**REPORT**

**Chapter 3: GENETIC ALGORITHM FOR THE TRAVEL SALESMAN PROBLEM**

**3.1 Introduction**

Genetic algorithms (GAs) are stochastic approach which does not rely on derivative based modeling (no-model approach). It is a simple yet way more effective algorithm designed based on the idea of biological evolution of the species. Core idea behind this algorithm is that the most suitable individuals are likely to survive and mate; which will result into better generation & that will be more healthy and fitter than their parents. GAs work with population of chromosomes that are represented by some underlying parameters set codes.

As species passes their genes into their next generation to make species capable to survive their life even better than them, same idea that have been used in genetic algorithm which brings the biological terms into it i.e., chromosome in the context of algorithm: a sample selection procedure from the available population, crossover is the way to generate child of the parents which means from two parents one next generation will be created, mutation is the way to mutate the current samples with other chromosome which helps to generate more accurate results from the sample & finally evaluation and selection; is to find the best mates out of selected samples.

**3.2 Genetic Algorithm**

As we have already introduced the terms like chromosome, crossover and mutation, in this topic we will explain in-depth approach of this algorithm to solve TSP (Traveling Salesman Problem).

**Algorithm flow in the context of TSP:**

Step-1 Randomly generate initial population.

Step-2 Based on the cost function, evaluate all individual chromosome.

Step-3 If generation limit exceeded, then go to step 5

Step-4 a) Perform Selective reproduction,

b) Do Crossover,

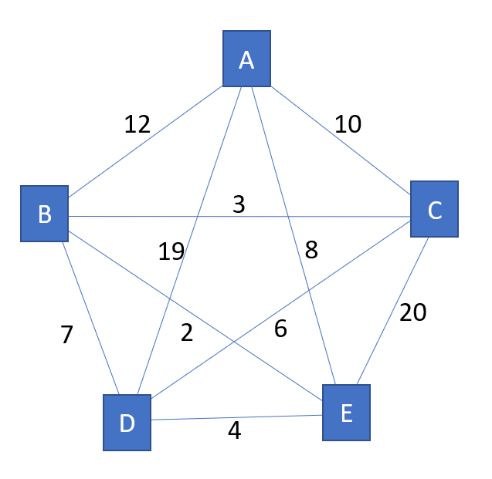
c) Do Mutation, then, Go to Step 3

Step-5 Select the best individual from the result and display.

**Step wise explanation:**

**Step-1:** Randomly generate initial population

Generate M number of paths from N! possibilities where N is the number of cities and M is the number of samples that we will select i.e., M = Initial population size.



For example, considering the one of test case of the problem statement, where number of cities N = 5. Thus, the possible valid paths in the TSP would be 5! = 120. Now, if we select M = 4 which means out of 120 we have selected the 4 number of samples, hence M = initial population size = 4. E.g., initial population P = [[5, 2, 4, 3, 1], [4, 1, 5, 3, 2], [3, 5, 2, 4, 1], [5, 1, 4, 2, 3]]

**Step-2:** Based on the cost function, evaluate all individual chromosome.

In this step, Individual path (chromosome) will be evaluated based on the cost or fitness function where the cost function = 1 / (total distance of individual path).In the previous example corresponding fitness value V for the path vector P would be, V = [0.029411, 0.017857, 0.016949, 0.012987]. Individual fitness values vector V calculated based on the formula,



Where dist(Pi[j],Pi[j+1]) is the distance between each pair of node. It indicates distance between jth and j+1th index of the node of ith path vector (chromosome) of initial population P. E.g., This fitness values vector V indicates less the distance, more fitness score it has.

**Step-3:** If generation limit exceeded, then go to step 5

Generation limit is the parameter which depicts how many times loop will iterate and generate new evolved batches of various paths.

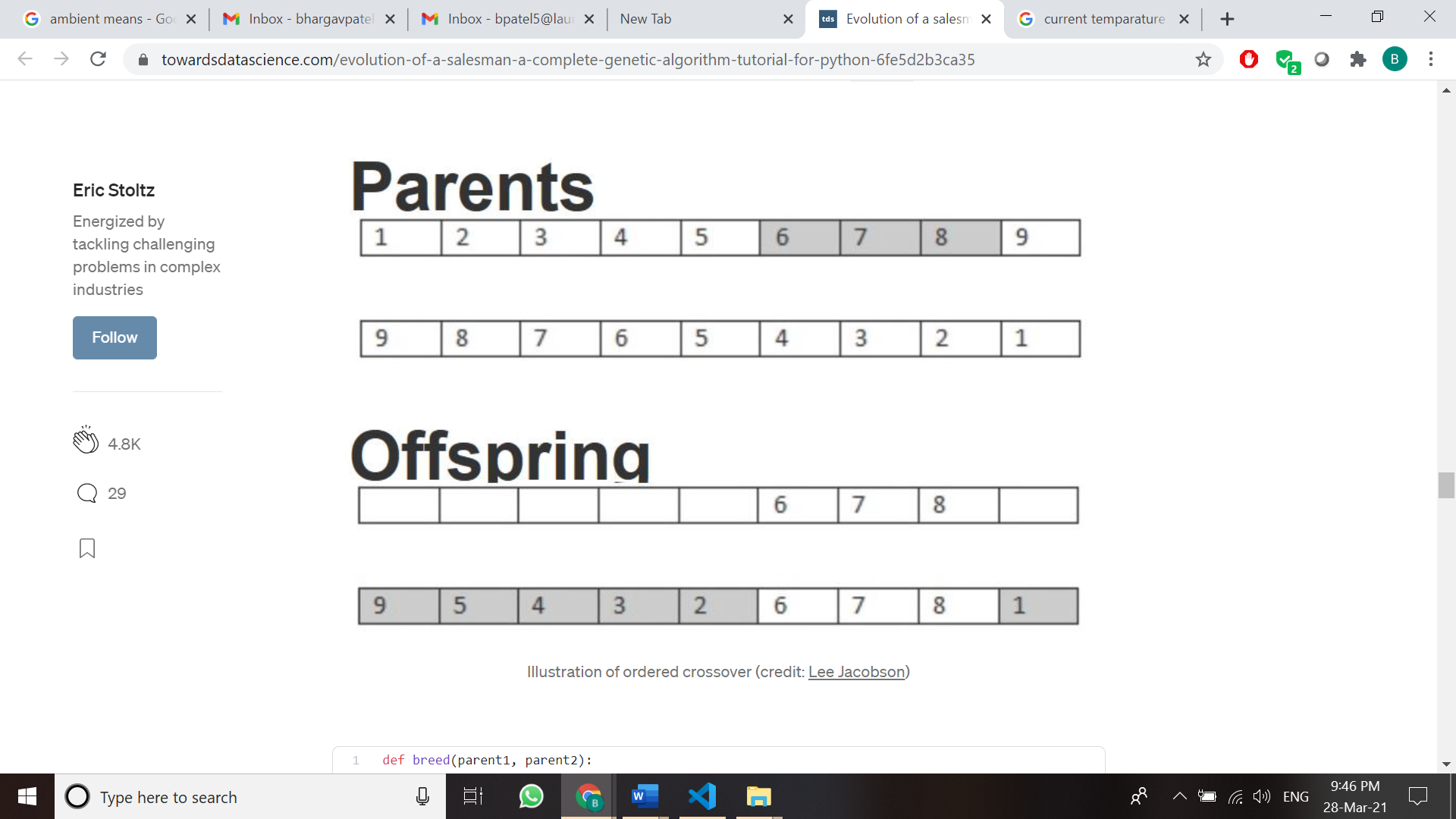
**Step-4:**

a) Perform Selective reproduction,

Selective reproduction is to Select the number of paths from the population based on the elite size given. Elite Size E is a parameter which indicates how many best paths (chromosomes) should be selected from the population. For example, Here, population size is M = 4 and the Elite size E = 3. So, it indicates out of 4 generated population we will select only 3 best paths (chromosomes).

b) Do Crossover,

After selecting the elite chromosome in selective reproduction, now randomly two of them will merge together by copying some random path portion of one chromosome to another and generates new one. E.g.,



c) Do mutation,

Mutation serves an important function in GA, as it helps to avoid local convergence by introducing novel routes that will allow to explore other parts of the solution space. Which means in the path with specified low fitness valued two cities will swap places in the rout which mimics the behavior of mutation in genes. E.g., [5, 2, 4, 3, 1] path will have [5, 2, 3, 4, 1] 3rd and 4th index mutation. Eventually this mutation will help to swap explore other parts when crossover will be performed. If the mutation rate like 0.25 is given then 1 path from vector P will have mutation effect. i.e., length(P) \* 0.25 = 4 \* 0.25 = 1 path of vector P.

**Step-5:** Select the best individual from the result and print it.

If whole algorithm approaches to maximum generation limit then at the end, from all elite population P with highest fitness valued path will be selected and displayed along with the distance cost.

**3.3 Numerical Experiments**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Number of cities** | **Generations** | **Population Size** | **Elite Size** | **Mutation Rate** | **Initial Path distance** | **Last generation’s**  **Elite Path distance** |
| 5 | 100 | 4 | 3 | 0.25 | **35** | **31** |
| 5 | 250 | 4 | 3 | 0.25 | **35** | **32** |
| 5 | 500 | 4 | 3 | 0.25 | **34** | **31** |
| 6 | 100 | 25 | 20 | 0.25 | **87** | **76** |
| 6 | 250 | 25 | 20 | 0.25 | **81** | **76** |
| 6 | 500 | 25 | 20 | 0.25 | **76** | **76** |
| 6 | 100 | 25 | 20 | 0.25 | **1272** | **1248** |
| 6 | 250 | 25 | 20 | 0.25 | **1427** | **1248** |
| 6 | 500 | 25 | 20 | 0.25 | **1272** | **1248** |
| 15 | 100 | 100 | 20 | 0.25 | **2049** | **1225** |
| 15 | 250 | 100 | 20 | 0.25 | **2241** | **1194** |
| 15 | 500 | 100 | 20 | 0.25 | **2276** | **1194** |
| 29 | 100 | 200 | 20 | 0.25 | **80966** | **35548** |
| 29 | 250 | 200 | 20 | 0.25 | **86412** | **33734** |
| 29 | 500 | 288 | 20 | 0.25 | **81833** | **31315** |

We have conducted the numerical experiments on test cases given, there are three generation limits (hyper parameter) on which algorithm was tested, i.e., 100, 250, 500. As generation increases elite generation increases and provides the best result. Thus, we can say that more the generation, more the accurate elite result will be.

Similarly, variation is possible into the mutation rate, elite size and population size as well but that can generate multiple rows of the experiments approx. more than 405 rows into the experiment table. Thus, instead of considering them, team have given the priority to Generations because that can show the perfect comparison of elite path distance in the large number of cities case scenario.

**3.4 Conclusion**

Evolutionary algorithms have been around since the early sixties. They apply the rules of nature: evolution through selection of the fittest individuals, the individuals representing the solutions to a mathematical problem. Genetic algorithms are so far generally the best and most robust kind of advanced algorithms which gradually approaches to the global optima of the entire problem. Such as in TSP case, 5,6 cities scenario the greatest result was approached within 100 generations. Similarly, global optima can be achieved for the large number of cities as well by manipulating the hyper parameters. So, out of N! it efficiently finds solution in just O(GXPXM) complexity where G is generations, P indicates Population and M indicates elite selection of individuals [[1](https://stackoverflow.com/questions/9146086/time-complexity-of-genetic-algorithm#:~:text=Given%20the%20usual%20choices%20(point,order%20of%20O(gnm)).)].