1. Are the generated characteristic values for the nodes in the simulation integers or floating-point numbers?

We will use integer coordinates to make the grid formulation more discrete.

3. How should we handle cases where two nodes have the same coordinates? Should this be treated as an impossible case, or should we prevent nodes from having identical coordinates?

You should prevent any node from having an already allocated coordinates. You can do that dynamically as you are generating the nodes.

4. Should we calculate simple Euclidean distances between nodes, or is there another metric we should use?

Simple Euclidean distances is what we will physically use to determine the distance between any two nodes in the network.

5. How should we resolve the tie for the cluster head election process if two or more nodes have identical election formula values and are equally distant from the cluster center? Should we select a node at random?

Usually in routing protocols if a tie exists based on all the requirements, selecting a head at random is the best way to break the tie.

6. In "greedy intuitive routing," a node always routes to the nearest node within its range closest to the destination. If no nodes within range are closer to the destination, that packet should be dropped as 'no route available,' correct?

Absolutely correct.

7. How should we handle routing loops? Should the simulation return "no route found" if it loops between two nodes, or should we prevent returning to previously visited nodes and choose the second closest node to the destination? I only ask as it would seem a violation of the constraint to use greedy intuitive routing if I understand the meaning correctly.

To prevent a loop, we should avoid visiting a nodes more than one time.

8. What does it mean for the program to "use a clean copy of network.txt" with every run? Should we reset or recreate the file after each simulation run? Does this refer to each cycle in a run before the user exits or a fresh network.txt each time 'main.pyhttp://main.py/' is run? Or should each run output a unique network.txt, such as sequentially numbered or such?

This simply means that you should fill the network.txt file from scratch every time you run main.py.

9. When outputting data to network.txt, should the format match input.txt exactly, or is there flexibility to include additional information like cluster head details and residency in a different format?

Try to use the same input.txt format to make the automation of grading easier. If you want to shift a bit from the format that’s fine but include a specification of your format in the Readme file.

10. Are any of the node characteristics dynamic in the simulation, such as energy depletion or power drop during operation?

For simplicity we don’t include such dynamic behavior in the simulation.

11. Are there specific conditions other than user exit commands that terminate the simulation? For example, should the simulation automatically end if all nodes lose energy?

This feature is not required. You can add it if you find the time to do it.

1. Are the generated characteristic values for the nodes in the simulation integers or floating-point numbers?
2. If the characteristics are floats, are there specific requirements for the number of significant digits? My random generator has unlimited digits, but I would like to use some cutoffs to keep the numbers shorter.
3. How should we handle cases where two nodes have the same coordinates? Should this be treated as an impossible case, or should we prevent nodes from having identical coordinates?
4. Should we calculate simple Euclidean distances between nodes, or is there another metric we should use?
5. How should we resolve the tie for the cluster head election process if two or more nodes have identical election formula values and are equally distant from the cluster center? Should we select a node at random?
6. In "greedy intuitive routing," a node always routes to the nearest node within its range closest to the destination. If no nodes within range are closer to the destination, that packet should be dropped as 'no route available,' correct?
7. How should we handle routing loops? Should the simulation return "no route found" if it loops between two nodes, or should we prevent returning to previously visited nodes and choose the second closest node to the destination? I only ask as it would seem a violation of the constraint to use greedy intuitive routing if I understand the meaning correctly.
8. What does it mean for the program to "use a clean copy of network.txt" with every run? Should we reset or recreate the file after each simulation run? Does this refer to each cycle in a run before the user exits or a fresh network.txt each time 'main.pyhttp://main.py/' is run? Or should each run output a unique network.txt, such as sequentially numbered or such?
9. When outputting data to network.txt, should the format match input.txt exactly, or is there flexibility to include additional information like cluster head details and residency in a different format?
10. Are any of the node characteristics dynamic in the simulation, such as energy depletion or power drop during operation?
11. Are there specific conditions other than user exit commands that terminate the simulation? For example, should the simulation automatically end if all nodes lose energy?