МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра МОЭВМ

ОТЧЕТ

по учебной практике

Тема: Генетические алгоритмы

Студенты гр. 2381 Мавликаев И.А.

Слабнова Д.А.

Дудкин М.В.

Преподаватель Жангиров Т.Р.

Санкт-Петербург

2024

Цель работы.

Разобраться в теме генетических алгоритмов, их видов и тонкостей реализации, а также создать проект, на них основанный.

Задание.

Для заданного полинома f(x) (степень не больше 8) необходимо найти параметры ступенчатой функции g(x) (высота ступеней), которая приближает полиномиальную функцию, то есть минимизировать расстояние |f(x) - g(x)| между функциями на заданном интервале [l, r]. Количество ступеней вводится пользователем.

Выполнение работы.

GUI будет реализовано на Tkinter.

Частично реализованы, частично в процессе реализации находятся классы:

class Chromosome - класс индивида.

Реализована дискретная рекомбинация.

Мутация реализована следующим образом: Берётся ген и ищется вокруг него интервал, не больше длины шага мутации (пока просто в define). И в пределах этого диапазона (и не выходя за верхнюю и нижнюю границу) - выбираем равномерным распределением мутацию - новое значение.
Chromosome - класс хромосомы (индивида) Поля:

- double probMutation - вероятность мутации каждого гена
- int number - номер хромосомы (возможно это поле избыточно и будет удалено)
- std::vector genes - вектор генов

- int length - длина хромосомы
- double max_mutation_step
- double down_border
- double up_border
- double estimate - оценка, наскоолько данная хромосома апроксимирует функцию. Методы:
- Chromosome(double probMutation, double down, double up, int number, int len) - инициализация. Создает вектор рандомных высот длины N между down и up и записывает в поле genes
- static std::vector recombination(Chromosome parent1, Chromosome parent2, int method = 0)
- void mutate(); - мутация
- double new_gene(double old_gene);
- static void discr_recomb(Chromosome parent1, Chromosome parent2, std::vector& answer); - дискретная рекомбинация
- void print_test();

class Population - класс популяции. Реализован элитарный отбор таким образом, что количество особей в следующей популяции такое же.
Поля:

- chromosomes - вектор индивидов.
- countIndivids - количество хромосом Методы:
- Population() - инициализация. Создание вектора рандомных хромосом.

Методы:

- updatePopulation - обновление хромосом (вызывается из elite_selection)
- elite_selection - Элитарная селекция.

Класс будет дорабатываться.

class Algorithm - класс, который инициализирует популяцию, запускает алгоритм. Его методы будут реализовывать канонический алгоритм ГА (возможно несколько)
Algorithm - класс, в котором выполняется алгоритм Поля:

- *Population population* - текущая популяция
- *Polynomial polynom* - полином
- *int iteration* - количество итераций (поколений)
- *double criterion* - критерий завершения Методы
- Инициализация (создается полином и первая популяция в зависимости от выданных критериев)
- *void start()*; - Запуск алгоритма Будут реализованы методы - различные ГА (первым канонический)

class Polynomial - класс многочлена. Хранит вектор коэффициентов, имеет методы вычисления значения многочлена в конкретной точке, а также интегральной разности между функцией и её апроксимацией (индивид).
Поля:

- *coeficients* - вектор коэффициентов многочлена
- *left* - левая граница интервала, на котором определён многочлен
- *right* - правая граница интервала
  Методы:
- Инициализация (инициализируется вектором коэффициентов coefs)
- *getValue* (получение значения многочлена)
- *evaluation* (оценка разности функции и её апроксимации)
- *print()*

class Starter - временный класс, который написан до того, как реализован GUI, отвечает за ввод данных и инициализацию класса Algorithm.

Метод *input*:

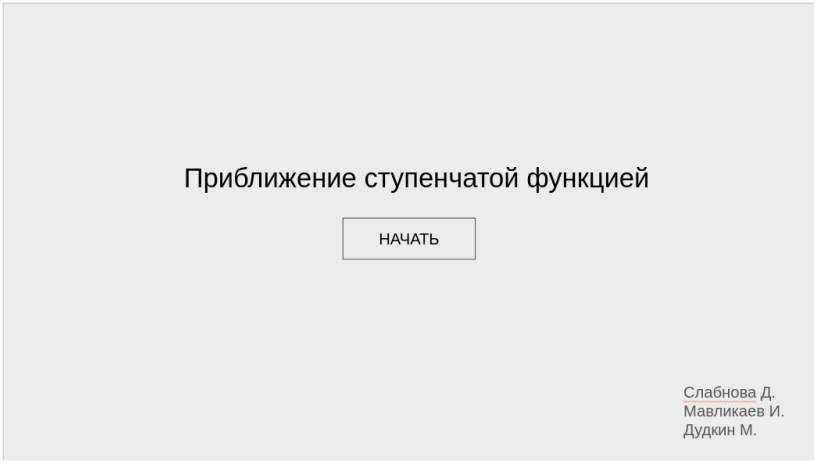
- вводится степень полинома
- вводится коэффициенты полинома
- вводятся left, right - границы интервала
- вводится count_ctep - количество ступеней (генов в хромосоме)
- вводится count_individs - количество хромосом (особей)
- вводится down, up - границы генов
- Инициализируется класс алгоритма Algorithm, с помощью его метода он ГА запускается

  Метод machineInput: -- будет реализован

  Метод fileInput -- будет реализован

GUI

1.На первой странице - информация о том, что за программа, кто её создал и кнопка начать. При нажатии на кнопку “начать” происходит переход на вторую страницу.

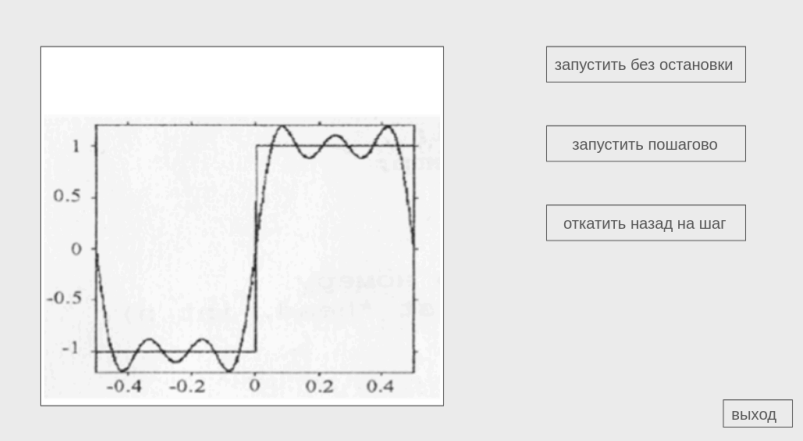


2.На второй странице, пользователю предлагается ввести значения параметров, таких как степень полином и его коэффициенты, количество

ступеней и интервал, на котором производится аппроксимация. После ввода, пользователю станет доступна кнопка продолжить, при нажатии на которую происходит переход на следующую страницу.

Также внизу страницы оставлено место под дополнительные параметры, такие как выбор метода рекомбинации хромосом, метода выбора родителей и метода выбора родительской пары.

3.На третей странице будет отображаться сам график, а также будет возможность запустить аппроксимацию пошагово или беспрерывно. Возможно, получится создать ползунок, передвигая который можно свободно перемещаться по шагам вперед и назад.



Связь GUI и основной программы

Так как GUI планируется писать на другом языке (на питоне), пока рассматриваются два способа связи двух программ:

1. запуск программы на C++ из программы (GUI на питон) через стандартные потоки ввода/вывода (библиотека os).

2. Использовать программу на C++ через ctypes как библиотеку в питоне.