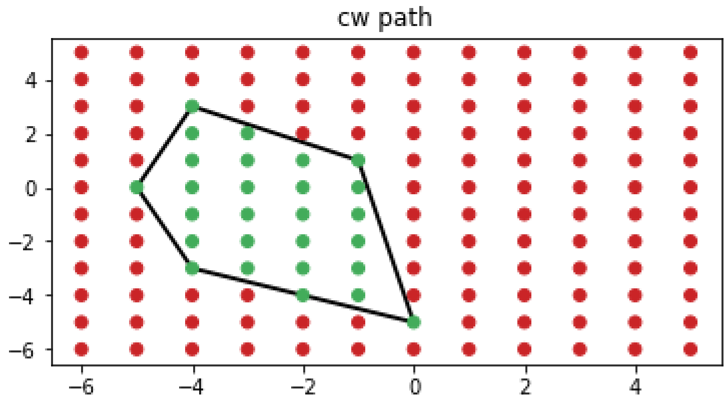
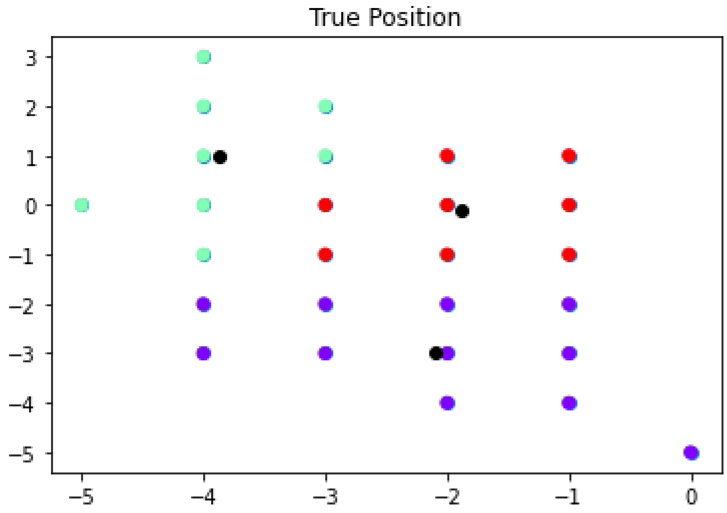
Automatic Drone Route Planning Prototype

Our drone route planning system uses the PyShp library to read the shapefile and outputs polygon by using Matplotlib. We plot the points all across the graph which are distanced by √2 times the radius of the drone coverage.



By using K-means Clustering we make n-clusters from points (n=number of drones). Now drones take in their course such that each cluster corresponds to a particular drone.

For 3 Clusters:



Travelling Salesman Problem will map the shortest possible route a drone can take to cover all the points in its corresponding cluster. The drone would have an initial capacity which would be decreased due to the weight of the path taken by the drone. A condition that the capacity of the drone must never reach negative is applied.

If the drone finds its capacity to be reaching a negative value by taking a path it immediately searches for the nearest recharge station and gets recharged to continue the current course. We run these multiple times to ensure success.

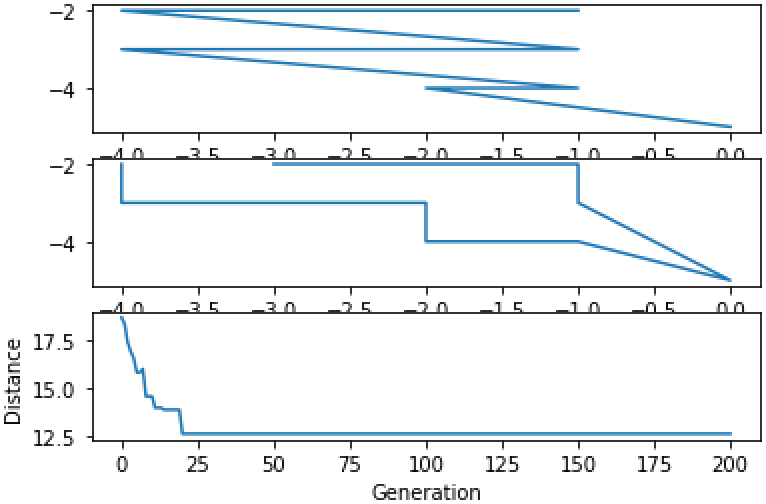
This Travelling Salesman problem can be solved optimally using the Genetic Algorithm.

NOTE:

- The first subplot of a graph represents the initial path chosen by the algorithm(random)
- The second subplot represents the optimal path algorithm found until nth generation(for below example we used 200 generations)
- The third subplot represents the decrement in a distance of path for every generation.

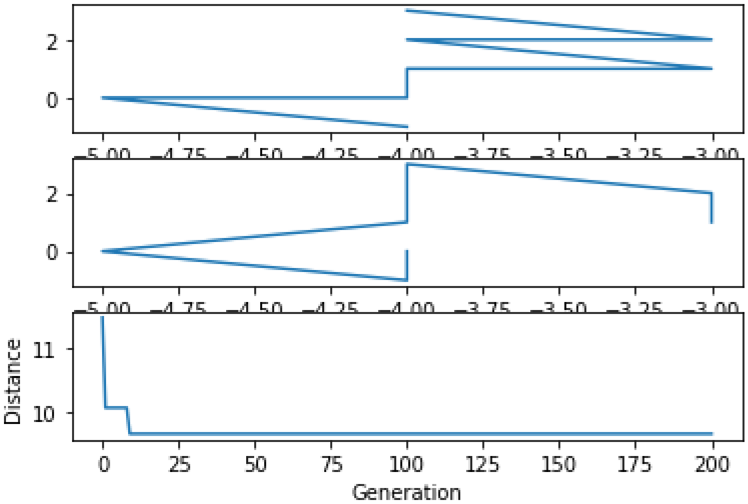
For Cluster 1:

Initial distance: 18.640986324787455
Final distance: 12.650281539872887



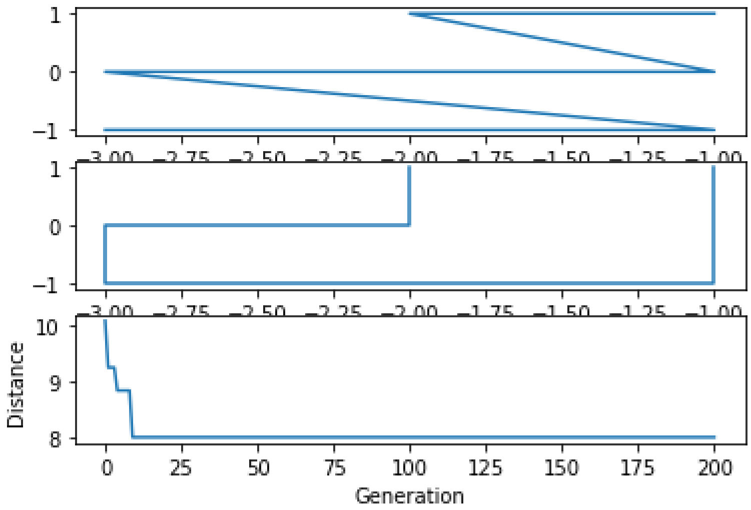
For Cluster 2:

Initial distance: 11.478708664619075
Final distance: 9.65685424949238



For Cluster 3:

Initial distance: 10.06449510224598
Final distance: 8.0



How to run

Run prototype.ipynb notebook.

It plots the points in polygon and runs a genetic algorithm which is imported from tsp_ga and finds the shortest path.

tsp_ga.py file is used to solve to Travelling salesman problem using genetic algorithm