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

ОТЧЕТ

по учебной практике

Тема: Нахождение экстремальных точек полиномиальной функции на заданном интервале

Студенты гр. 0304 _____ Гурьянов С.О.
Свечников И.В.

Руководитель _____ Жангиров Т.Р.

Санкт-Петербург
2022

ЗАДАНИЕ

НА УЧЕБНУЮ ПРАКТИКУ

Студенты Гурьянов С.О., Свечников И.В.

Группа 0304

Тема практики: Нахождение экстремальных точек полиномиальной функции (степень не выше 9) на заданном интервале

Задание на практику:

Для заданного полинома не выше 9 степени необходимо найти глобальный минимум.

Входные данные:

- Коэффициенты полинома 9 степени a0, a1, a2, …, a10
- Интервал поиска [l, r]

Сроки прохождения практики: 29.06.2022 – 12.07.2022

Дата сдачи отчета: 11.07.2022

Дата защиты отчета: 12.07.2022

Студенты _____ Гурьянов С.О.
Свечников И.В.

Руководитель _____ Жангиров Т.Р.

АННОТАЦИЯ

Данная работа подразумевает реализацию генетического алгоритма с использованием графического интерфейса. Она разделена на несколько итераций. Под каждую итерацию выделяется один пункт отчёта.

СОДЕРЖАНИЕ

	Введение	5
1	Итерация 2	6
1.1	Скетч с GUI, который планируется реализовать.	6
1.2.	Описание сценариев взаимодействия пользователя с программой	6
1.3	Определение и обоснование параметров модификации ГА для решения задачи.	7
2.1	Итерация 3	7
2.1.	Скриншоты начальной реализации GUI.	7
2.2.	Описание ввода данных (через терминал).	7
2.3.	Описание реализации алгоритма.	8
3	Итерация 4	9
3.1	Изменение алгоритма	9
3.2.	Изменение графического интерфейса	9
4	Итерация 5	10
4.1	Добавление ввода через файл	10
4.2	Исправление багов алгоритма	10
	Заключение	11
	Список использованных источников	12

ВВЕДЕНИЕ

Целью работы является программная реализация решения поставленной оптимизационной задачи на языке Python с использованием ГА. Основными задачами выполнения работы являются: формирование прототипа GUI и выбор метода решения задачи, частичная реализация программы, в которой присутствует GUI и реализовано хранения данных и основные элементы ГА. Также создана инструкция по сборке и запуску программы. На конечной итерации должна быть выполнена цель работы, а именно программа должна полностью работать вместе с её графической частью, ГА должен гарантированно находить решения.

1. Итерация 2

1.1. Скетч с GUI, который планируется реализовать.

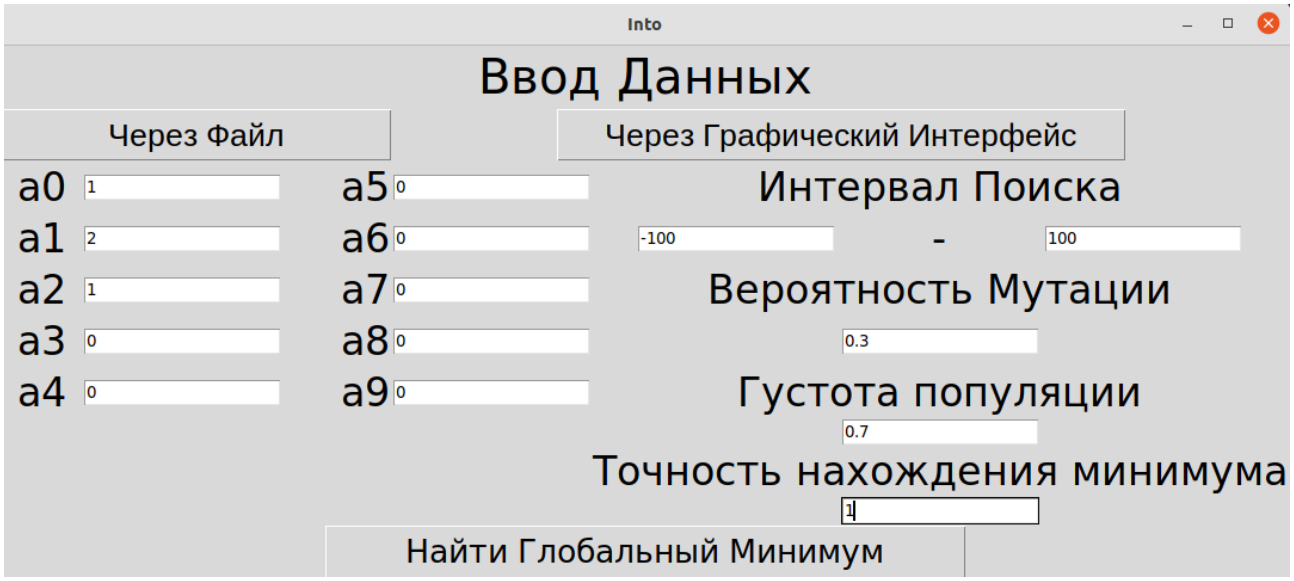

Рис. 1 — начало взаимодействия с программой

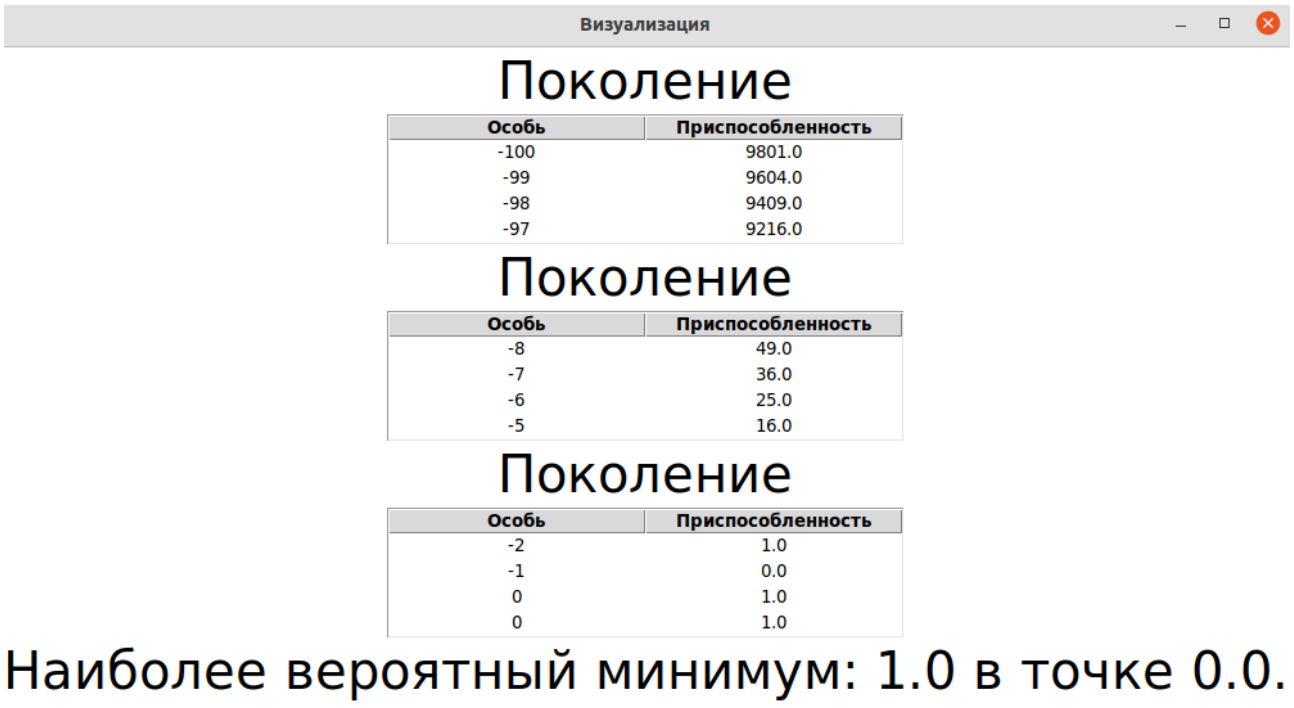

Рис. 2 - шаг алгоритма

1.2. Описание сценариев взаимодействия пользователя с программой

Сценарий взаимодействия пользователем с программой:

1) Запуск программы.

2) Выбор способа ввода данных.
3) Проверка корректности данных.
4) Вывод результата в виде таблиц.

1.3 Определение и обоснование параметров модификации ГА для решения задачи.

Для решения данной задачи будет использоваться метод прерывистого равновесия. Обосновано это тем, что данный метод позволяет эффективно выходить за пределы локальных ям и формировать глобальный минимум. Оператором выбора родителей будет являться панмиксия. Получившиеся в результате кроссинговера потомки и наиболее пригодные родители случайным образом смешиваются. Из общей массы в новое поколение попадут лишь те особи, пригодность которых выше средней. Тем самым достигается управление размером популяции в зависимости от наличия лучших особей. В качестве оператора кроссинговера будет выбран одноточечный кроссингновер, а в качестве оператора мутации — одноточечная мутация, поскольку метод прерывистого равновесия предполагает их использование. Таким образом, необходимо учитывать вероятность мутации, заданную пользователем. Оператор отбора в новую популяцию - элитарный отбор, поскольку необходимо случайным образом смешать родителей и получившихся потомков.

2. Итерация 3

2.1. Реализация GUI.

GUI реализован при помощи библиотеки Tkinter языки Python. Первое окно (стартовое) нужно для выбора ввода данных и возможности ввода параметров через графический интерфейс (файл *interface.py)*. Второе окно (*interface2.py)* используется для демонстрации итерации алгоритма. Существует возможность перейти к следующей итерации или в конец алгоритма (когда формируется конечное решение).

2.2. Описание ввода данных (через терминал).

На этом этапе поддерживается работа с программой только через терминал. При запуске программы вначале требуется через пробел ввести коэффициенты полинома, начиная от x^0, заканчивая x^9 (10 чисел). Далее вводятся 2 числа — это интервал нахождения минимума полинома. Далее вводится параметр гу-

стота — данный параметр показывает, сколько целых чисел промежутка будет в начальной популяции (густота 1 означает, что в начальной популяции будут все целые числа в данном интервале, густота 0.1 означает, что в начальной популяции будет 10% всех целых чисел данного промежутка). Следующий параметр — вероятность мутации хромосомы при одной итерации.

2.3 Описание реализации алгоритма.

Для реализации алгоритма используются функции *find_suitability(value, coeffs)*, *make_descendants(generation)*, *mutate(sequence, credibility)*. Функция *find_suitability(value, coeffs)* позволяет определить приспособленность решения. Она принимает на вход точку *value* и список коэффициентов в том порядке, в каком они подавались на вход через терминал. Данная функция возвращает значение полинома в данной точке. Функция *make_descendants(generation)* принимает на вход поколение (последовательность точек) и выполняет формирование потомков. Для этого каждому элементу списка *generation* ставится в соответствие случайное число — индекс этого же списка. В каждой такой паре формируется 2 потомка (то есть общее число потомков в 2 раза больше общего числа родителей). Формирование происходит по следующему принципу: выбирается случайная точка в хромосомах родителей, первый потомок принимает все гены до этой точки у первого родителя и после этой точки — у второго родителя. Второй потомок, наоборот, принимает все гены первого родителя после данной точки и все гены второго родителя до этой точки. Такое действие проделывается для каждой пары родителей и формируется список потомков. Функция *mutate(sequence, credibility)* принимает последовательность хромосом *sequence* и вероятность возникновения мутации *credibility*. Выполняется обход последовательности, для каждого её элемента генерируется случайное число от 0 до 1, если это число меньше *credibility*, то происходит мутация. Случайным образом инвертируется какой-либо ген хромосомы. Для этого выполняется операция «исключающее или». Реализация алгоритма находится в файле *algorith.py*.

3. Итерация 4.

3.1. Изменение алгоритма.

В алгоритм был добавлен параметр «точность», который позволяет определять порядок нахождения минимума. Данный параметр принимает значение от 0 до 1. При значении 1 минимум будет определяться в целых точках, при значении 0.1 — точность минимума будет формироваться до десятых долей и т. д. Это достигается с помощью изменения коэффициентов полинома и расширения интервала. Например, при точности 0.1 коэффициент при x будет уменьшаться в 10 раз, коэффициент при x^2 будет уменьшаться в 100 раз и т. д., а интервал будет расширяться в 10 раз. Если минимум для изменённого полинома будет в точке x0, то минимум для исходного полинома будет определён с точностью до десятых долей в точке x0/10. Однако при очень высокой точности алгоритм может выдавать некорректный результат. Связано это с тем, что чем больше степень x, тем в большее количество раз надо уменьшить коэффициент при нём. Таким образом, коэффициент при x^9 уменьшится в число раз 10^(9k), где k — число цифр после запятой. В результате вычисления могут получиться неточными. Так, для точности результата этот параметр лучше всего менять тогда, когда коэффициенты при высоких степенях равны нулю. В ином случае лучше поставить точность до целых (1).

3.2. Изменение графического интерфейса.

Теперь для взаимодействия с программой нужно использовать графический интерфейс. При запуске программы открывается окно с вводом параметров. На текущий момент ввод данных возможен только при помощи графического интерфейса (пока что не реализован ввод через файл). После ввода данных следует нажать кнопку «Найти Глобальный минимум». При введении некорректных данных выводится сообщение об этом. Если данные введены корректно, то алгоритм отрабатывает и выводит промежуточные результаты, а также итоговый результат на отдельное окно. Промежуточные результаты выводятся в виде таблиц. При этом окно для ввода не закрывается, что даёт возможность выполнить алгоритм для других данных.

4. Итерация 5.

4.1. Добавление ввода через файл

Изменён модуль *interface.py*. Теперь в нём функционируют кнопки выбора ввода. При выборе ввода через файл предоставляется возможность выбрать файл. Данный файл должен иметь текстовый формат. Он должен содержать данные, введённые через пробельные символы, в следующем порядке: сначала идут 10 коэффициентов, затем — левый и правый интервалы поиска, потом — вероятность мутации, густота популяции и точность нахождения минимума.

4.2. В текущей итерации исправлен баг с некорректным формированием интервала нахождения минимума. На прошлых итерациях могла возникнуть ошибка, связанная с некорректными границами интервала (к примеру, если левое значение нахождения минимума было равно -10, то интервал определялся от точки -9). Исправлен баг с пустым поколением. Это может возникнуть в результате того, что после мутации все потомки могли выйти за пределы интервала (особенно, когда минимум был близко к границе интервала). Теперь при возникновении такой ситуации выполняется повторные формирование потомков и мутации.

ЗАКЛЮЧЕНИЕ

Созданы скетчи графического интерфейса и выбрана модификация генетического алгоритма. Её выбор был обоснован. Разработана программа, реализующая поставленную задачу на целых числах. Разработка осуществлена при помощи генетического алгоритма. Данный алгоритм даёт не всегда точное решение, однако его точность оказывается высокой (в большинстве случаев даёт правильный результат). Конечный результат, при этом, зависит от входных параметров. Реализовано взаимодействие с программой при помощи графического интерфейса. Вывод осуществляется в отдельном окне графического интерфейса. Есть возможность просмотреть каждую итерацию. Добавлен ввод через файл.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Панченко Т.В. Учебно-методическое пособие “Генетический алгоритмы”.