Solution Overview

Frequency Allocation for Cell Phone Towers

Code

GitHub repo: [Cell-Phone-Towers]

Introduction

This document presents a comprehensive solution for the allocation of frequencies to a list of Cell Phone Towers, taking into consideration interference reduction and efficient frequency usage. The solution is implemented in Python, and the accompanying command-line utility ensures flexibility for different sets of cell towers.

Problem-solving approach

1. **Understanding the Problem:**
   - I started by ensuring that I have a thorough understanding of the problem statement and requirements.

     Key Requirements:

     - Develop a command-line utility for frequency allocation.
     - Utilize a range of frequencies (110 to 115) to allocate unique frequencies to each cell.
     - Implement an algorithm that considers the geographical proximity of cells to prevent interference.
     - Provide flexibility for accommodating different datasets by ensuring the program's generic nature.
     - Generate a graphical representation (scatter plot) of the frequency allocation plan

2. **Research and Analysis:**
    - **Easting and Northing:**
        - Easting and northing are coordinates used in a Cartesian coordinate system to represent locations on the Earth's surface. Easting measures the distance eastward from a specified reference point, while northing measures the distance northward.
        - These coordinates provide a two-dimensional representation of a location in relation to a specific origin point.

    - **Longitude and Latitude:**
        - Longitude and latitude are angles that define a point's location on the Earth's surface. Longitude measures the east-west position, while latitude measures the north-south position.
        - These coordinates offer a spherical representation of a location, crucial for precise global positioning.

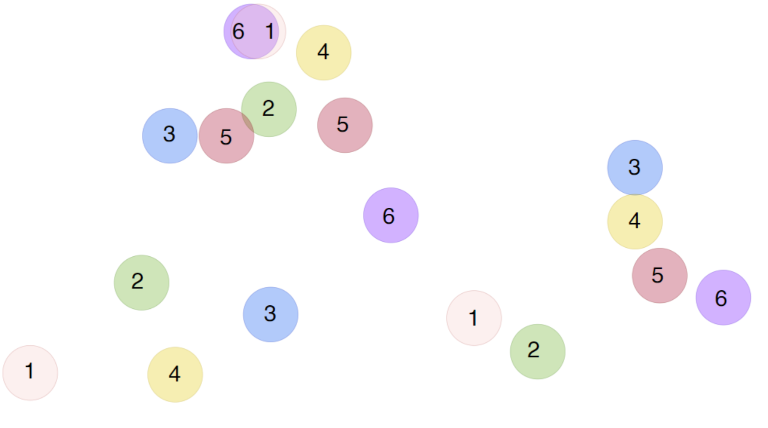
        The aim is to demonstrate a frequency allocation plan rather than conducting a precise geographic analysis, in my approach I plotted longitude and latitude on a 2D graph. Longitude and latitude are spherical coordinates, and mapping them directly to a flat plane can introduce distortions. If the scope of the project expands to cover larger areas or if more precise geographic representations are required, I need to consider incorporating appropriate map projections.

3. **Algorithm Design:**

Possible Solutions

- Solution 1: Simple West-to-East Allocation

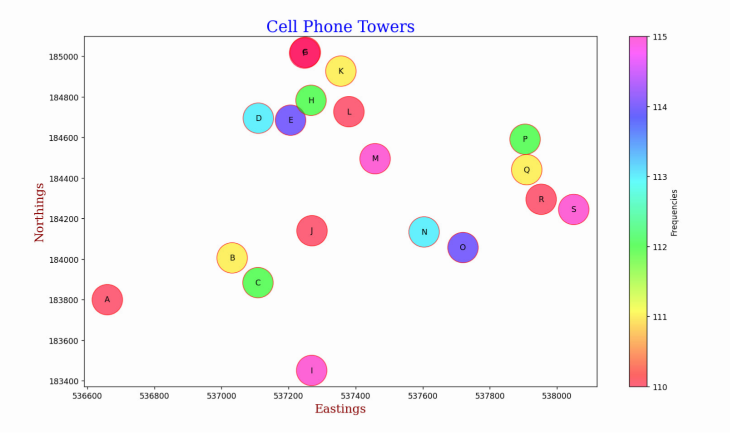
With a sorted list from west to east. Allocate the frequencies from 110 to 115 to the first six nodes in the list, then continue in the list with groups of sixes to allocate the frequencies from 110 to 115.



This sequential process optimizes frequency utilization, minimizing unused channels. However, it's important to note that this approach does not guarantee absolute interference reduction, as neighboring towers may end up sharing the same frequency in certain scenarios. Future iterations could explore additional optimizations for interference reduction based on proximity.

- Solution 2: Improved West-to-East Sorting

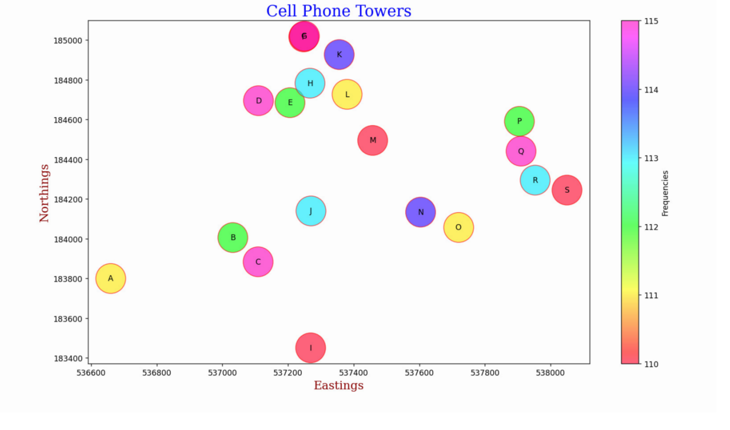
With a sorted list from west to east, this solution initiates the sorting process based on proximity to the first cell in the list. The sequential order is established from the nearest to the farthest. Initially, frequencies ranging from 110 to 115 are allocated to the first six cells in this ordered list, ensuring unique frequencies for the westernmost cells. Subsequent frequency assignments are influenced by distances from the furthest frequency already assigned.



The approach minimizes interference by favoring frequencies at the less utilized end of the range. The algorithm dynamically tracks frequencies assigned to neighboring cells, enabling adjustments based on proximity to minimize interference. In essence, this algorithm optimizes for interference reduction based on proximity, effectively mitigating interference concerns.

- Solution 3: Centralized Sorting

Beginning with the determination of the central cell, this solution orchestrates the sorting of the list based on proximity to this central cell. The resulting sequential order spans from the nearest to the farthest. Initial frequency allocations from 110 to 115 are reserved for the first six cells in this ordered list, ensuring unique frequencies for the centrally located cells. Subsequent frequency assignments are determined by distances from the furthest frequency already allocated.



Similar to Solution 2, this approach seeks to minimize interference by favoring frequencies at the less utilized end of the range. The algorithm dynamically tracks frequencies assigned to neighboring cells, facilitating adjustments based on proximity to minimize interference. Notably, this algorithm strikes a balance between central and peripheral cells, optimizing frequency allocation. Its flexibility allows it to operate with unsorted lists, enhancing its adaptability to diverse input scenarios.

4. **Implementation:**
    - I choose to implement solution 3 using Python.
5. **Testing**
    - I tested the solution with different sample data to ensure the correctness of the whole program.
6. **Documentation:**
    - I added Source Code Comments and a Readme file on GitHub to help with setup.
7. **Continuous Improvement:**
    - The algorithm can be further improved, if the range/distance of the frequency of each tower is known

Solution Components

1. **Distance Calculation:**
    - I began by implementing a function to calculate the Euclidean distance between two cell towers. This distance metric is crucial for determining interference and allocating frequencies efficiently.
2. **Center Cell Identification:**
    - To establish a reference point for frequency allocation, I identified the center cell. The center cell is pivotal for optimizing interference reduction by allocating the closest frequencies to neighboring cells.
3. **Frequency Allocation Algorithm:**
    - I designed a frequency allocation algorithm that prioritizes the center cell and allocates the closest frequencies to neighboring cells. The algorithm ensures that the same frequency is not assigned to cells in close proximity, minimizing interference.
4. **User Input and File Handling:**
    - The command-line utility prompts the user to select a file containing cell tower data. It provides flexibility for different datasets. Additionally, the user can choose between Easting/Northing and Latitude/Longitude coordinate systems.
5. **Graphical Representation:**
    - The solution includes a graphical representation of the cell towers, displaying their locations and allocated frequencies. I utilized matplotlib for plotting the scatter plot.

Execution Steps

1. Run the program.
2. Choose the dataset file (given-data-1.txt, sample-data-2.txt, or sample-data-3.txt).
3. Select the coordinate system (Easting/Northing or Latitude/Longitude).
4. View the scatter plot with frequency allocation or the list of towers with their frequencies.

Conclusion

The implemented solution successfully addresses the challenge of allocating frequencies to cell towers, considering interference reduction and efficient frequency usage. The command-line utility provides adaptability for different datasets, and the graphical representation enhances the understanding of the frequency allocation plan.

Contact

If you have any questions or comments, please reach out to me.

Albert Willemse	albertjordaanw@gmail.com	081 828 8838