Extending Existing Electrical Infrastructure to the Cell Tower Network

Nigel S. Paine, CSE, SuoAn Gao, EE

***Abstract*—Through this data study we created a Python algorithm that was able to extract power grid data from a given image. The extracted power grid was then mapped to an array of longitude/latitude coordinates which we then used to find the length of fiber optic cable necessary to directly connect all local cell towers to the power grid. The goal of this study was to explore the feasibility of power utilities providing fiber optic cell services.**

I. INTRODUCTION

Earlier this year, Whip City Fiber, established by the Westfield Gas & Electric company, began connecting residential homes in Westfield, Massachusetts with gigabit speed fiber optic services [1]. This new idea that a power utility could now provide internet and phone services is what sparked us to explore the feasibility of connecting a pre-existing electrical grid to a cell tower network. While this idea seems to only just be surfacing, it has been in existence for a while. In 2003, IEEE communications magazine discussed having power lines transmit data such that residents could simply plug into a power outlet and have internet access [2].

While transmitting high amounts of data would perhaps not be so feasible through power lines inherently, the benefit to having an existing power grid infrastructure is that many infrastructures already incorporate separate fiber optic communication lines between relay stations [3]. Tapping into these lines, we can then extend the power grid to surrounding cell towers to enable power utilities to have access to cell tower communications, potentially reducing the costs of implementing an entirely new communication infrastructure.

II. DATA & ALGORITHM

The main algorithm was written in Python 3 and employed several libraries including: *numpy, cv2, pandas, matplotlib,* and *scipy*. Where numpy is the most efficient matrix calculation library in Python, cv2 is an image processing library. In addition, pandas is the most widely used data frame processing library, matplotlib is a mathematical plotting library, and finally but not least, is the scipy library, which we use to implement our minimum spanning tree algorithm.

To be able to calculate the minimized length of fiber optic cable needed to connect from all of the cell towers to the nearest power grid line, we firstly need to prepare our data. Using a 4.9 GB dataset from OpenCelliD, an open source database for cell tower info, our priority is to pick the right dataframe to use which includes how to find the position of desired cell towers in a tiny segment of power grid [4].

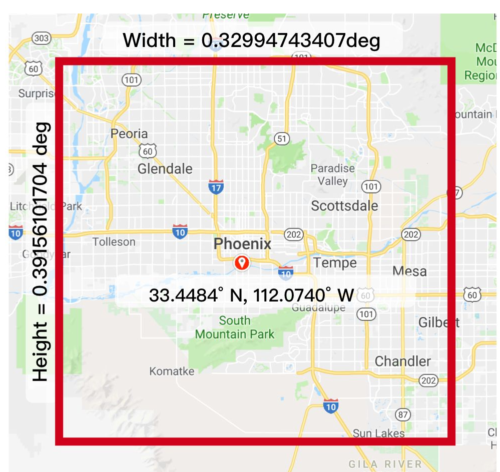

Fig. 1. An example of the selected Phoenix area

Let's use Phoenix, Arizona as an example to introduce how we select our desired data entries. In our original data set, there are 14 columns and almost 40 million entries. Two of the columns are latitude and longitude. We are going to need them to figure out whether a cell tower is in Phoenix. As in Figure 1 above, we first figure out the coordinates of the city's center, in this case the center of Phoenix is 33.4484 N and 112.0740 W. In our data set, it becomes -112.0740, 33.4484. Second, we find the area of Phoenix is 1339.02 square kilometers. We assume the shape of Phoenix is a square box (it's a rather legit assumption, say if a person lives in Newton, he would still claim that he lives in Boston Area). And then we can find the width = height of this box, which is 36.5926222072 km. Next, we need to think about longitude and latitude in our earth sphere. In 33 deg of latitude, 1 degree of latitude is 110.90444 km and 1 degree of longitude is 93.45318 km. Thus, the width of the city is 0.32994743407 degrees and the height of the city is 0.39156101704 degrees. We cut half, add the result to the center coordinates, and with another half, subtract it from the center coordinates. Using the following line of code (Figure 2), we could get all of the cell towers whose longitude and latitude are within our range:

```
Phoenix_df = train_df.loc[((train_df['lat'] < 33.613373717) & (train_df['lat'] > 33.2834262831)) & ((train_df['lon'] < -111.878219491) & (train_df['lon'] > -112.269780509))]
```

Fig. 2. Python code for latitude/longitude constraints

In the same way, we could apply the data processing algorithm to our cell towers in Nairobi, Kenya.

Next, we need to find out all of the coordinate positions of a tiny segment of the power grid. Take the Nairobi power grid map for example in Figure 3. For each pixel of the power grid map, we store all of the pixels that match the color of the power grid into an array. Then, we display what is left in our power grid matrix as an extracted grid (see Figure 4).

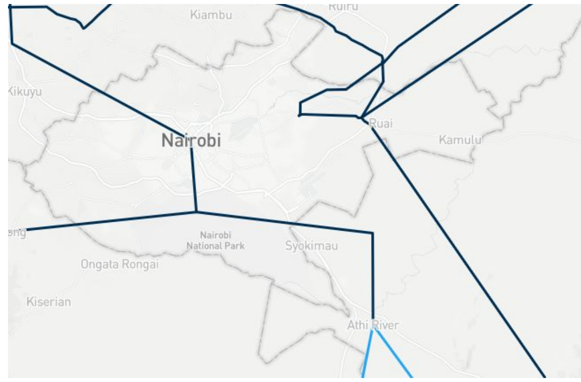

Fig. 3. Power grid map in Nairobi region [5]

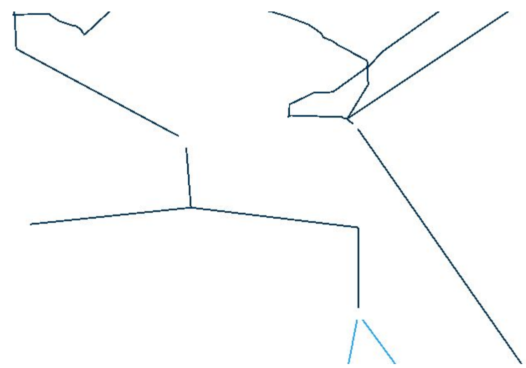

Fig. 4. Nairobi extracted power grid

Now, we have all we need. Let's dive into the algorithm we use to calculate the desired length of fiber optic if we would like to connect cell towers to a power grid. Firstly, we did some straight-forward implementation where we just find each cell towers' coordinate, calculate the distance matrix from a cell tower to all of the segments of the power grid, then find the shortest distance segment using the distance formula (see Equation 1). Each cell tower will have their own shortest distance, we sum them all, then get the shortest length of fiber optic connection.

$$Shortestdistane = \sqrt{(latcell - latgrid)^2 + (loncell - longrid)^2}$$

Eq. 1. Shortest distance formula

However, this algorithm wouldn't always be true if cell towers are too far from the power grid. In fact, in Nairobi and Phoenix, this method has a poor performance. Thus, we develop a more optimal solution:

Let's say the extracted blue lines are a part of the power grid, and the orange towers are cell towers (see Figure 5).

Fig. 5. Disconnected power grid & cell tower network (not to scale)

Take a look at the cluster of cell towers near the bottom of the power grid map. The shortest distance from the closest cell tower to the power grid is connected by the dotted blue line and the tower is now connected to the grid (represented by changing the color to blue in Figure 6). We add the length of the blue dotted line to our total length of fiber optic cable required.

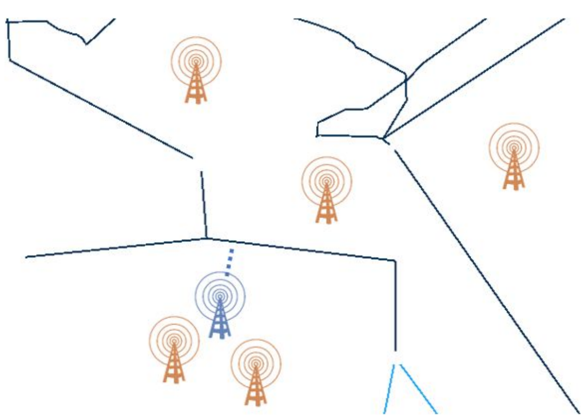

Fig. 6. Single connected cell tower

When we try to find the shortest distance from the power grid to the cell towers near the connected cell tower, instead of connecting the new cell towers directly to the power grid as our initial approach, we connect it to the previous cell tower due to a shorter distance (see Figure 7). Then, after we connect each of the cell towers, the cell towers become part of the power grid network. For each new cell tower connected to the grid the coordinates of that cell tower are stored in the power grid matrix to extend the grid's reach.

Fig. 7. Connected cell tower cluster

And as we keep following this procedure and repeat the process for all the cell towers, we will notice that the size of the grid coordinate matrix keeps increasing as the network grows. The final network looks something like Figure 8.

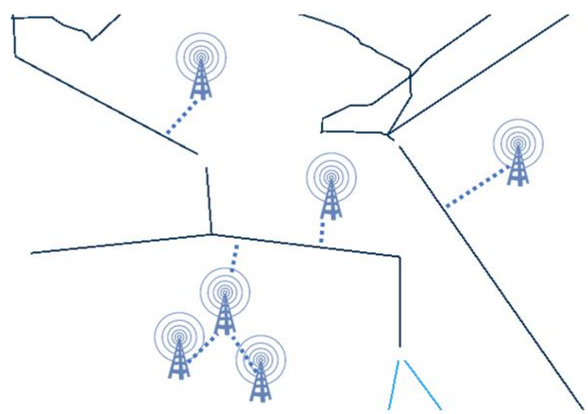

Fig. 8. Connected power grid cell tower network

Still, we need to discuss another situation: there is still a possibility that a large cluster of cell towers are close to each other, and far from the power grid. In this situation, instead of connecting the cell towers directly to the build a self-contained network. In that situation, we need to introduce the minimum spanning tree algorithm to calculate the shortest fiber optic network connection. An illustration of how the minimum spanning tree algorithm calculates the distance between cell towers can be seen in Figure 9.

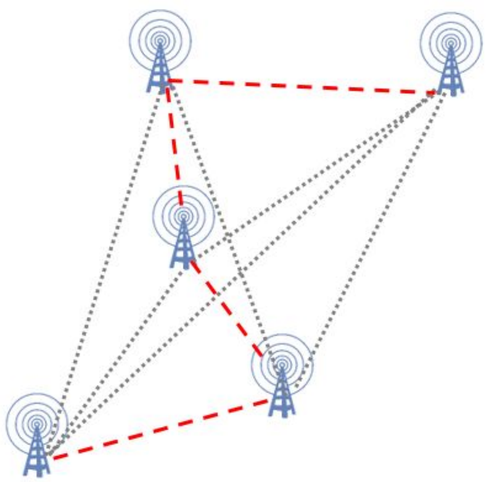

Fig. 9. Minimum spanning tree algorithm on a cell tower network

This algorithm is slightly different from the greedy method. From the beginning, it's quite similar: we pick a random point, find the shortest distance from that point to any other point. Secondly, unlike the greedy approach which just simply finds the shortest distance from the new point, we put the new point and the older points into a matrix, then we figure out the shortest distance from that point to any other points outside the matrix.

III. DATA ANALYSIS

Analyzing the data output from the algorithm, we were able to display the amount of total fiber optic cable and compare the difference in Phoenix and Nairobi. As you can see in Figure 10, the amount of fiber optic cable Phoenix requires to connect all of the cell towers (greater than 10 miles radius of coverage) in the area is roughly 350 km (or ~220 miles) while Nairobi is only about 50km (~30 miles). That's power grid, it might be more feasible to simply roughly 1/7th of the amount of fiber optic cable needed compared to Phoenix.

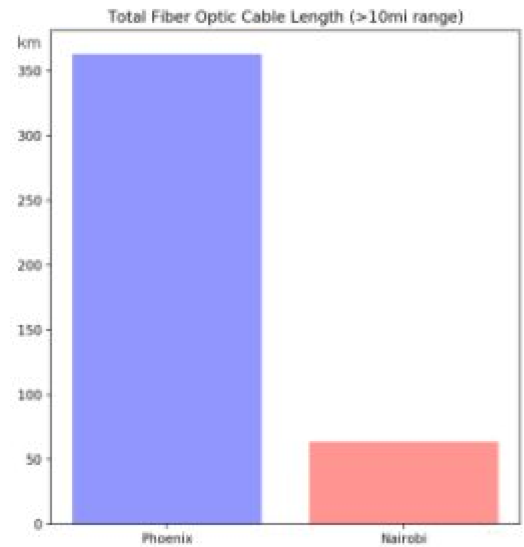

Fig. 10. Total fiber optic cable needed to connect all cell towers with >10 mi range (Nairobi v. Phoenix)

Why the drastic difference? The main factor is that Nairobi has less powerful cell towers in general with only 5 towers having a coverage radius greater than 10 miles compared to 192 towers with >10mi radius coverage in Phoenix. While there is a noticeable difference between the two cities, the comparison only samples a small amount of the cell towers within the region. As a visual example for just how dense these maps can be with cell towers, Figure 11 displays the region in Phoenix used in the algorithm.

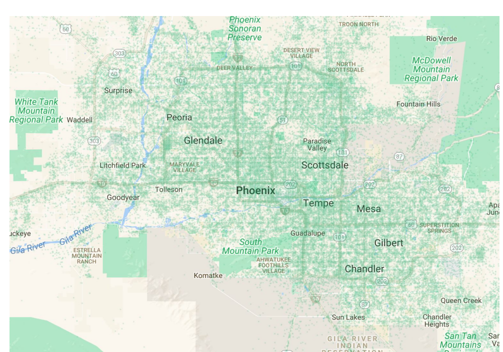

Fig. 11. Phoenix cell tower location map

Each of the green dots represent individual cell towers, numbering 33,018 total within the selected area. By comparison, there are 32, 671 cell towers total in the selected Nairobi region.

Now looking at each individual region, we compare the two potential types of connections we could have in an area: cell tower-to-power grid and cell tower-to-cell tower. The first region we analyze is Nairobi with more than 60km (~40mi) of fiber optic cable required to connect all the cell towers to the power grid, and roughly 35km (~20mi) required to solely connect the cell towers (see Figure 12).

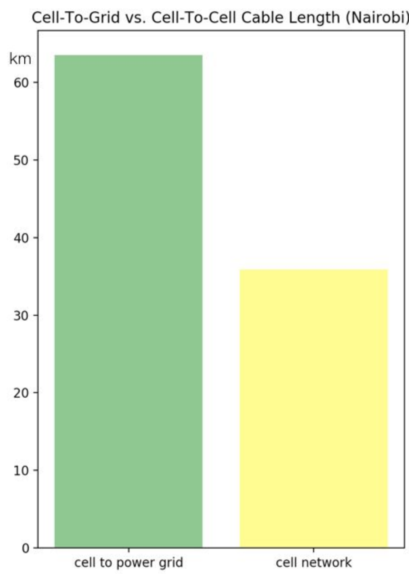

Fig. 12. Cell-To-Grid v. Cell-to-Cell fiber optic cable length comparison (Nairobi)

Observing the same comparisons in the Phoenix, AZ region we see roughly 350km (~220km) and 60km (~40mi) needed for cell-to-grid and cell-to-cell connections, respectively (Figure 13). An interesting and somewhat unexpected difference can be seen in the amount of fiber optic cable needed to connect to the grid vs directly from cell tower to cell tower. The cell tower-to-cell tower connection seems to take less distance than connecting an existing power infrastructure. A couple of factors may be at play here. Firstly, the

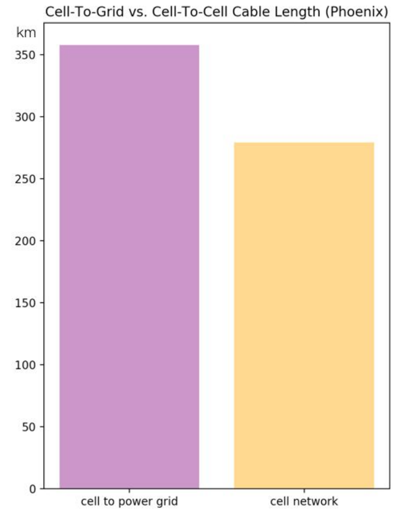

Fig. 13. Cell-To-Grid v. Cell-to-Cell fiber optic cable length comparison (Phoenix)

density of cell towers may play a big role in whether a connection to the power grid is needed. If a large cluster is dislocated from the power grid it may not be feasible for a connection to the power grid to exist. To prove this point in the future, more countries could be analyzed to see the conditions needed for the fiber optic cable connection to cost less per mile for direct connections to the power grid versus a simple cell tower network.

IV. CONCLUSION

Overall, the algorithm is in a primitive stage and can be vastly improved with a few changes. One potential improvement could be to take in GIS power grid data versus power grid image data. Using an image introduces several flaws such as location error margins and data that isn't always accurate and up-to-date. The main location errors stem from physically pinpointing on a map where the boundaries of a map image are. It may be difficult to do this physically and accurately without an image processing algorithm linked to an API such as Google Maps although this may be implemented in the future. Another improvement could be to take into account the geography of an area such as mountains, rivers, and other obstacles. This may change our results drastically, especially if there are many physical obstructions between each of the cell towers and the power grid. In tandem with this change would be the ability to calculate non-linear distances from the power grid. Instead of calculating the straight line distance from one point to another it might be more useful to calculate a route from one point to another via an open area optimal for fiber optic cable placement.

In the future, this algorithm may be used for a variety of applications not limited to fiber optic networks but any network connection given coordinate data. A further cost analysis may also be beneficial to see the improvement of using existing power grid infrastructure versus the existing cell tower network. Enabling power companies to provide the internet as a utility may change the way providers run their businesses dramatically.

REFERENCES

[1]https://www.whipcityfiber.com/

[2]http://ieeexplore.ieee.org/abstract/document/1200105/

[3]http://www.ofsoptics.com/blog/?p=168

[4]https://opencellid.org/

[5]https://www.eia.gov/state/maps.php

[6]http://www.mapsearch.com/gis-asset-data/electric-power-gis-data.html