**Amazon Delivery Truck**

The problem we are trying to solve is like **Travelling Salesman** problem such that the traveling salesman travels around different points or location in the city to sell his products. He wants to cover all the different locations and points by travelling through the smallest distance.

Our problem is based on **Amazon Delivery truck** which is helping customer get their products, who are situated in different cities across the United States. The truck wants to reach and deliver products to all of its customers in the different cities, while covering the smallest distance possible.

Our program tries to compute different routes which the delivery truck can take, using *Genetic Algorithms*. Our Genetic Algorithm tries to figure out the smallest route possible by trying different combination of routes possible. We provide an initial route based on which it then tries to converge towards the smallest route possible. As the final output, it displays the best route which the solution has computed and draws it as the output.

There are different variables in our **GeneticAlgorithm** which are as follows:

1) **POPULATION\_SIZE** - It is the population size for the genetic breeding. Since we have 10 Cities in to our consideration for the delivery truck to cover, hence the population size of '10' is optimum for our solution.

2) **MUTATION** - It is the mutation rate for our algorithm. We tried with different values of Mutation, and figured out that the mutation rate of 0.5% was optimum for the solution to appear for 750 to 800 generations.

3) **TOURNAMENT\_SELECTION** - It is the tournament selection size for our algorithm. It picks out 3 best routes from our pool of best solutions.

4) **ELITE\_ROUTES** - It is the number of best routes which should be considered as the prime best route from the pool. In our case we have taken it to be 1.

5) **GENERATIONS** - It is the number of Generations for which the Genetic Algorithm should carry out its evolution. We found the optimal value for this field to be 750 which is well suited for the mutation rate of 0.5%.

*Both the value of Generation as 750 and the mutation rate of 0.5% was the most optimal solution which converges towards the best route in the least number of generations. And it was also the most suitable value which gave out the most optimum route most frequently.*

The **Fitness Function** in our program calculates the fitness value on the basis of the **Total Distance Calculated**.

There are also different variables in the City class. They are explained below:

1) **EARTH\_RADIUS** - Value of Earth Radius for our problem.

2) **D\_TO\_R** - Constant value for converting Degree to Radian.

3) **KM\_TO\_MILES** - Constant value for converting Kilometer to Miles.

4) **CalculateDistance** - This function calculates the distance between the two cities based on their Latitudes and Longitudes. The mathematical formula used in the method was taken from an already existing source.

Some output are as follows:-

