

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 — вид программы после запуска.

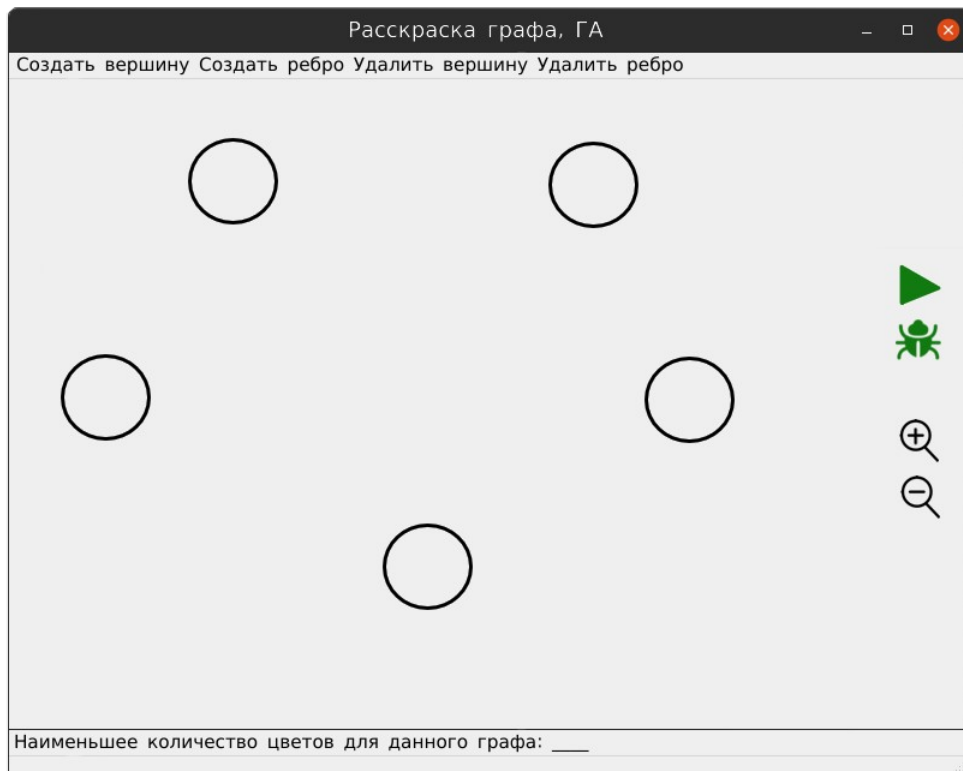


Рисунок 2 — результат использования кнопок панели инструментов.
(создание графа)

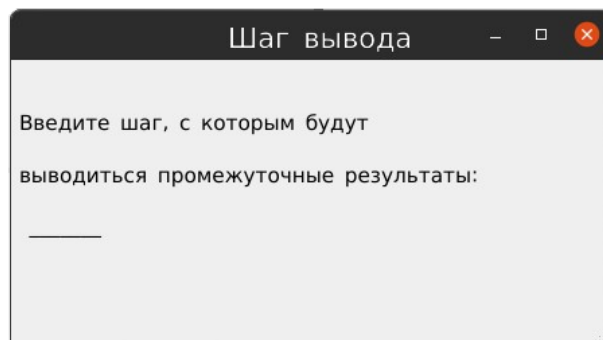


Рисунок 3 — дополнительное окно для подробного вывода.

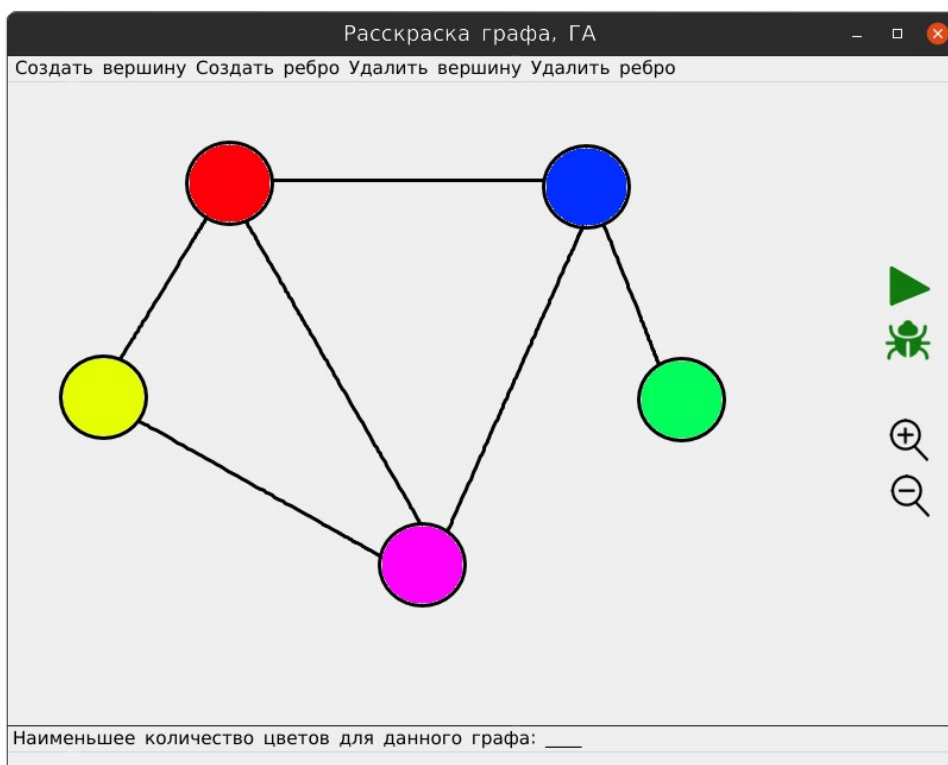


Рисунок 4 — промежуточный результат работы программы.

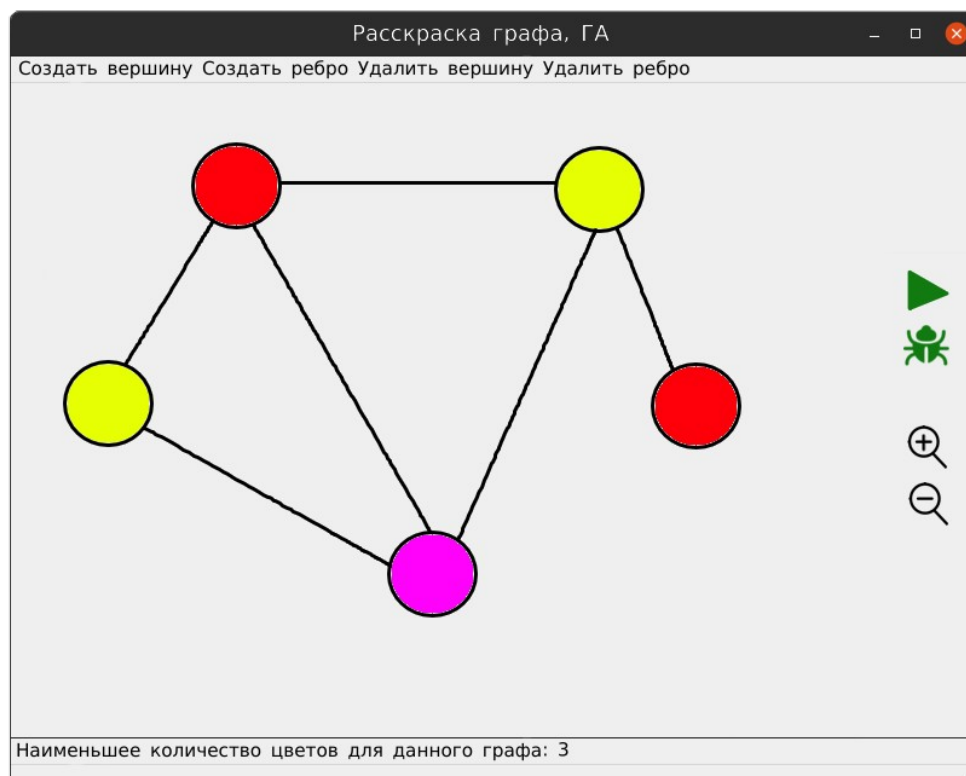


Рисунок 5 — положение после выполнения ГА на графе.

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 6).

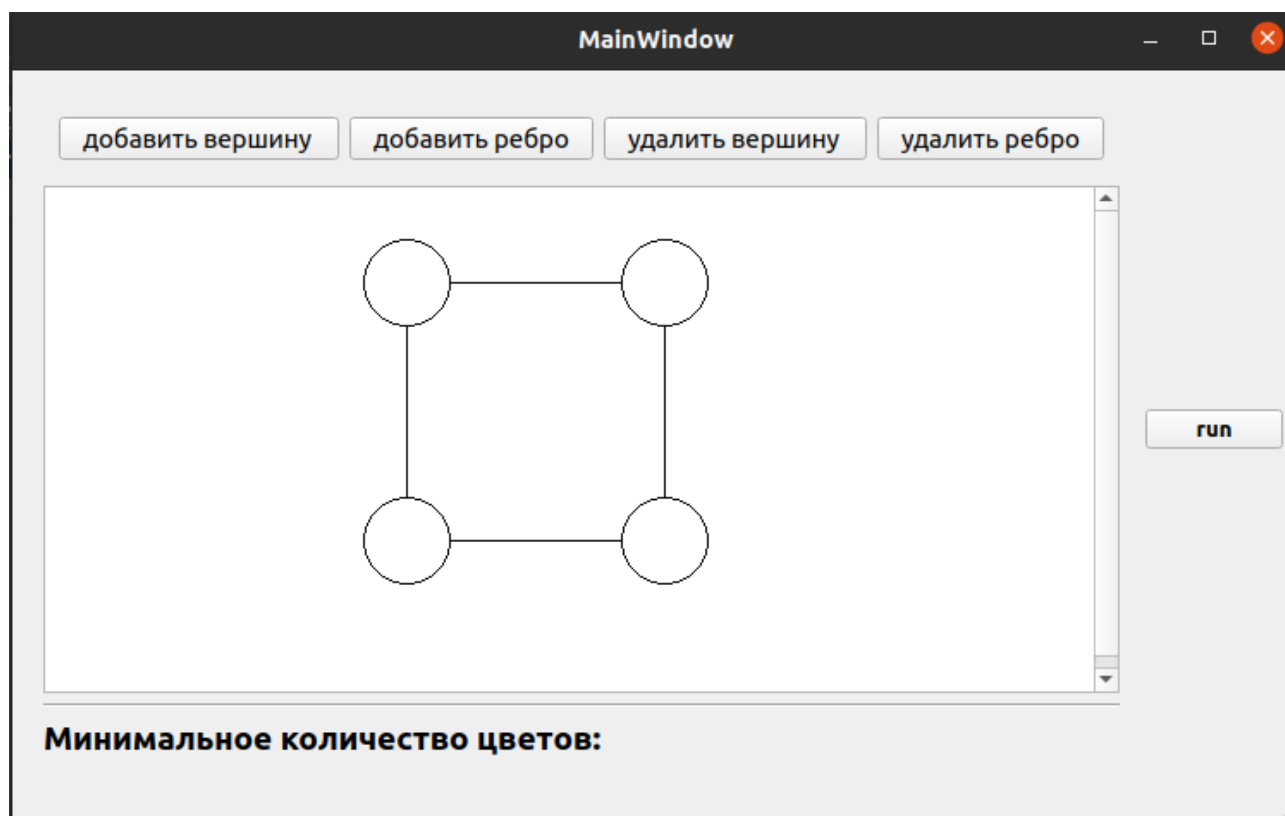


Рисунок 6 — реализованная часть GUI на момент итерации 2.

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

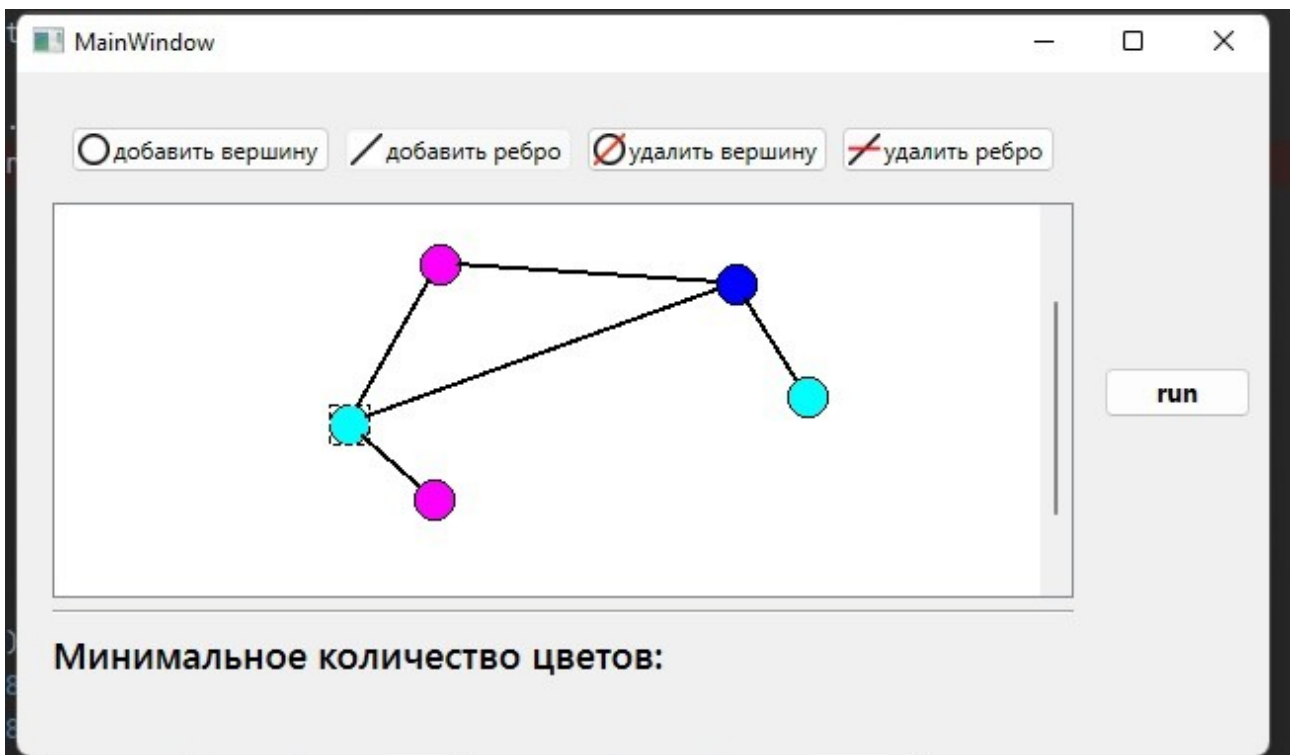


Рисунок 7 — GUI с подключённым ГА.

На данный момент на отдельном примере реализовано подключение к ГА (Рисунок 7).

2. Генетический Алгоритм

Окончательно реализован алгоритм.

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

Итерация 4.

Окончательно реализован GUI, в соответствии с предоставленными скетчами, однако некоторые моменты были изменены. Подключён полностью функционирующий ГА.

Изменения GUI относительно скетчей:

1. Кнопка вызова окна подробного вывода удалена.
2. В верхний правый угол добавлена флаговая кнопка(check box) на панель инструментов — «эволюция», отвечающая за подробный вывод, справа от данной кнопки находится поле для ввода шага, описанного в первой итерации.
3. Изменена кнопка запуска и перенесена в нижний правый угол.
4. Убраны возможность приближения и отдаления рабочей области и соответствующие кнопки за ненадобностью.

Ниже предоставлены изображения реализованной программы(Рисунок 8 - 11). Очерёдность в соответствии с очерёдностью изображений первой итерации(скетчами), за исключением дополнительного окна(Рисунок 3).

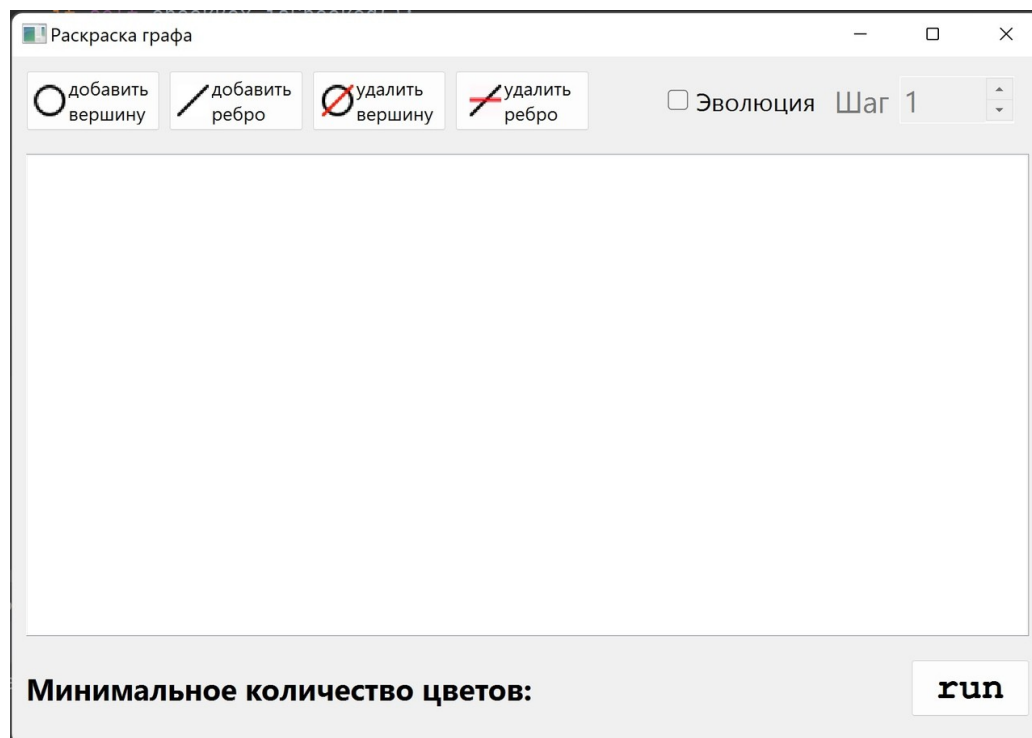


Рисунок 8 — вид программы после запуска.

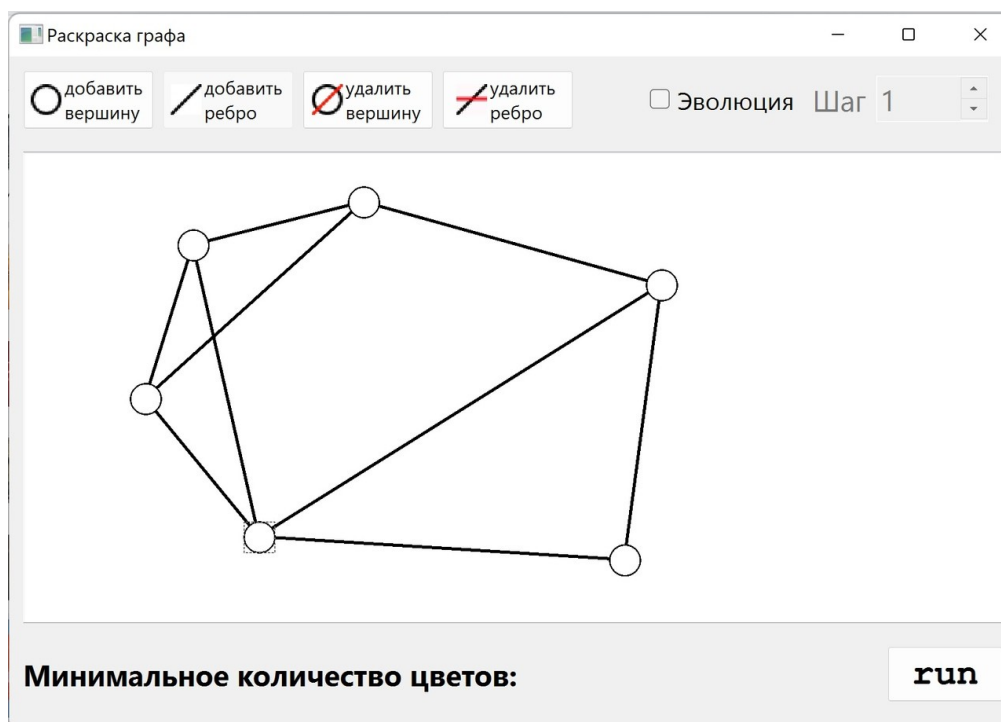


Рисунок 9 — результат использования кнопок панели инструментов.
(создание графа)

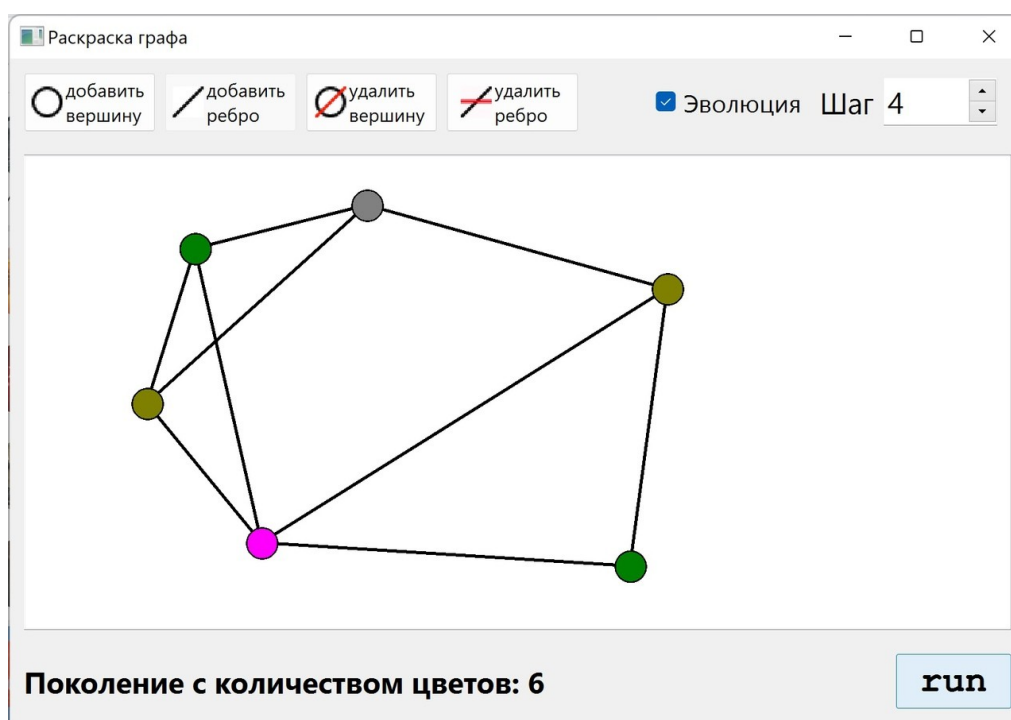


Рисунок 10 — промежуточный результат работы программы.

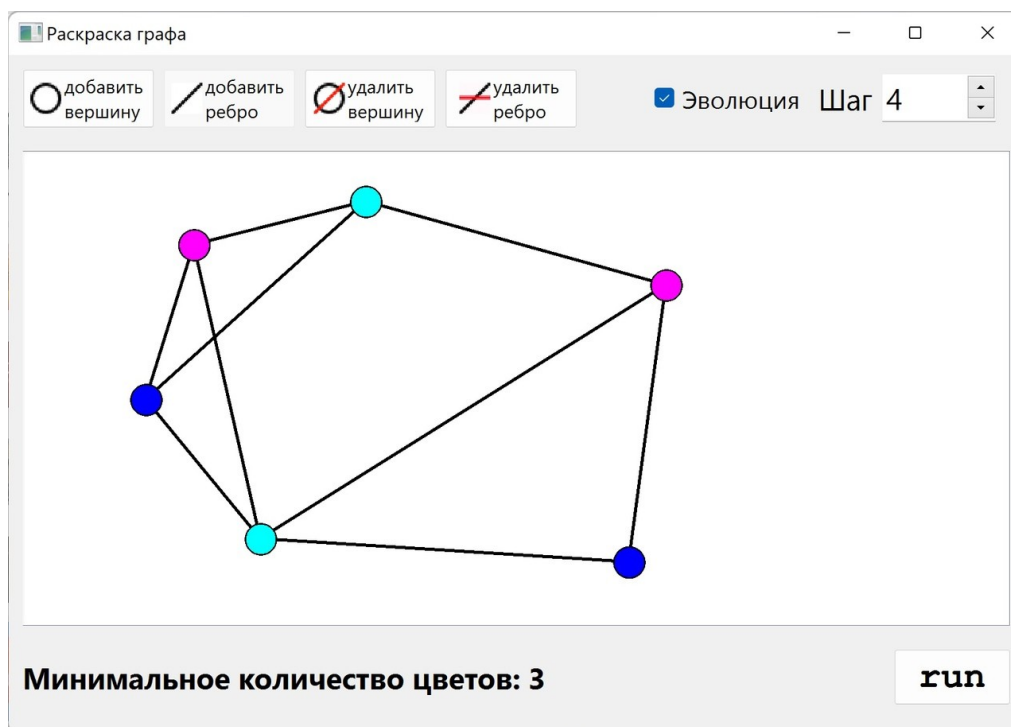


Рисунок 11 — положение после выполнения ГА на графе.

Генетический алгоритм был протестирован, результаты тестирования представлены в таблице 1.

ГА тестировался отдельно от GUI, предоставлены матрица смежности и данные полученные после обработки генетическим алгоритмом.

Номера вершин	Матрица смежности				Раскраска (номер вершины: номер цвета)	Результат	Пояснение
1) 0	0				0: 0	Верный	Проверка для единственной возможной хромосомы
2) 0 1	0	1	1	0	0: 1 1: 0	Верный	Проверка для малого кол-ва вершин
3) 2	0	1	0	1	2: 0	Верный	Проверка для

4 8 6	<table><tr><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>	1	0	1	1	0	1	0	1	1	1	1	0	4: 2 8: 0 6: 1		переименован ных вершин																																																																																																																																																																																																																																				
1	0	1	1																																																																																																																																																																																																																																																	
0	1	0	1																																																																																																																																																																																																																																																	
1	1	1	0																																																																																																																																																																																																																																																	
4) 0 1 2 3 4	<table><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	0: 0 1: 1 2: 2 3: 3 4: 4	Верный	Проверка для полного графа из 5-ти вершин (полные графы — крайние случаи)																																																																																																																																																																																																																							
0	1	1	1	1																																																																																																																																																																																																																																																
1	0	1	1	1																																																																																																																																																																																																																																																
1	1	0	1	1																																																																																																																																																																																																																																																
1	1	1	0	1																																																																																																																																																																																																																																																
1	1	1	1	0																																																																																																																																																																																																																																																
5) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	<table><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0: 1 1: 0 2: 1 3: 0 4: 1 5: 0 6: 1 7: 0 8: 1 9: 0 10: 1 11: 0 12: 1 13: 0 14: 1 15: 0	Верный	Проверка для графа являющегося циклом.
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1																																																																																																																																																																																																																																					
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																					
0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																					
0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																					
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																					
0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																					
0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																					
0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0																																																																																																																																																																																																																																					
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																					
0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0																																																																																																																																																																																																																																					
0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0																																																																																																																																																																																																																																					
0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0																																																																																																																																																																																																																																					
0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0																																																																																																																																																																																																																																					
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0																																																																																																																																																																																																																																					
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0																																																																																																																																																																																																																																					

6) 0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0: 0	Верный	Проверка для графа, у которого все вершины соединены со всеми остальными вершинами, помимо некоторой (случай близкий к крайнему)
1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1: 3		
2	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2: 3		
3	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3: 1		
4	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	4: 1		
5	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	5: 7		
6	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	6: 0		
7	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	7: 7		
8	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	8: 5		
9	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	9: 5		
10	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	10: 4		
11	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	11: 4		
12	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	12: 6		
13	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	13: 6		
14	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	14: 2		
15	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	15: 2		
	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1			
	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1			
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0				
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0				

7) 0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0: 9	Верный	Проверка для полного графа из 15-ти вершин, плюс одна отдельная вершина (крайний случай для графа из 15-ти вершин)
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1: 1		
2	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	2: 14		
3	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	3: 11		
4	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	4: 8		
5	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	5: 3		
6	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	6: 6		
7	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	7: 4		
8	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	8: 10		
9	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	9: 12		
10	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	10: 13		
11	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	11: 2		

12	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	12: 7		
13	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	13: 0		
14	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	14: 5		
15	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	15: 0		
	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0			
	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0			
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0			
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

Результат

Была исследована работа ГА, реализован ГА для раскраски графа с представленными выше характеристиками. Для упрощения работы с ГА был создан GUI по заранее образованному скетчу.

В ходе реализации генетического алгоритма надёжно себя показала непараллельная модель с совмещением элитарного и турнирного отборов. Выбранная реализация хранения данных об особях оказалась корректной, введение однозначности раскраски практически не влияет на результат работы, но замедляет алгоритм.

Для создания GUI был изучен PySide6 и освоено приложение Qt Designer, реализованы все заявленные варианты взаимодействия с интерфейсом: построение графа с помощью кликов мыши по окну и соответствующих кнопок, полосы прокрутки, возможность визуализировать работу алгоритма анимацией и задать шаг прохождения по поколениям.

Ссылки на использованные материалы:

- <https://drive.google.com/drive/folders/1b5EUEkGA2X7t-G1kg7P4fLbeIRU-Hm-Y>
- <https://www.geeksforgeeks.org/project-idea-genetic-algorithms-for-graph-colouring/>
- <https://www.youtube.com/playlist?list=PLA0M1Bcd0w8zkG8DJSyqrKyBfDtRj2Ohh>
- <https://wiki-qt-io-staging.herokuapp.com/Main>
- <https://www.pythonguis.com/tutorials/pyside-signals-slots-events/>