# Map Visualization

To visualize the solutions produced by the various algorithms, we implemented a real and interactive geographic map with the sensors and the base station. This was done using Folium, a Leaflet JavaScript library which we interacted with through an API. We followed through the API documentation (Story 2013) to achieve the following:

1. Show the best route obtained from a sensor to a base station by a particular algorithm
2. Show all other possible routes found (not the best but have end-to-end connectivity)

The locations of the sensors were provided in form of X, Y grid coordinates in meters for purposed of calculating their heuristic distances. However, since the API needs values of geographic coordinates, coordinates of the sensors had to be computed and represented on the leaflet. The computation of geographic coordinates from X, Y grid coordinates was inspired by the Haversine formula (Scripts 2024).

latitude = central latitude + y/110574

longitude = central longitude + x/(11320 \* cos(central latitude))

where x and y are the x, y values of the sensors provided in the csv file while central latitude and central longitude are the (0, 0) coordinates on the grid from which all other sensor distances are calculated from. From the map provided in the graphic in the assignment instructions we estimated the geographic locations of the base stations in Lyndhurst and Beaulieu as shown in figure 1 by referring to google maps.

A map of a route

Description automatically generated

Figure 1: Search Areawith sensors and base stations

The values ‘110574’ and ‘11320’ are the distances between the latitudes and longitudes in meters respectively (Rosenberg 2024). With the geographic coordinates derived from the grid coordinates, we were able to feed the API with the sensor and base station information which was useful in plotting paths.

A map with blue dots

Description automatically generated

Figure 2: Search area on map created using Folio

## Module Functionalities

Data Fetching and constant variable initialization: The X, Y grid data is fetched from the CSV file and all constants related to the data such as the ‘distance bandwidth data’ defined in the instructions are initialized. It is at this point that a blank map in the search area of interest is initialized.

Grid to Geographic coordinates conversion: Using the formulas defined above, the X, Y data is converted to geographic coordinates using the function `get\_geographic\_coordinates()`

Initialization of a map with sensors: Using the computed geographic coordinates, the sensors and base stations are represented on the blank map. This initialization happens every time an algorithm is executed for visualization to be done on a fresh map. This is done using the `init\_blank\_map()` function.

Plotting routes: Joins the sensors that are part of a transmission path using lines from the origin node to the destination node. This is done for the best path found which is represented using a yellow line and all the other possible paths that are represented using a variety of colors. The ‘best path’ and ‘possible paths’ have been categorically layered in a way that you can view each type independently without struggle. For further visual intuition, the best path’s sensors have been marked with pins which show their sensor IDs. This can be done using a layer controller found on the top left corner. This functionality has been implemented in the `plot\_route()` function. With takes in the coordinates as arguments with a specification of whether they are the best routes or they are possible routes.

A map with points and lines

Description automatically generated

Figure 3: Representation of the various paths on the leaflet

## References

Rosenberg, M., 2024. *The Distance Between Degrees of Latitude and Longitude*. Available from: <https://www.thoughtco.com/degree-of-latitude-and-longitude-distance-4070616> [Accessed

Scripts, M. T., 2024. Calculate distance, bearing and more between Latitude/Longitude points. Available from: <https://www.movable-type.co.uk/scripts/latlong.html> [Accessed

Story, R., 2013. *Folium*. Available from: <https://python-visualization.github.io/folium/latest/> [Accessed

## Appendix

A map with points and dots

Description automatically generated

APPENDIX A: Map with the best path layer toggled

A map with blue dots and points

Description automatically generated

APPENDIX B: Map with possible routes only toggled