

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

**ОТЧЕТ**  
**по учебной практике**  
**Тема: Генетические алгоритмы (Вариант 8).**

Студент гр. 3381

Гладилин Е.О.

Студент гр. 3388

Лутфулин Д.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2025

### **Задание.**

Дано N точек в двумерном евклидовом пространстве. Необходимо найти координаты левого нижнего угла и длины сторон для M (задается пользователем) квадратов такие, чтобы квадраты суммарно содержали максимальное количество точек внутри и не пересекались между собой.

### **Выполнение работы.**

#### **2 этап:**

Лутфулин – план решения

*Ограничения на решения:*

1. Все решения алгоритма (и в начальной популяции, и далее) должны состоять из непересекающихся квадратов. Если какие-то квадраты в решении пересеклись, пытаемся устранить проблему изменением размеров квадратов.

Если квадраты пересеклись – пытаемся решить проблему изменением размера одного квадрата, другого квадрата, возможно попробовать уменьшить оба

2. Нежелательно появление квадратов, в которые не попала ни одна точка. Будем корректировать это штрафом в функции приспособленности на решения с такими квадратами.

*Начальная популяция:* узнаём диапазон, в котором нужно сгенерировать квадраты: находим среди всех точек самую левую, правую, верхнюю, нижнюю координаты. Увеличиваем эту область на 10% по ширине и высоте (иначе крайние точки почти никогда не попадут ни в какой из квадратов).

*Функция приспособленности:* очевидно, она должна зависеть от количества точек, покрытых квадратами. Но также хочется, чтобы решение не сводилось к тому, что один квадрат покроет все точки а остальные будут пустыми. Тогда будем штрафовать квадраты, в которые не попали точки:

$$F_1 = \text{Кол} - \text{во точек} * \left(1 - \frac{\text{кол} - \text{во пустых квадратов}}{M}\right)$$

Можно попробовать немного модифицировать данную функцию:

$$F_2 = \sum_1^m \text{точек в } i\text{-ом квадрате} * \left(1 - \frac{\text{площадь } i\text{-го квадрата}}{\text{площадь диапазона}}\right)$$

Если квадрат большой, а все точки, которые он покрывает, расположены (например) в его центре, то это решение будет при всех прочих равных считаться менее приспособленным по данной функции. Тогда при скрещивании при прочих равных будут выбираться более «плотные» квадраты, которые не будут мешать своим большим размером квадратам, добавляемым при скрещивании.

*Формат данных:*

Класс точки будет содержать в себе только координату

Класс квадрата будет хранить в себе его координаты, сторону и список точек, которые он покрывает. Так как точки фиксированы, это можно вычислять прямо в конструкторе.

*Хромосома* в таком случае просто будет состоять из массива квадратов (или троек чисел, из которых они состоят). Ген – один квадрат

*Функция отбора:* планируется взять правило рулетки и ранжированный отбор, возможно с элитарным

*Мутация* – подвинуть координату квадрата, изменить его размер.

Скрещивание можно попробовать равномерное, двухточечное

Гладилин – прототип интерфейса

Графический интерфейс.

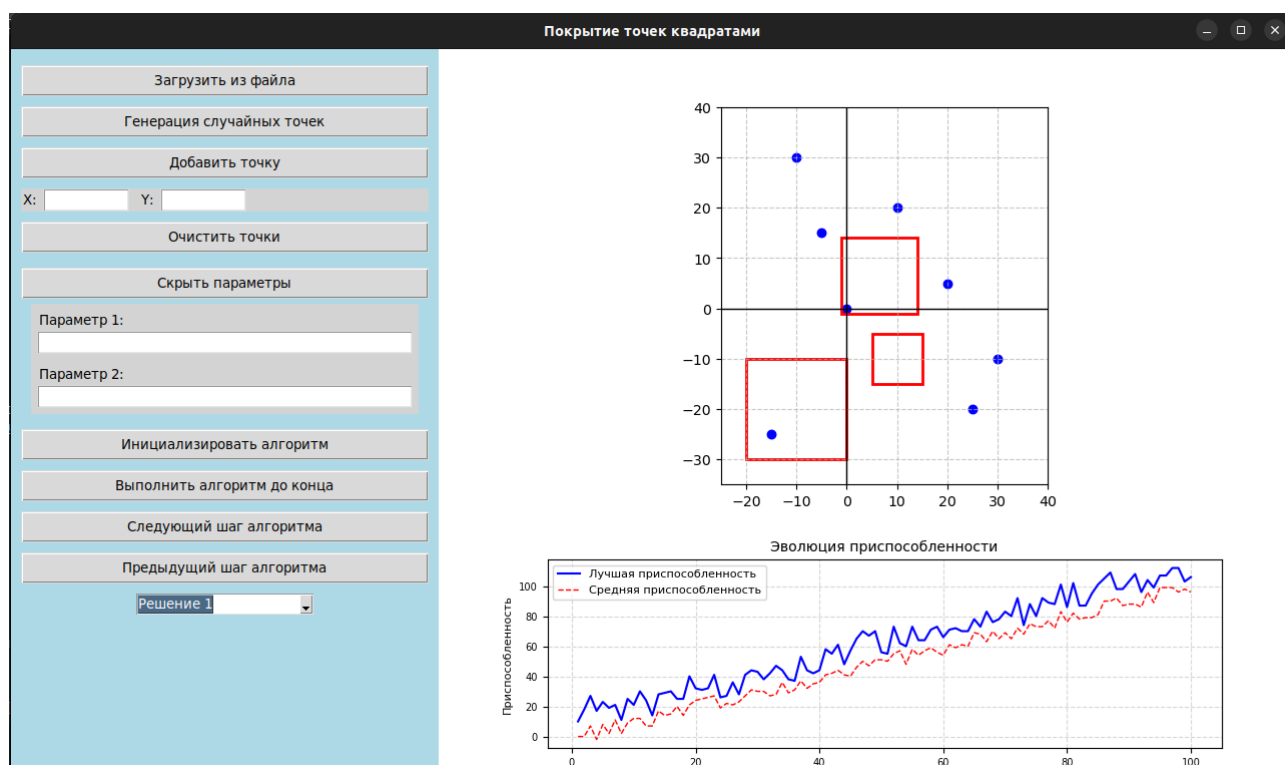


Рисунок 1 — Прототип графического интерфейса

Экран приложения содержит изображения полученного решения, график приспособленности и кнопки взаимодействия пользователя с программой.

Данные могут быть загружены из файла, сгенерированы случайно или добавлены через GUI. Кнопка «Очистить точки» убирает все загруженные точки.

Кнопка «Показать параметры» открывает поля для ввода параметров генетического алгоритма (в полях введены начальные значения). Параметры можно скрыть.

Кнопка «Инициализировать алгоритм» применяет для заданных точек алгоритм с текущими параметрами. Оставшиеся кнопки позволяют сделать шаги алгоритма. После инициализации алгоритма поменять данные нельзя.

Последний элемент — это выпадающий список, в котором можно выбрать отображение необходимого решения.

### 3 этап

Гладилин — окончательный дизайн GUI.

При запуске программы пользователю доступны кнопки загрузки данных, изменения параметров и инициализации генетического алгоритма. Остальные кнопки неактивны, пока не инициализирован алгоритм. Чтобы новые введенные параметры были применены, необходимо нажать на кнопку «Инициализировать алгоритм».

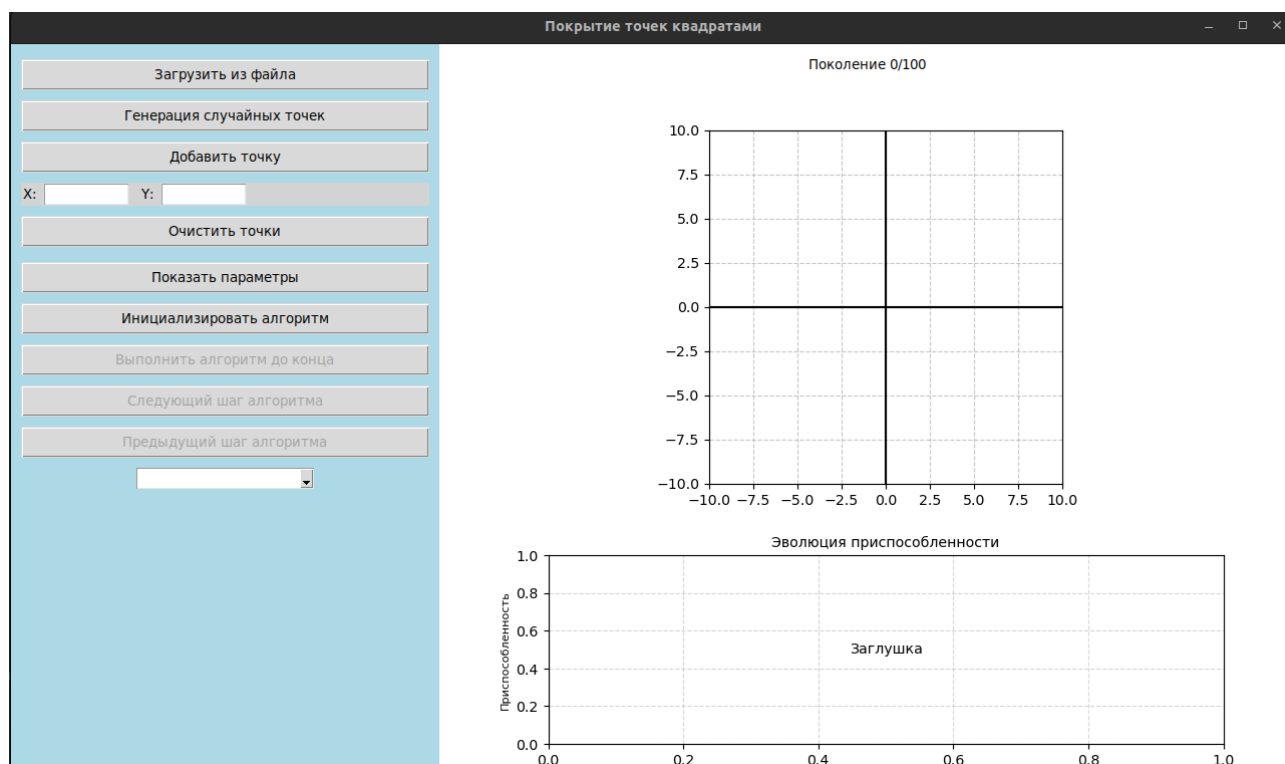


Рисунок 2 — Вид приложения при запуске.

Данные для программы можно загрузить через файл (координаты на каждой строки через пробел). Сгенерировать случайно, выбрав диапазон, или добавить через графический интерфейс.

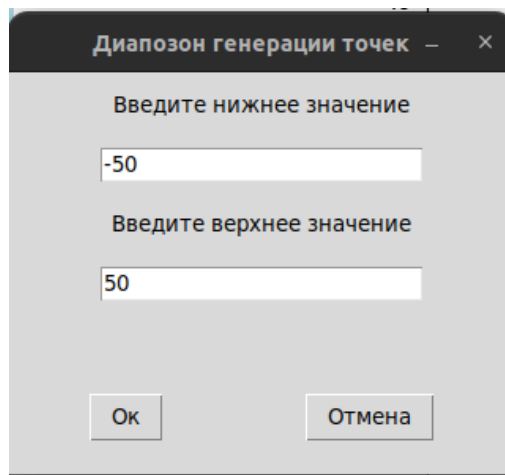


Рисунок 3 — Окно выбора диапазона генерации случайных точек.

Во время работы алгоритма обновляется текущее поколение, изображение решения и график приспособленности.

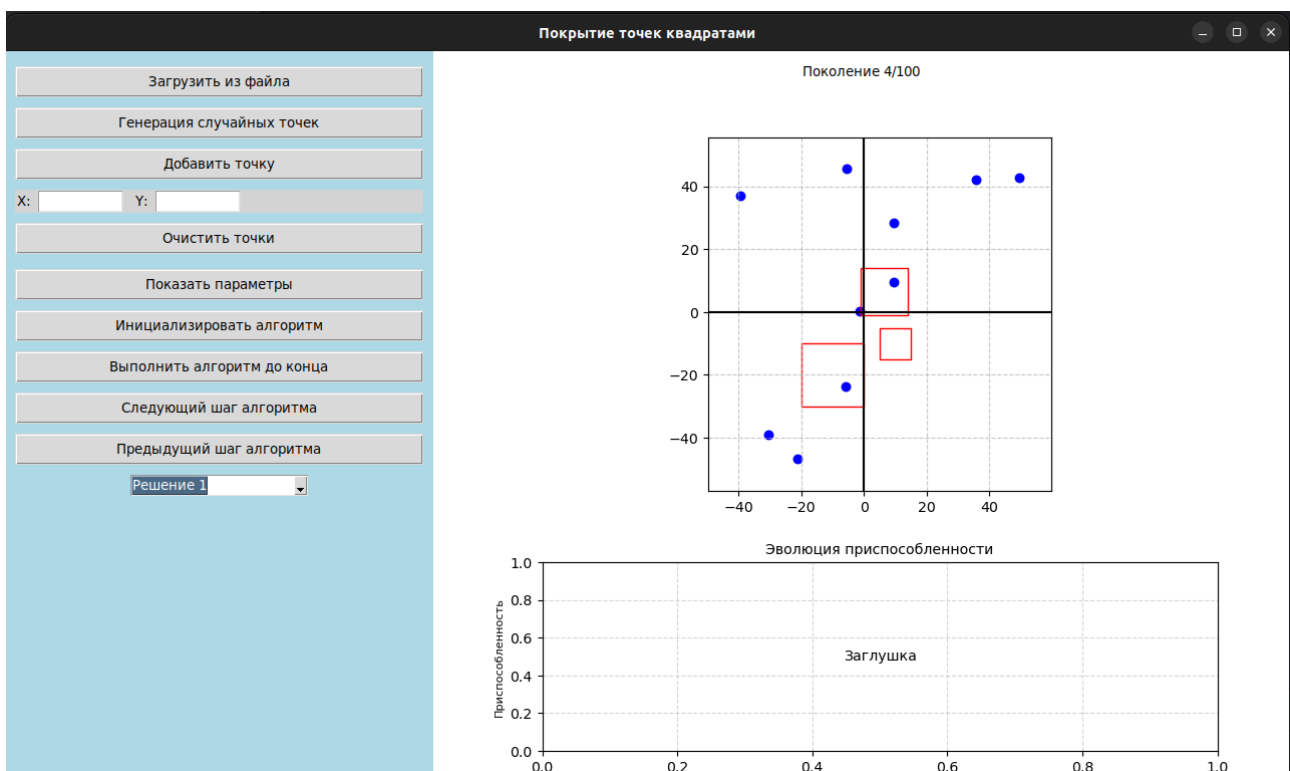


Рисунок 4 — Демонстрация при запуске алгоритма

Лутфулин – генетический алгоритм:

Написаны классы точки, квадрата, популяции и генетического алгоритма.

Point - содержит координаты

Square – квадрат, хранит координаты, длину стороны и точки, попавшие в него

Generation – поколение, хранит массив квадратов

GeneticAlg – класс генетического алгоритма. Генерирует начальную популяцию, производит эволюцию поколения, исправляет пересечения квадратов, появившиеся при скрещивании и мутации.

Скрещивание – равномерное.

В качестве функции отбора использовалось правило рулетки, а также указывается доля особей, участвующих в элитарном отборе.

Для исправления пересечений уменьшается размер одного из квадратов. Если один полностью вложен в другой, длина стороны внешнего квадрата уменьшается так, чтобы она была максимальной, но левая нижняя точка внутреннего квадрата не попала. А длина внутреннего становится максимальной не выходящей за рамки старого внешнего квадрата.

Немного изменена функция приспособленности для того, чтобы сильнее штрафовать решения с пустыми квадратами:

$$F = \frac{(\sum_{i=1}^m \text{точек в } i - \text{ом квадрате})}{2 \text{ кол-во пустых квадратов}}$$

#### Этап 4

Лутфулин – генетический алгоритм

Добавлен класс Solution – экземпляр решения, содержащий в себе массив квадратов. Имеет методы копирования и вычисления значения функции приспособленности.

В класс генетического алгоритма добавлена функция get\_populaton, возвращающая популяцию, среднее и максимальное значение приспособленности в ней. Также добавлен турнирный отбор и параметр, определяющий вид отбора для алгоритма. Перенесены методы в подходящие классы.

Гладилин – интерфейс

Генетический алгоритм интегрирован в интерфейс. Добавлена возможность указать число точек при случайной генерации. Кнопка Выполнить

до конца запускает поэтапное выполнение алгоритма и автоматически обновляет график функции приспособленности и результат работы.

Чтобы корректно выполнить алгоритм и его визуализацию, блокируются кнопки, которые могут помешать работе. Например, после нажатия «Выполнить алгоритм до конца» блокируются все кнопки кроме, и становится доступной «Остановить алгоритм».

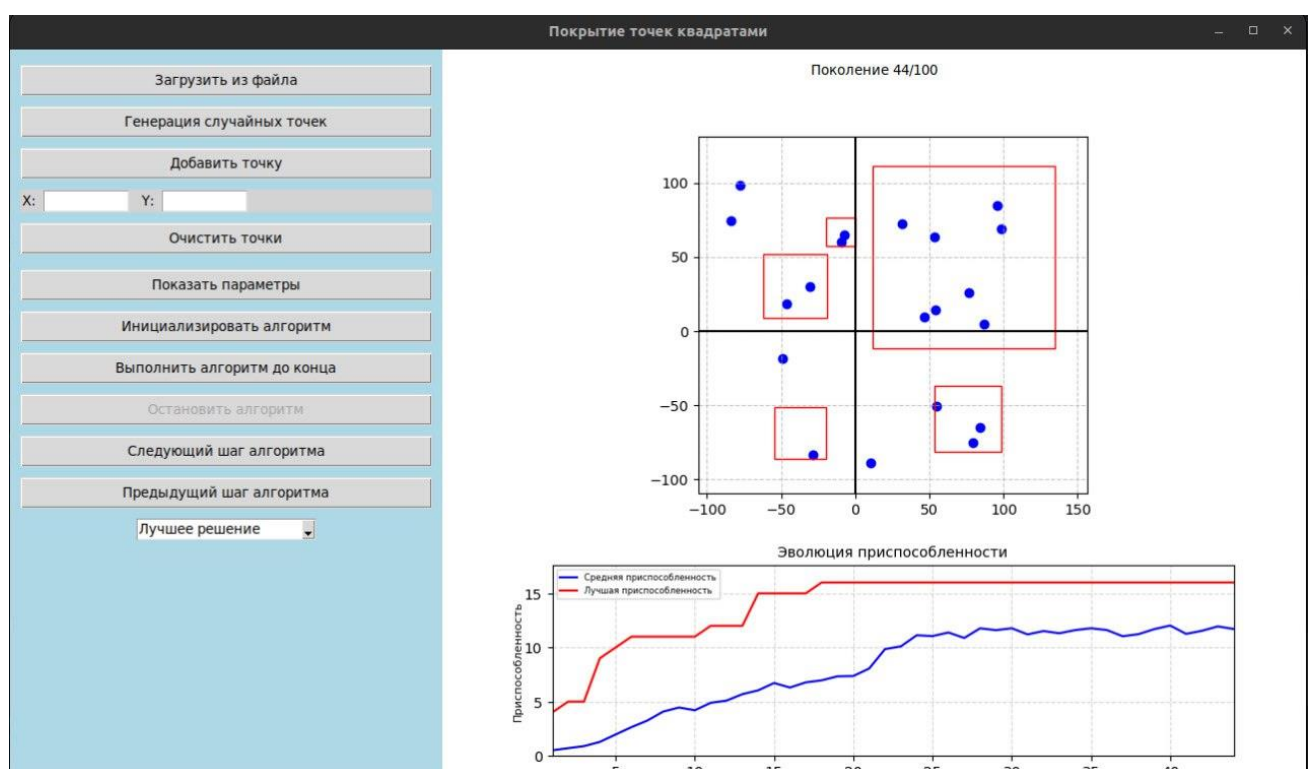
При добавлении или очистки точек, очищаются графики и у пользователь не может продолжить выполнение алгоритма для старых данных. Например, если был начат алгоритм для точек, то после добавления новой точки, необходимо инициализировать алгоритм перед выполнение алгоритма.

Отображение покрытия точек, теперь адаптирует размеры плоскости, чтобы все квадраты и точки были видны в интерфейсе.

Пользователь может выбирать решение в каждом поколении, отсортированные по убыванию функции приспособленности.

При совершении шага на уже созданное поколение, график приспособленности не меняется, а изображение покрытия меняется.

Запрещается ввод недопустимых значений.





## **Вывод**

Задача о покрытии точек квадратами с помощью генетического алгоритма и создан графический интерфейс для визуализации решения. Программа позволяет гибко настраивать гиперпараметры генетического алгоритма, благодаря чему можно сравнить его поведение при разных данных.