САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Отчет по лабораторной работе №6
по курсу «Современные инструменты анализа данных»
Тема: **Решение задачи коммивояжера с помощью генетического алгоритма**

Выполнили:
-

Проверила:
-

Санкт-Петербург
2023 г.

Задание 1. Подготовка данных

Нами было выбрано 5 городов с абсолютно разных точек земного шара, для которых мы составили таблицу соответствия координат по широте и долготе

	X	Y
Magnitogorsk	53.230000	59.020000
BuenosAires	-34.355900	-58.225500
Shanghai	31.230400	121.473700
Perth	-31.953512	115.857048
Honolulu	21.315603	-157.858093

Координаты городов

Задание 2. Подсчет расстояния

Сначала посчитаем расстояние между точками на сфере по их широте и долготе, используя формулу Haversine. Формула имеет вид:

$$a = (\sin\frac{\Delta\phi}{2})^2 + \cos\phi_1 * \cos\phi_2 * (\sin\frac{\Delta\alpha}{2})^2$$

где Δϕ,Δα,ϕ1,ϕ2 – разница в широтах между двумя точками, разница в долготах между двумя точками, широты первой и второй точек соответственно. В таком случае итоговое расстояние будет равно:

$$d = round(\arcsin\sqrt{a} * R)$$

Теперь попарно подсчитаем расстояния между городами и выведем полученные данные в симметричную относительно главной диагонали матрицу

	Magnitogorsk	BuenosAires	Shanghai	Perth	Honolulu
Magnitogorsk	0	14757	5482	10942	10998
BuenosAires	14757	0	19666	12616	12176
Shanghai	5482	19666	0	7051	7948
Perth	10942	12616	7051	0	10910
Honolulu	10998	12176	7948	10910	0

Расстояния между городами

Задание 3. Подсчет оптимального пути с помощью генетического алгоритма

Для начала опишем основные этапы алгоритмы, которые мы реализовали в коде:

Создание новой популяции: в зависимости от размера матрицы расстояний (в данной задаче - набор из пяти городов, без повторений, так как мы смотрим на путь, проходящий через каждый город по одному разу) - набор уникальных хромосом; в нашем случае хромосомой является набор городов (их кодов), указывающий на их порядок посещения (порядок поиска расстояния между ними)

Размножение: кроссовер между двумя выбранными родительскими хромосомами x и y заключается в выборе некоторой подпоследовательности хромосомы x и вставки по соответствующим позициям в хромосому y. Образованная таким образом комбинация родительских хромосом x и y - результат размножения

Мутация: случайная перестановка двух символов (генов в хромосоме, у нас - кодов городов в последовательности) с установленной вероятностью (у нас - 10%) при проведении размножения (при получении новой рекомбинационной хромосомы).

Отбор: отбор по признаку приспособленности (fitness), который надо максимизировать: в нашем случае мы ищем такую последовательность городов, что общая длина пути минимальна, что эквивалентно поиску хромосомы (индивида) с наибольшим значением приспособленности. Тогда приспособленность рассчитываем, как коэффициент, обратный расстоянию пути. Для заданного количества поколений (=количество итераций размножения) в популяции проводим размножение индивидов (выбор которых производится, опираясь на вероятность выбора каждого индивида=признак приспособленности индивида), такое количество раз, сколько индивидов в начальной популяции.

```
[[0, 1, 4, 3, 2], [4, 3, 1, 0, 2], [4, 3, 1, 2, 0], [3, 0, 1,
2, 4], [3, 1, 2, 0, 4], [0, 1, 2, 4, 3], [0, 4, 2, 1, 3], [4,
0, 3, 1, 2], [2, 4, 0, 1, 3], [0, 3, 2, 4, 1], [0, 1, 2, 4, 3],
[2, 4, 0, 3, 1], [4, 1, 3, 0, 2], [4, 2, 1, 3, 0], [2, 0, 4, 1,
3], [1, 0, 4, 2, 3], [2, 1, 0, 3, 4], [0, 3, 1, 2, 4], [0, 1,
2, 3, 4], [3, 4, 1, 0, 2]]
```

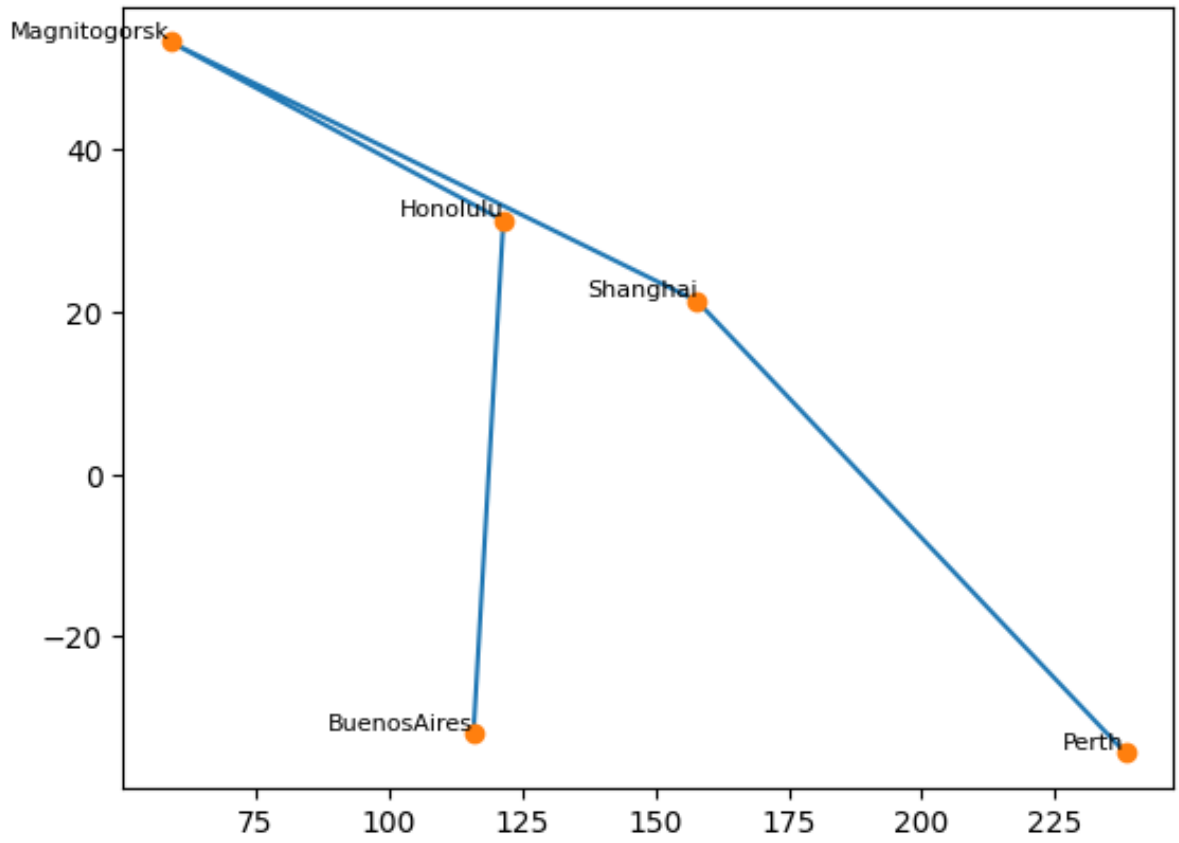
Пример сгенерированной популяции

```
Best path: [1, 4, 0, 2, 3]
Total distance: 35707
Fitness: 2.800571316548576e-05
```

Результаты алгоритма при **population_size=20** и **generations=50**

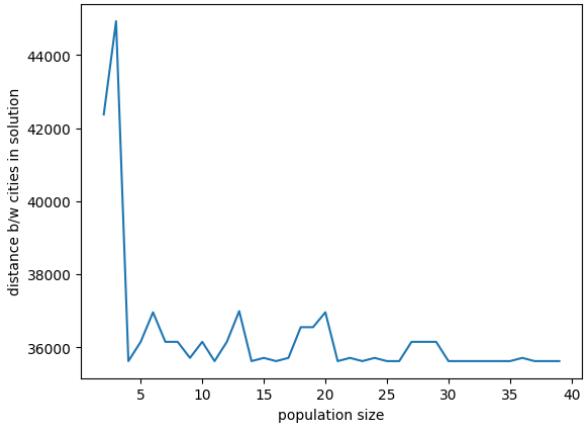
Итого, самый оптимальный путь составляет **35 707км** в порядке: **Perth - Shanghai - Magnitogorsk - Honolulu – BuenosAires**

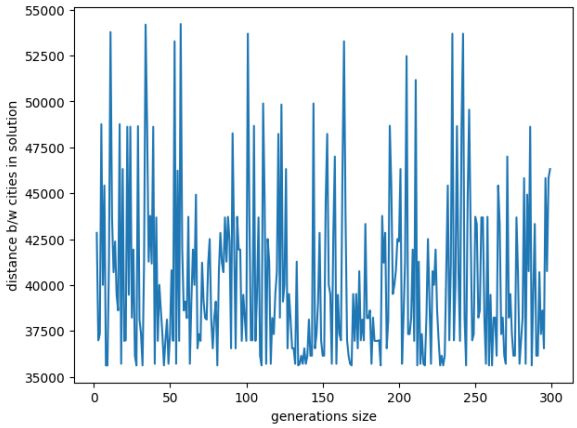
Немного модифицировав для удобства и корректности отображения координаты, мы можем изобразить весь полученный путь



Результаты алгоритма при **population_size=20** и **generations=50**

Можно попробовать рассмотреть изменение решения посредством перебора кол-ва популяции и кол-ва поколений соответственно





Рассмотрев различные значения, можно сделать вывод, что при **population_size > 12** (или гарантированно при **populations_size > 30**) расстояние достигает минимума, во всех этих случаях **max_generations = 50** по умолчанию. Если же наоборот фиксировать **population_size = 4** и задавать диапазон поколений до **300**, то алгоритм ведет себя неустойчиво и не сходится