всех раскрасок обработанных программой может быть очень большим и дабы иметь возможность не выводить все промежуточные раскраски вводится шаг).

Промежуточный результат работы программы представлен на рисунке 4.

В самом низу выводится результат — минимальное кол-во цветов для раскраски графа.

После работы программы вершины окрашиваются в определённую раскраску подходящую требованиям задачи и содержащую выведенное кол-во цветов(Рисунок 5).



Рисунок 1.

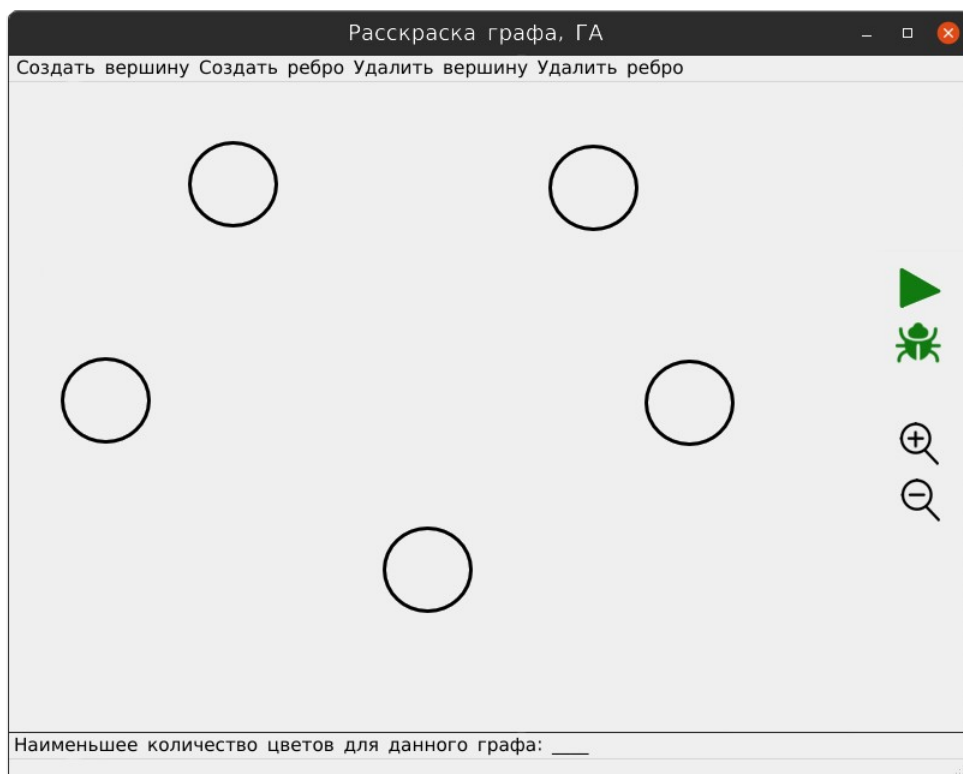


Рисунок 2.

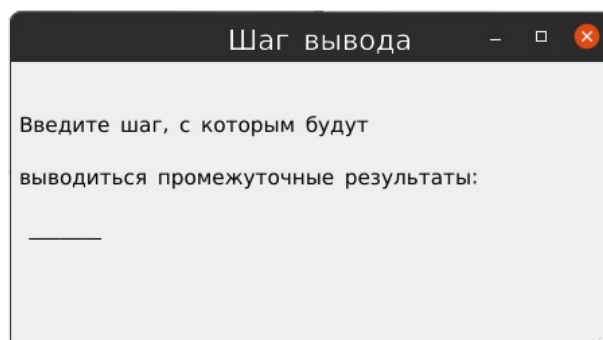


Рисунок 3.

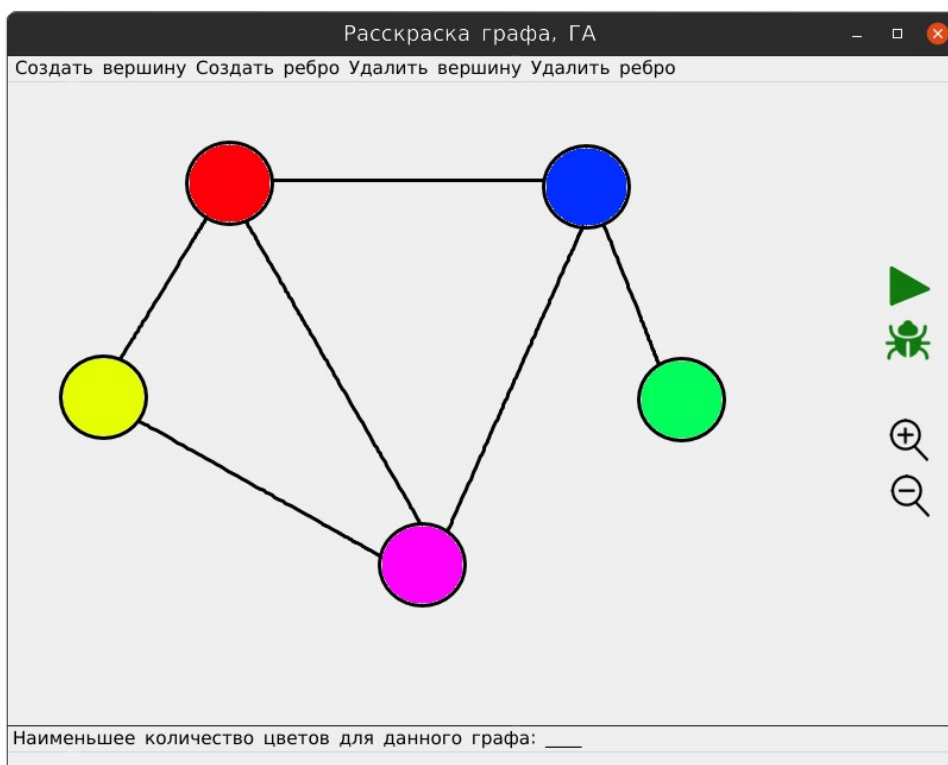


Рисунок 4.

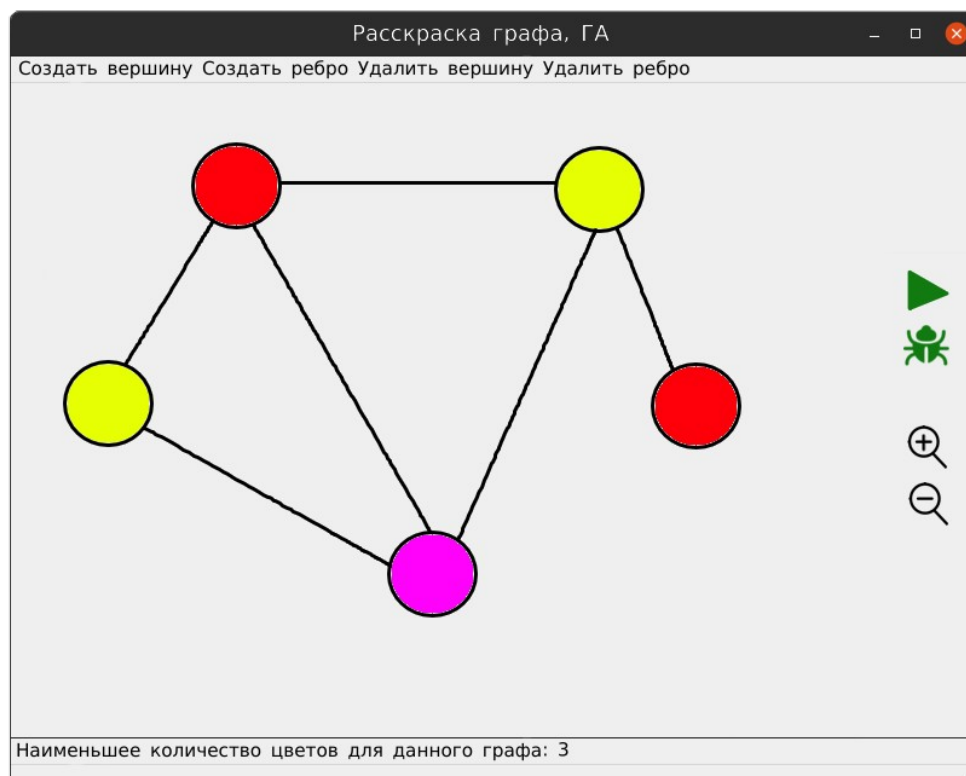


Рисунок 5.

## 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).

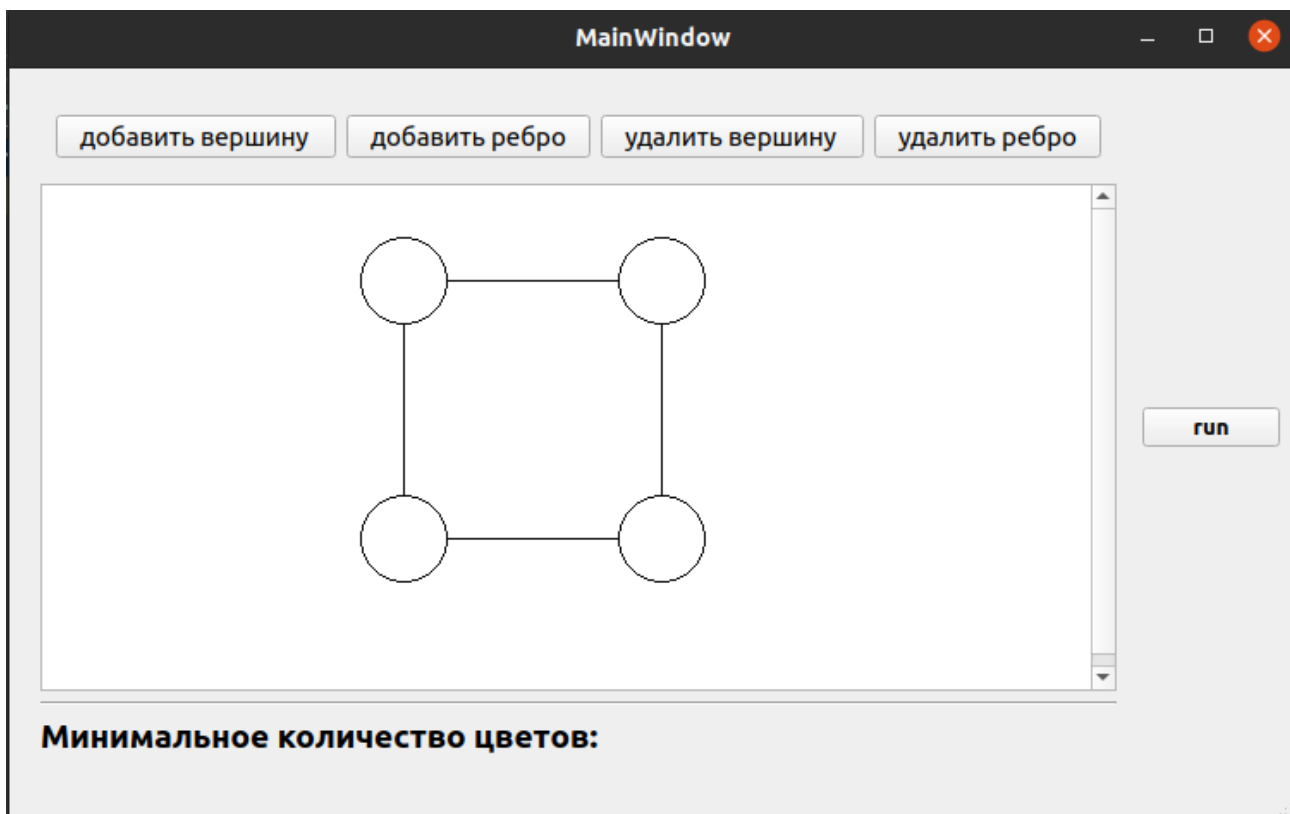


Рисунок 1.

### 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, так же данная функция высчитывает приспособленность отдельной хромосомы, зависящей от правильности раскраски ребра.

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, если поколение превышает некоторую константу, 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».



### Итерация 3.

#### 1. GUI

Реализованы следующие элементы GUI:

- 1) Добавление ребра
- 2) Удаление ребра
- 3) Добавление вершины
- 4) Удаление вершины

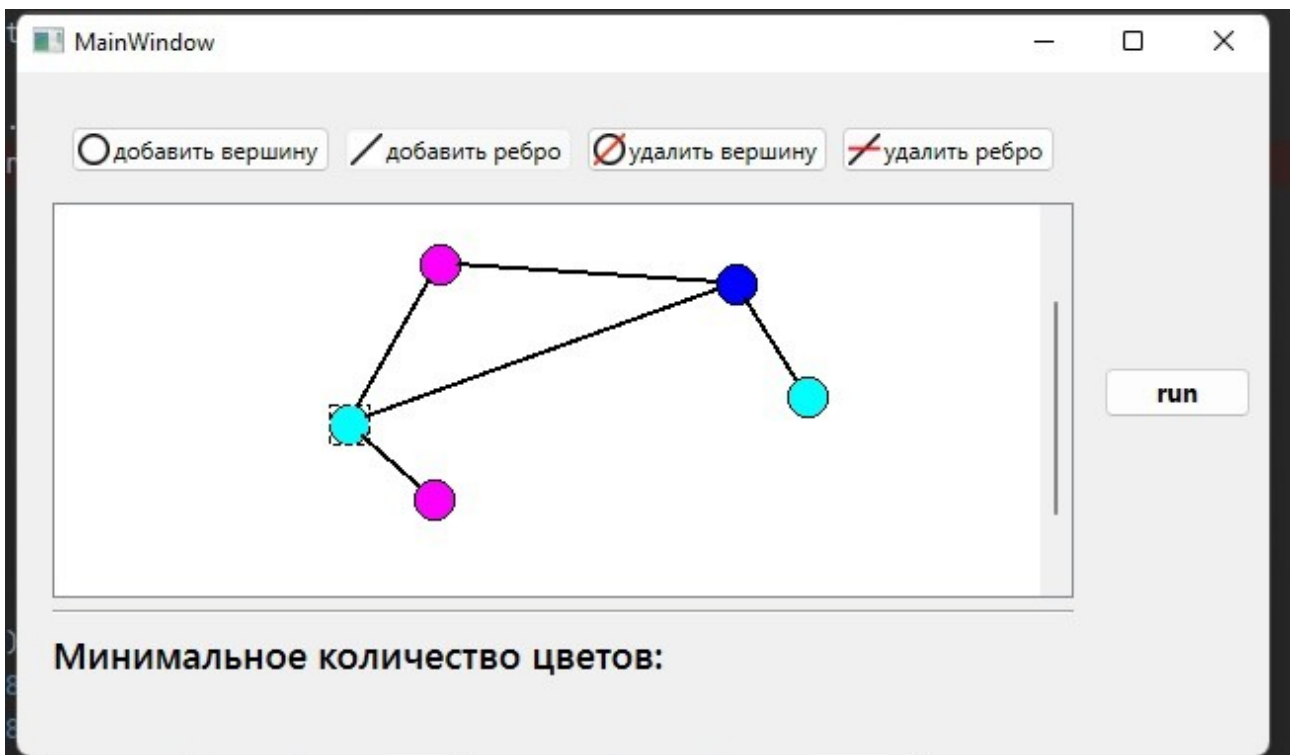


Рисунок 1.

На данный момент на отдельном примере реализовано подключение к ГА.

#### 2. Генетический Алгоритм

Окончательно реализован алгоритм.

Добавлены на данной итерации функции: отбора, кроссинговера, скрещивания и мутации(параметры функций указаны выше).