

**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**  
**(степень не выше 9) 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  $a_0, a_1, a_2, \dots, a_{10}$
- Интервал поиска  $[l, r]$

Сроки прохождения практики: 29.06.2022 – 12.07.2022

Дата сдачи отчета: 05.07.2022

Дата защиты отчета: 05.07.2022

Студенты

\_\_\_\_\_

Гурьянов С.О.  
Свечников И.В.

Руководитель

\_\_\_\_\_

Жангиров Т.Р.

## **АННОТАЦИЯ**

Целью работы является знакомство и применение на практике генетических алгоритмов, а также их оптимизаций, для решения поставленной задачи о назначениях. Генетические алгоритмы — это адаптивные методы поиска, которые в последнее время используются для решения задач оптимизации. В них используются как аналог механизма генетического наследования, так и аналог естественного отбора. При этом сохраняется биологическая терминология в упрощенном виде и основные понятия линейной алгебры.

## СОДЕРЖАНИЕ

	Введение	5
1	Итерация 3	6
1.1.	Скриншоты начальной реализации GUI.	6
1.2.	Описание ввода данных (через терминал).	7
1.3.	Описание реализации алгоритма.	
	Заключение	9
	Список использованных источников	10

## **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 3

### 1.1. Скриншоты начальной реализации GUI.

Раздел будет позже.

### 1.2. Описание ввода данных (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 густота — 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.

### 1.3 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 продлевается 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».

## **ЗАКЛЮЧЕНИЕ**

Была 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.



## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

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