Project Title: Path Planning Strategies

Overview:
The script generates target points in a specified area and then explores three different traversal strategies to visit these points.

Here's a step-by-step explanation of the code:

1. **Setting Up the Environment**: The script sets up the parameters for the simulation, such as the network size (width and height), the number of cells, the number of target points (n), the coverage area radius (r), and the clustering exponent (alpha) for the scale-free distribution.

2. **Generating Target Points**: The script generates a set of n target points distributed across a 2D area. The distribution of these points follows a scale-free distribution, with clustering defined by the clustering exponent alpha.

3. **Finding Neighbours**: For each target point, the script calculates the number of neighbouring points within a radius r. These Neighbours are stored in the variable `loc2`.

4. **Sorting Points by Neighbours**: The target points are sorted in descending order based on the number of Neighbours they have. The sorted points are stored in the variable `loc4`.

5. **Visualizing the Target Points**: The target points are plotted with blue 'x' markers, and circles are drawn around each target point to represent the coverage area with a radius of r. This visualization provides an overview of the distribution of target points and their clustering.

6. **Selecting Traversal Strategy**: Depending on the value of `pp_strategy`, one of the three strategies is executed:

- **Strategy 0 (RAND)**: Randomly selects the next point from the target points and plots the traversal path as a sequence of red lines connecting the selected points. The traversal path is chosen randomly.
- **Strategy 1 (NNF)**: Uses the nearest neighbour algorithm to find the next nearest point to visit from the current point. The traversal path is plotted accordingly. The nearest neighbour algorithm tries to minimize the total traversal distance.
- **Strategy 2 (DF)**: Directly plots the target points in the order they were generated, without considering any specific traversal strategy. This strategy likely represents a straightforward approach.

7. **Visualizing the Traversal Path**: The traversal path for the selected strategy is visualized on a separate figure. The path is represented as a red line connecting the target points in the order they are visited.

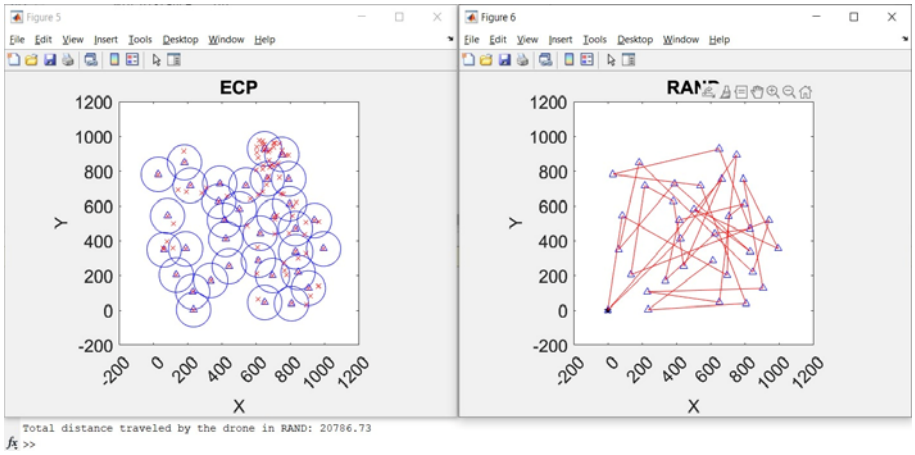
8. **Final Output**: A final figure shows the target points along with the traversal path for the chosen strategy. The title of each figure indicates which strategy is being visualized. The total flying distance from base, left and bottom of the network (0,0).

Since this is a simulation, the effectiveness of the traversal strategies would depend on the specific characteristics of the generated target points, such as their distribution, clustering, and density. The script allows us to explore how different strategies perform in visiting the target points, based on the given distribution and network parameters.

Instructions on how to run the program:
1. Open the file proj_2.m in matlab.
2. Run the program with function proj_2(0):
   - It runs with **RAND** strategy and plots two graphs (clustering & drone path) and prints total flying distance.
3. Run the program with function proj_2(1):
   - It runs with **NNF** strategy and plots two graphs (clustering & drone path) and prints total flying distance.
4. Run the program with function proj_2(2):

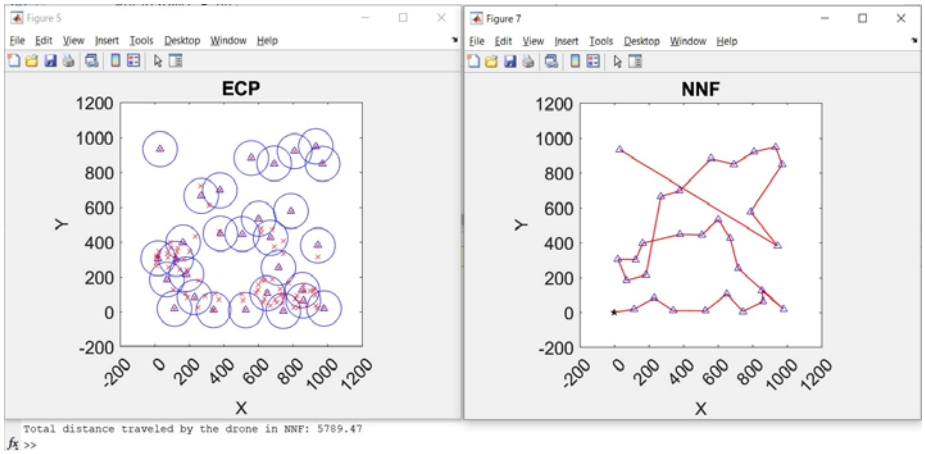
- It runs with **DF** strategy and plots two graphs (clustering & drone path) and prints total flying distance.

Results:
Proj_2(0): RAND



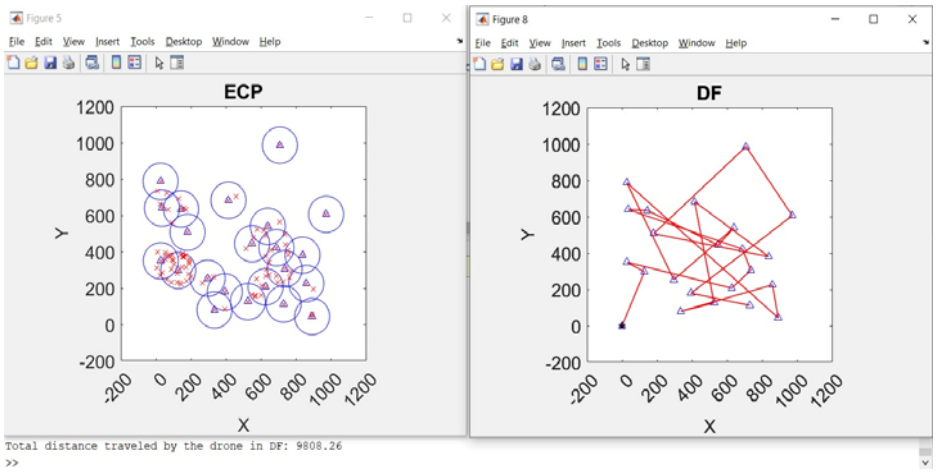
The points are selected randomly from a set of target points. After visiting one point, the next point is visited randomly chosen from unvisited points.

Proj_2 (1): NNF



In Nearest Neighbour, the points are selected from nearest unvisited points from the current point.

Proj_2(2): DF



In Density-Based Planning, the points are selected with highest number of neighbours with a specific radius. This approach explores regions of higher target point density.

- Random Planning is simple and useful for basic exploration when the primary concern is not finding the most optimal path.
- Nearest Neighbour Planning is a reasonable balance between simplicity, efficiency and having least distance, well-suited for uniform distributions of target points.
- Density-Based Planning is effective for exploring dense regions and clusters of target points but may not be as suitable for uniform or sparse distributions.

- For RAND Planning, we got flying distance of **20786.73 meters**.

- For Nearest Neighbour Planning, we got flying distance of **5789.47 meters**.

- For Density-Based Planning, we got flying distance of **9808.26 meters**.

Note: Two graphs will be generated one on top of the other.