

## EXPLANATION OF THE GENETIC ALGORITHM PROGRAM

In the current Genetic Algorithm program,

The target message is set to "Colorado" for testing purposes. A population density generated by a random module of the python has been used. As of now, the population density/population size is equivalent to

= 10 \* length (target message). Population size can be changed by replacing 10 by the desired number.

The consideration of converting punctuations into the specified binary has been addressed by inculcating the source code's ASCII elements. Somehow, the fitness function has also been modified to increase the scope of the program. With the introduction of ASCII elements in place of the specified fitness function, it has also drastically reduced computation time for the given target message.

As mentioned for the extra credit consideration, the time taken to complete iteration over the specified population size has been duly specified in the "**Assignment\_ME8710\_Output.pdf**". Also, notifying the total number of iterations, it worked through to generate the given solution.

The code has some computational time limitations. If the user gives the input string, then the total time of computation drastically increases. To reduce the time of computation, the randomly generated string is used.

The dependencies:

Custom Module:

genetic.py (Need to be sourced into the site-packages folder for the code to run effectively. Though this package can be directly used by pushing the same into the catkin\_workspace)

Common Modules:

- random
- datetime
- string
- itertools (Though comes as a part of the scikit and NumPy packages)

The code inspiration has been drawn from Corey Schafer's OOP tutorials for python.