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

ОТЧЕТ

по учебной практике

Тема: Генетические алгоритмы

Кузнецова С.Е.
Максимова Е.Д.
Студентки гр. 3341 _____ Чинаева М.Р.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург
2025

Цель работы.

Составление плана работы над проектом, создание прототипа GUI, распределение ролей в бригаде.

Задание №18.

Задача о клике.

В заданном графе G = (V,E) необходимо найти клику максимального размера.

Распределение ролей в бригаде.

Максимова Екатерина: Реализация GUI

Чинаева Маргарита: Реализация основного алгоритма и подготовка для связи с GUI

Кузнецова Светлана: Загрузка и выгрузка данных и вспомогательные функции для основного алгоритма (проверка смежности, прохождение по соседям, проверка подграфа, является ли он кликой и т.п.) + генерация случайных данных

План реализации генетического алгоритма:

В качестве хромосомы используем массив размером n (количество вершин), в котором возможны только 2 значения – 0 и 1. Где 0 – вершина не включена в клику, 1 – вершина включена в клику.

Сортируем вершины в порядке невозрастания степени, переназначаем номера вершин согласно их порядку в итоговом массиве и храним массив изначальных порядковых номеров. Например, степень 3 вершины больше степени 1, а степень 1 больше 2. Тогда храним «массив перехода» 312. Это нужно для того, чтобы при разрыве хромосомы вершины, наиболее вероятно

находящиеся в одной клике, не разрывались. Наиболее важно это для вершин с большой степенью, так как они с большей вероятностью будут находиться в наибольшей клике.

Для генерации изначальной популяции используем «правило рулетки», в котором, чем больше степень у вершины, тем больше шанс случайно выбрать ее при составлении клики.

Алгоритм генерации одной хромосомы для начальной популяции:

1) Случайно выбираем вершину

2) Из ее соседей случайно выбираем следующую, которая при этом связана со всеми находящимися в клике.

3) Повторяем, пока есть допустимые вершины.

Функция приспособленности – сумма единиц в хромосоме – количество вершин в клике.

Оператор выбора родителей – метод рулетки, в котором размеры максимального и минимального секторов отличаются не более, чем на заданное пользователем число процентов. Это нужно для разнообразия популяции, чтобы одна особь не породила слишком много потомков.

Скрещивание родителей происходит многоточечно с постепенным уменьшением количества разрезов, так как в начале алгоритма нужно наибольшее разнообразие особей, а при приближении к решению это уже менее важно.

Отбор особей в новую популяцию будет проводиться отбором вытеснением, в котором из всех особей с одинаковой приспособленностью будет отдаваться предпочтение с разными генотипами.

Для каждой популяции производим мутации. С заданной пользователем вероятностью выбирается будет ли мутирована данная хромосома. Затем с заданной пользователем вероятностью мутируется каждый ген в хромосоме.

После мутации пока хромосома не станет кликой, постепенно ее уменьшаем. Из множества вершин с минимальной степенью случайно удаляем

одну, снова проверяем на клику, и так далее. Это нужно, так как в популяции должны быть только особи, которые являются кликами.

Генетический алгоритм останавливает работу, если было выполнено некоторое, заданное пользователем количество итераций «застоя», то есть лучший результат не менялся в течение нескольких итераций. А так как по свойству графов размер максимальной клики не превосходит наибольшей степени вершины в графе, то алгоритм также останавливает свою работу, если лучшая хромосома задает клику этого размера.

Загрузка/выгрузка, хранение данных

Граф хранится в виде списка смежности – в виде списка множеств длины n, где n – число вершин графа, и элемент с индексом v – это множество всех вершин графа, смежных с v. Таким образом, проверка смежности двух вершин происходит за O(1), проход по всем соседям вершины v – за O(deg(v)), где deg(v) – степень вершины v. Наличие множеств также гарантирует отсутствие дубликатов вершин.

Классы, реализуемые для хранения данных:

1. Класс Parameters – будет отвечать за хранение параметров генетического алгоритма (размер популяции, максимальное количество итераций, количество итераций в застое и т.д.), будет содержать функцию для загрузки параметров из json-файла.
2. Класс Graph – будет отвечать за хранение графа. Будет иметь функции загрузки графа из json-файла, генерации случайного графа, перевода графа из матрицы смежности в список смежности (список множеств).
3. Класс Chromosome – будет отвечать за хранение хромосомы (потенциальной клики) и ее приспособленности. Будет иметь функцию для вычисления приспособленности.
4. Класс Population – будет хранить текущую популяцию (список хромосом текущего поколения), лучшее решение и среднюю

приспособленность. Будет содержать функцию для обновления значений поколения: лучшего решения и средней приспособленности на каждом шаге.

5. Класс History – будет хранить промежуточные данные о ходе выполнения алгоритма – список номеров поколений, для которых записаны промежуточные данные, приспособленность лучшего решения на каждом поколении, среднюю приспособленность популяции на каждом поколении, лучшие решения на каждом поколении. Будет содержать функции для обновления данных, сохранения данных в json-файл.

6. Класс Manager – будет отвечать за загрузку/выгрузку всех данных из файла и хранить данные для работы алгоритма (экземпляры классов параметров, графа и истории). Будет содержать функции для загрузки всех данных из json-файла и сохранения их в json-файл.

Хранение входных и выходных данных:

1. Входные данные – параметры и граф будут храниться в json-файле с соответствующими ключами.

2. Выходные данные – историю эволюции – список лучших и средних приспособленностей для каждого поколения, список лучших решений – также будут храниться в json-файле.

Прототип графического интерфейса (GUI).

В графическом интерфейсе предусмотрено 3 основных окна:

Главное окно. Слева расположен виджет с визуаизацией графа, под которым расположен блок с проигрывателем, отображением размера текущей лучшей клики, кнопками сохранения и загрузки. Также на виджете распологается маленькая кнопка для вывода всей популяции на текущем шаге алгоритма. Справа — столбец текстовых полей с соответствующими параметрами, вводимыми пользователем и кнопка с вызовом окна для ввода матрицы.

Параметры для ввода пользователем:

- Population size – размер популяции.
- Max generations amount – максимальное количество поколений.
- Mutation rate – вероятноть мутации хромосомы.
- Gene mutation – вероятность мутации одного гена.
- Fitness scale – масштабирование (в процентах) приспособленности хромосом при селекции родителей.
- Max cut points – максимальное количество точек разреза при кроссинговере.
- Decr m.prob step (Decrease mutation probability step) – шаг (как количество поколений) для уменьшения вероятности мутации.
- Reduce cuts step – шаг (как количество поколений) для уменьшения количества точек при кроссинговере.

Окно с графиком изменения лучший и средней приспособленности в зависимости от поколения. Появляется при нажатии кнопки запуска алгоритма

Окно ввода матрицы смежности. Пользователь вводит размер матрицы в соответствующее поле. После нажатия кнопки заполнения матрицы создается таблица для ввода, где доступны ячейки выше диагонали (остальные значения заполняются симметрично). Также предусмотрена кнопка рандомной генерации. Визуализация графа и возможность запуска алгоритма на нем появляется после нажатия кнопки «Create graph».

При возникновении ошибок, связанных с вводом пользователя, появляется всплывающее окно с описание ошибки.

Для реализации gui будет использоваться tkinter - стандартная библиотека Python для создания графического пользовательского интерфейса.

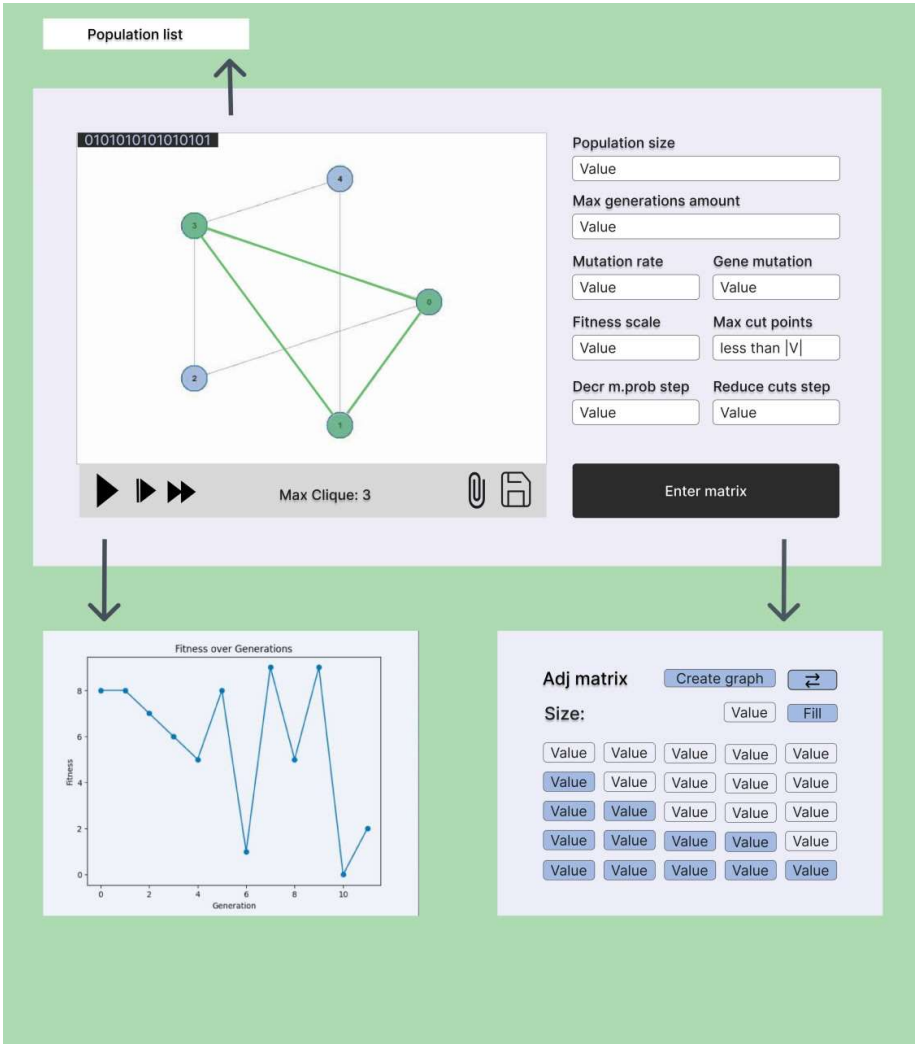

Рисунок 1 - Макет GUI