CS5331: Aerial Computing
Summer II 2023

- Project number: Project #3 – Path Planning Strategies
- Your name: Manvitha Aathmuri (R11847781)

This project is about implementing two major operations: Clustering and Path planning.
The three Path planning strategies that are discussed in this project are:

i. Random Planning path (RAND)
ii. Distance based Planning path (NNF)
iii. Density based Planning path (DF)

Let us briefly discuss on how to run the code and then get into details of the three path planning strategies.

The Matlab code for this project is written using the given *scale_free_only.m* source code. This code generates around 100 non-uniformly distributed point of interests (POI's). Using the neighbor density-based clustering algorithm, we need to scale and group the densely populated POIs in early stage and remove the number of candidates POIs to cover quickly. Once clustered, the path has been determined using the above three strategies. On a basic note, there is one parameter named 'path' given to choose any of the planning strategies.

Ex: The function I am calling in my code is scale_free_only()

scale_free_only(0) is given in the command window and the code runs to plot the graph for RAND.
Similarly **scale_free_only(1) & scale_free_only(2)** are given to plot NNF and DF graphs.

The three path planning strategies that are implemented in this project are,

Random Path Planning (RAND):

In the Random approach, the drone picks a scan point at random and moves in that direction from where it is now. When it arrives at the selected scan point, it again picks a scan point at random and moves in that direction. This procedure is repeated until all scan points have been visited by the drone, or until all POIs have been scanned. The path the drone takes may not be the most effective, and it could result in longer trip lengths and times compared to alternative tactics because the selection is completely random. Even though it is not optimal for efficiency, the randomization may allow for better coverage of various places and may be helpful when investigating the entire area is necessary.

Distance-based Planning / Nearest Neighbor First (NNF):

The drone chooses the scan point that is closest to its present position and flies towards it while using the Nearest Neighbor First technique. It then chooses another scan point that is the nearest to its present location among the still-unvisited scan points after arriving at the first one it had chosen. The drone continues doing this until all POIs have been surveyed and all scan sites have been visited. The NNF method prioritizes the closest scan spots first in order to save trip distance and time. When the POIs are dispersed around the area, this technique typically works well, and proximity is an important consideration for effective scanning.

Density-based Planning (DF):

The drone chooses the scan point that covers the greatest number of POIs while using the Density First technique. After arriving at the chosen scan point, it chooses the following scan point that, among the remaining unvisited scan sites, covers the second-highest number of POIs. The drone keeps doing this until all POIs have been surveyed and all scan sites have been visited. When there are clusters of significant POIs localized in particular places, the DF method favors covering areas with higher POI density first. It seeks to maximize the total amount of coverage provided by POIs and might work better when the distribution of POIs is not random.

Let's get through the brief analysis of what this program does.

1. Create Random POIs:
   The first thing the code does is create a collection of random POIs in a 2D network region with a given width (w_begin to w_end) and height (h_begin to h_end). Both the coverage radius (r) and the number of POIs (n) are given.
2. Scale-Free Distribution:
   To simulate real-world phenomena where certain areas have higher POI density than others, the positions of the POIs are created using a scale-free distribution. The distribution is managed by the clustering exponent (alpha).
3. DBSCAN Clustering:
   The generated POIs are grouped together based on how close they are to one another using the DBSCAN algorithm. In addition to forming clusters of nearby POIs, it finds core points (POIs with an adequate number of neighbors within a given distance). Spatially adjacent POIs are grouped together with the aid of the algorithm.
4. Path Planning and Visualization:
   The algorithm uses three distinct path planning strategies (RAND, NNF, and DF) to decide the order of drone visits to all scan sites (POIs) after clustering the POIs. To scan every POI, the drone launches from its base at (0, 0) and soars toward the scan sites in the middle of each coverage area. Each POI is represented as a triangle on the pathways, and a circle with a radius of 100 meters is drawn around each POI.

Results and Graphs:

Below is the graph of POIs generated.

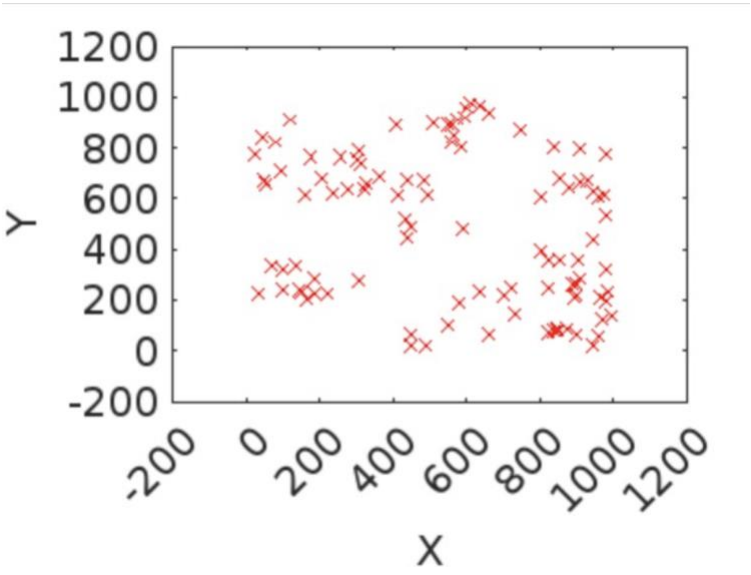

Fig (1)

With the help of neighbor density-based clustering algorithm, we are efficiently covering all the POIs according to density and representing it in the below shown way.

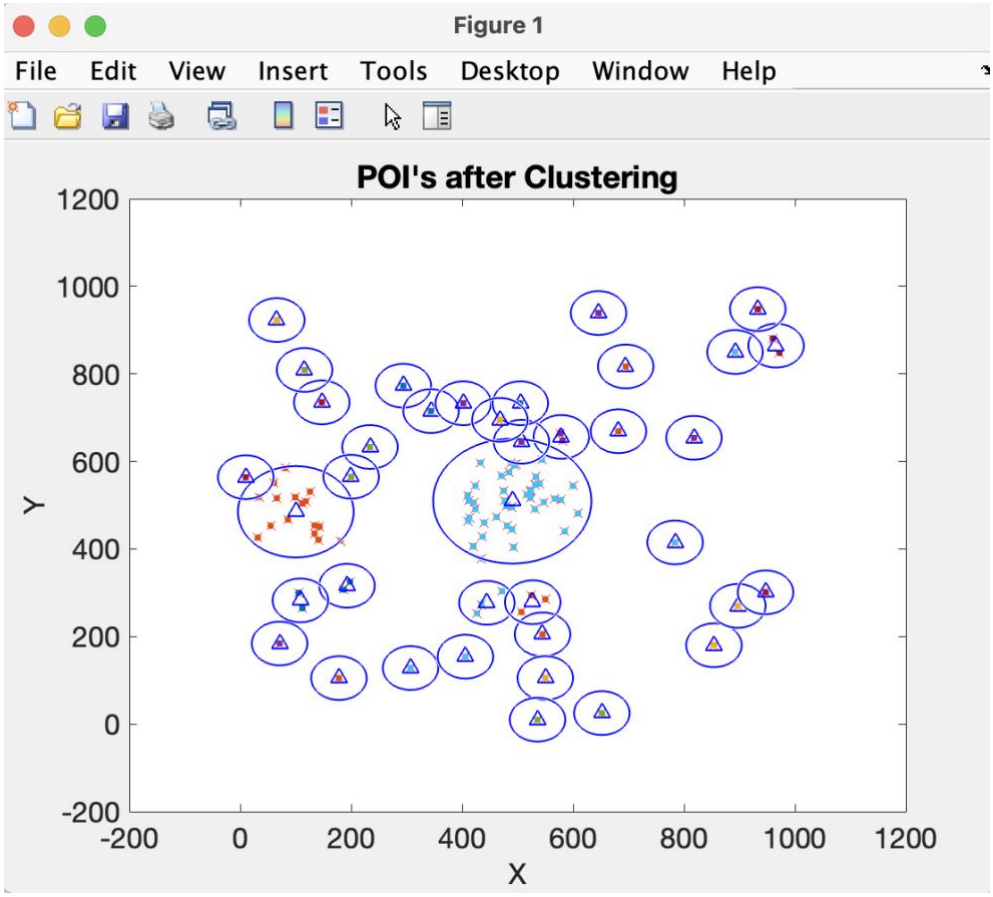

Fig (2)

Below is the screenshot of the individual routes taken by drone using all three path planning strategies along with the total distance covered.

RAND: When the path strategy is 0, it generates Random path graph as shown below.

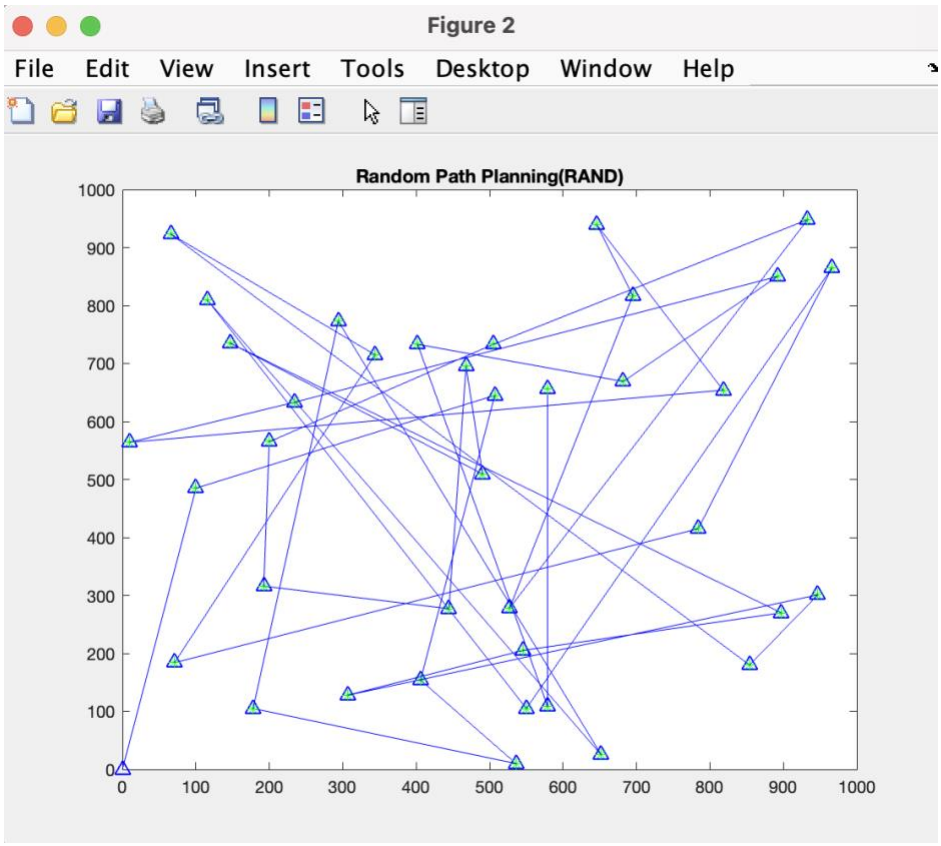
Figure 2
File Edit View Insert Tools Desktop Window Help
Random Path Planning(RAND)
1000
900
800
700
600
500
400
300
200
100
0
0 100 200 300 400 500 600 700 800 900 1000

Fig (3.a)

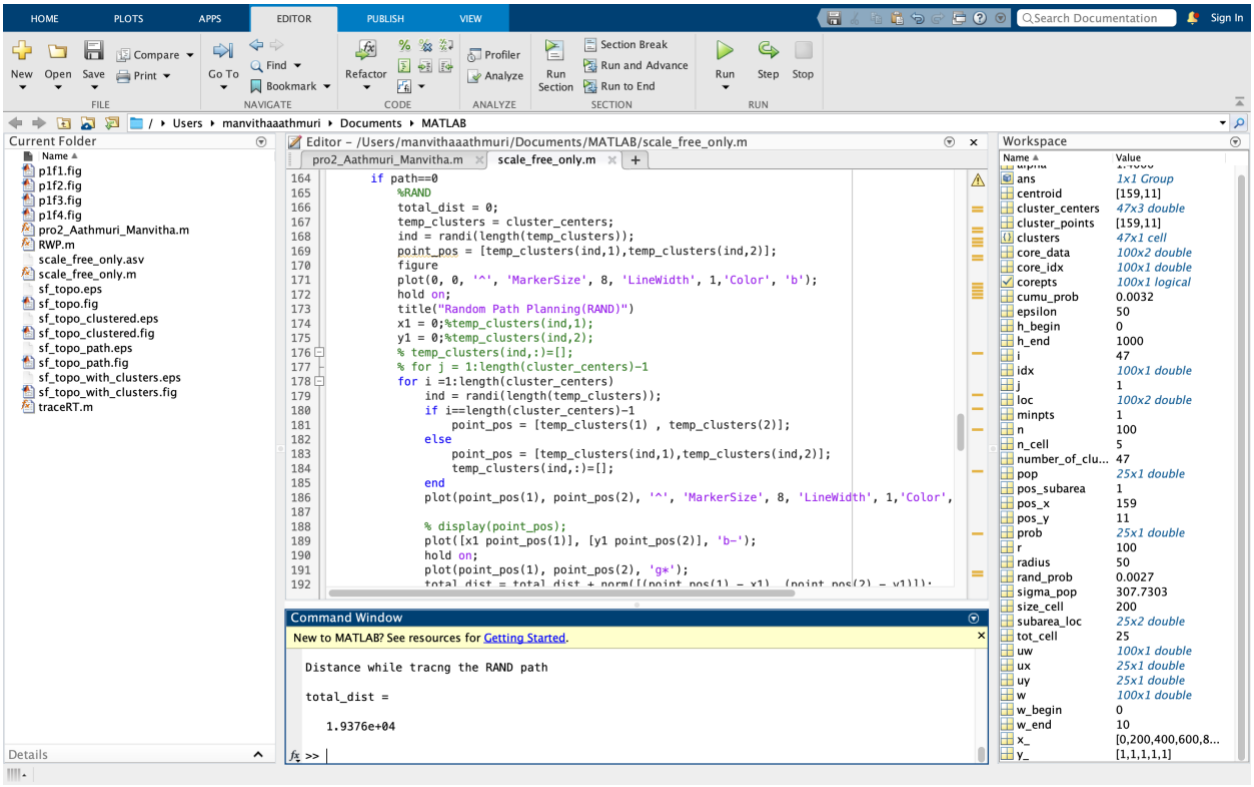
HOME PLOTS APPS EDITOR PUBLISH VIEW
Search Documentation
Sign In
New Open Save Compare Print Go To Find Bookmark Refactor Profiler Analyze Run Section Section Break Run and Advance Run to End Run Step Stop
FILE NAVIGATE CODE ANALYZE SECTION RUN
Users manvithaaathmuri Documents MATLAB
Current Folder
p1f1.fig
p1f2.fig
p1f3.fig
p1f4.fig
pro2_Aathmuri_Manvitha.m
RWP.m
scale_free_only.asv
scale_free_only.m
sf_topo.eps
sf_topo.fig
sf_topo_clustered.eps
sf_topo_clustered.fig
sf_topo_path.eps
sf_topo_path.fig
sf_topo_with_clusters.eps
sf_topo_with_clusters.fig
traceRT.m
Details
Editor - /Users/manvithaaathmuri/Documents/MATLAB/scale_free_only.m
pro2_Aathmuri_Manvitha.m
scale_free_only.m
if path==0
%RAND
total_dist = 0;
temp_clusters = cluster_centers;
ind = randi(length(temp_clusters));
point_pos = [temp_clusters(ind,1),temp_clusters(ind,2)];
figure
plot(0, 0, '^', 'MarkerSize', 8, 'LineWidth', 1,'Color', 'b');
hold on;
title("Random Path Planning(RAND)")
x1 = 0;%temp_clusters(ind,1);
y1 = 0;%temp_clusters(ind,2);
% temp_clusters(ind,:)=[];
% for j = 1:length(cluster_centers)-1
for i =1:length(cluster_centers)
ind = randi(length(temp_clusters));
if i==length(cluster_centers)-1
point_pos = [temp_clusters(1) , temp_clusters(2)];
else
point_pos = [temp_clusters(ind,1),temp_clusters(ind,2)];
temp_clusters(ind,:)=[];
end
plot(point_pos(1), point_pos(2), '^', 'MarkerSize', 8, 'LineWidth', 1,'Color',
% display(point_pos);
plot([x1 point_pos(1)], [y1 point_pos(2)], 'b-');
hold on;
plot(point_pos(1), point_pos(2), 'g*');
Command Window
New to MATLAB? See resources for Getting Started.
Distance while tracng the RAND path
total_dist =
1.9376e+04
>>
Workspace
Name Value
ans 1x1 Group
centroid [159,11]
cluster_centers 47x3 double
cluster_points [159,11]
clusters 47x1 cell
core_data 100x2 double
core_idx 100x1 double
corepts 100x1 logical
cumu_prob 0.0032
epsilon 50
h_begin 0
h_end 1000
i 47
idx 100x1 double
j 1
loc 100x2 double
minpts 1
n 100
n_cell 5
number_of_clu... 47
pop 25x1 double
pos_subarea 1
pos_x 159
pos_y 11
prob 25x1 double
r 100
radius 50
rand_prob 0.0027
sigma_pop 307.7303
size_cell 200
subarea_loc 25x2 double
tot_cell 25
uw 100x1 double
ux 25x1 double
uy 25x1 double
w 100x1 double
w_begin 0
w_end 10
x_ [0,200,400,600,8...
y_ [1,1,1,1,1]

Fig (3.b)

NNF: When the path strategy is 1, it generates Distance-based/Nearest neighbor path graph as shown below.

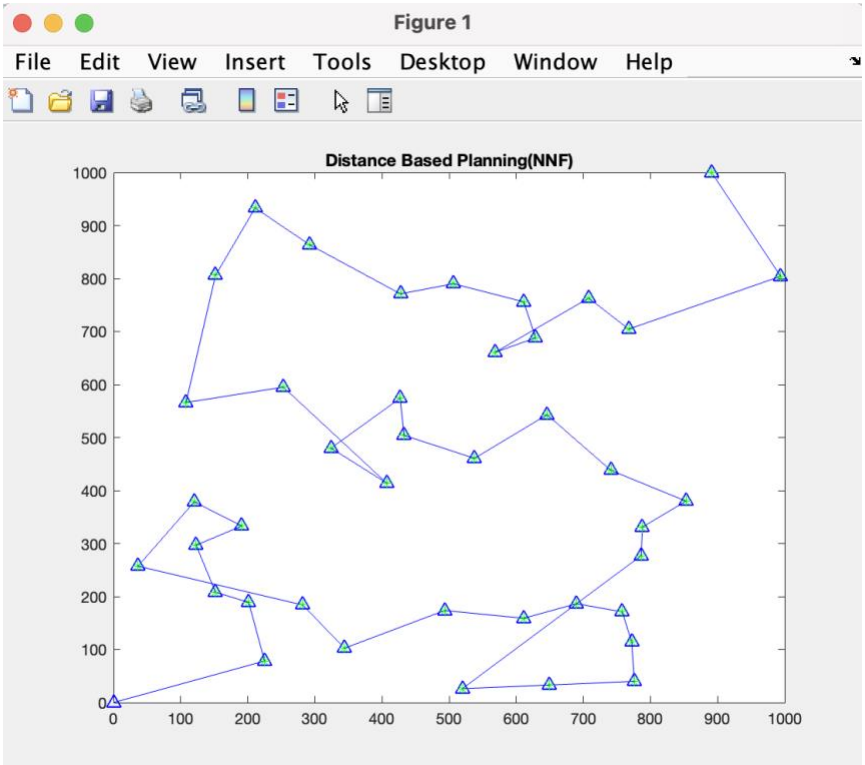

Fig (4.a)

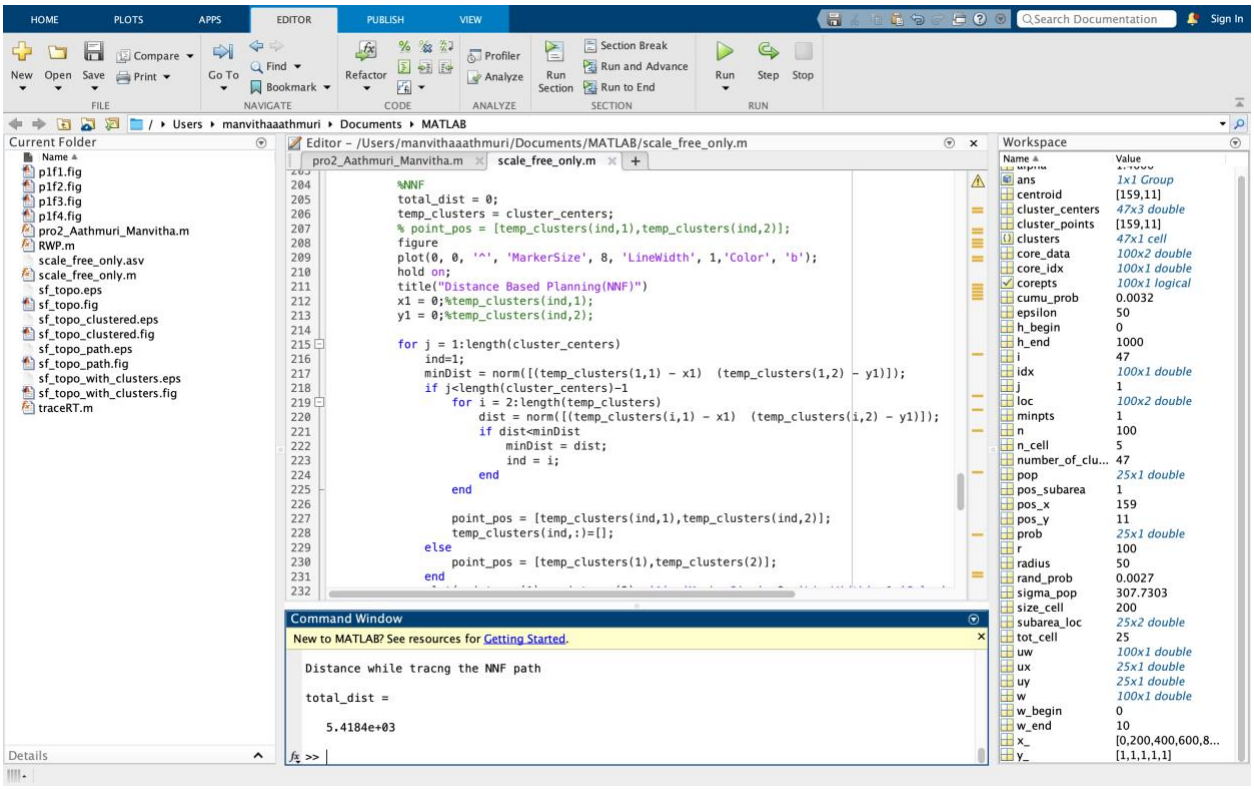

Fig (4.b)

DF: When the path strategy is 2, it generates Density-based path graph as shown below.

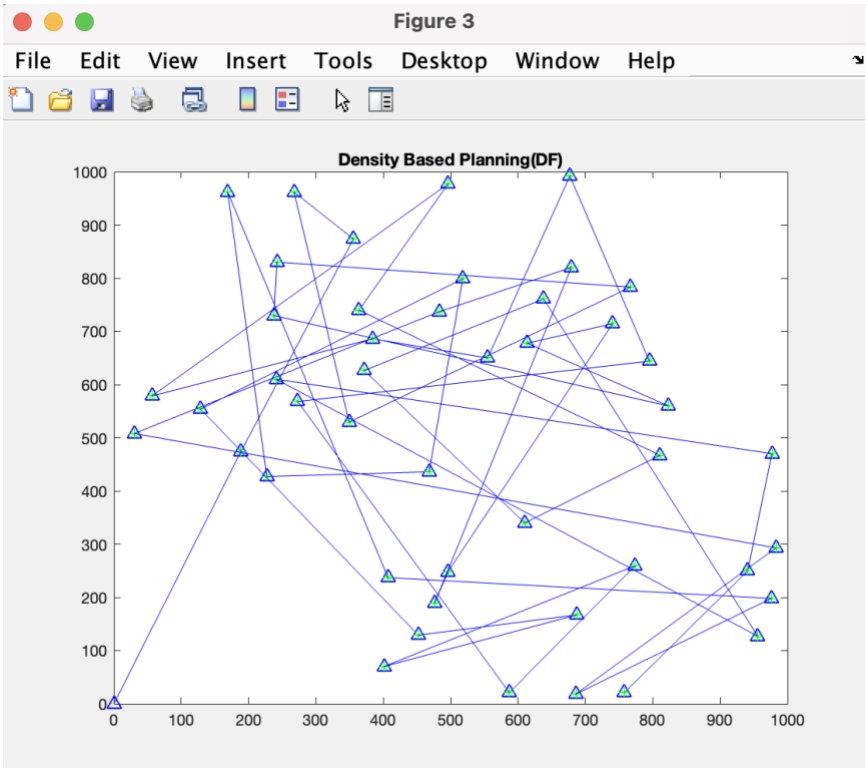

Fig (5.a)

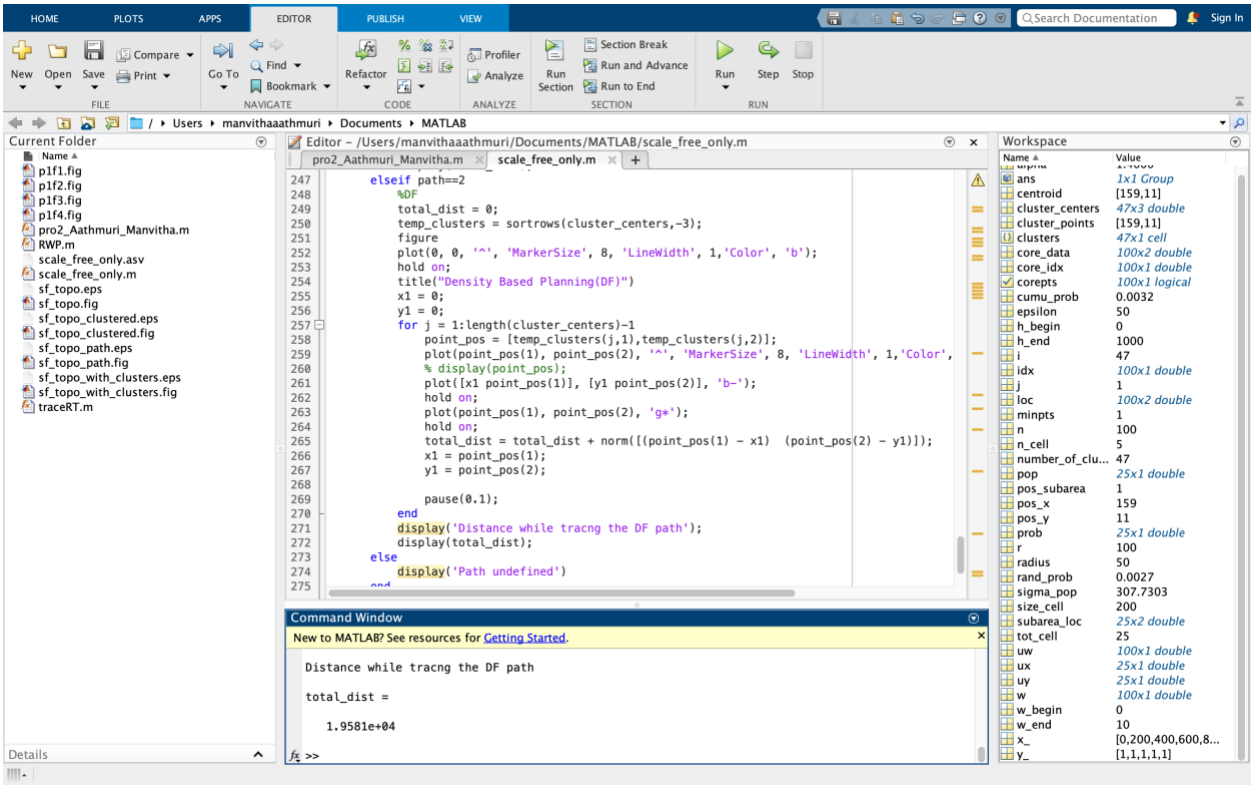

Fig (5.b)

Observations:

As discussed above, the graph depicts two major operations such as clustering and path planning.
Fig (1) represents the 100 POIs on a single graph.
Fig (2) shows us the clustering of all the POIs.
Fig (3.a & 3.b) traces the drone path when a random planning path is used and calculates the total distance the drone covered.
Fig (4.a & 4.b) traces the drone path when a distance-based planning path is used and calculates the total distance the drone covered.
Fig (5.a & 5.b) traces the drone path when a density-based planning path is used and calculates the total distance the drone covered.

In conclusion, each path planning technique has advantages and disadvantages:

i. Random (RAND): Easy to use and might improve overall exploration, but not tuned for efficiency.
ii. Reduces travel time and distance, making Nearest Neighbor First (NNF) ideal for locations where proximity is essential for effective scanning.
iii. When significant POIs are grouped together, density first (DF) maximizes POI coverage and gives the highest priority to those areas.