1/10/2025

Dewald Oosthuizen

0734297951

WIM

Technical Assessment

Contents

[Optimal Cell Tower Frequency Assignment 3](#_Toc187697257)

[Background 3](#_Toc187697258)

[Initial Decision Making 4](#_Toc187697259)

[Coordinate System 4](#_Toc187697260)

[Distance Calculations 5](#_Toc187697261)

[Programming Language 5](#_Toc187697262)

[Application Development 5](#_Toc187697263)

[Input Functionality 5](#_Toc187697264)

[Graph Creation 8](#_Toc187697265)

[Distance Calculation 9](#_Toc187697266)

[Graph Illustration 10](#_Toc187697267)

[Frequency Assignment 11](#_Toc187697268)

[Frequency Assignment Correctness 13](#_Toc187697269)

[Conclusion 16](#_Toc187697270)

[References 17](#_Toc187697271)

Table of Figures

Figure 1. Input Data 6

Figure 2. Cell Tower Class 6

Figure 3. Frequency Range Input 7

Figure 4. N Closest Towers 7

Figure 5. Graph Creation Code 8

Figure 6. Haversine Formula (Dauni et al., 2019:3) 9

Figure 7. Distance Matrix 9

Figure 8. Full Graph Network 10

Figure 9. Frequency Calculation Code 11

Figure 10. Frequency Assignment 12

Figure 11. Closest 4 and 5 Towers 13

Figure 12. Correctness 14

Figure 13. Manual Frequency Verification 15

# Optimal Cell Tower Frequency Assignment

This project was initiated in response to a task provided by WIM Technologies to assess a candidate’s potential in software development (Technologies, 2025). This project required the creation of a simple application to optimally assign frequencies to a range of Cell Tower, with the goal of assigning frequencies in such a manner that interference is minimised between Cell Towers (Technologies, 2025).

To accomplish this goal, a console application was developed that offered functionality ranging from dynamic input, allowing any number of Cell Towers to be evaluated, to the visualisation of distance calculations and the overall Cell Tower network. It is important to note the Cell Tower network will be illustrated using a graph, as per the project’s requirements.

The rest of this paper will document the process of analysing the given Cell Tower information and the creation of the console application’s functionality. Furthermore, visuals will be provided to illustrate the solution to the problem, as well as being used to provide additional context in some sections.

Following this overview of the paper’s contents, the detailed structure includes the context of this paper, covering the background, initial research on provided information, and an evaluation of possible technologies to optimally accomplish the task. It then moves to the process of creating the project application, outlining thought processes, the creation of application functionality, and the testing of the created functionality to ensure correctness. Finally, this paper concludes with an evaluation of the project’s success in addressing the initial problem, as well as providing future recommendations and improvements for any subsequent projects.

# Background

The detailed problem revolves around the optimal assignment of frequencies (presuming radio frequencies) to several Cell Towers ensuring that those farthest apart can share the same frequency to minimise interference (Chou *et al.*, 1982:1321). This is important because when Cell Towers are close proximity and share the same frequency, they can interfere with each other’s broadcasts, potentially leading to communication failures or distortion (Mawrey, 1998:2; Porko, 2011:12-15, 18; An *et al.*, 2017:2).

To develop a frequency allocation plan that minimises potential interference, necessary information is provided. This information includes a list of potential Cell Towers, each with a unique identifier, easting and northing coordinates, as well as longitude and latitude coordinates (Langley, 1998:46-47). Along with the Cell Tower information, an initial set of available frequencies is provided, which ranges from one hundred and ten (N = 110) to one hundred and fifteen (N = 115). All this information can be used to develop an algorithm that determines the most optimal frequency, of a given range, to allocate to a Cell Tower while minimising potential interference between Cell Towers.

## Initial Decision Making

Given the information about the Cell Towers, certain design decisions needed to be made in order to effectively create a frequency assignment algorithm, such decisions include the programming language used to create the project and the manner in which the frequencies will be assigned.

Though some initial research and troubleshooting, it was decided that Cell Tower frequencies will be assigned to a target Cell Tower while considering the frequencies used by the tower’s closest neighbours. In other words, the frequency given to a target Cell Tower will take into consideration the closest N neighbours to the Cell Tower (N is the number of neighbours to consider) and determine the frequencies that they are using. This list of used frequencies will be compared to the list of all frequencies to determine the frequencies that have yet to be assigned. From the unassigned frequencies, the smallest frequency, or the frequency closest to the lower bound of all available frequencies, will be assigned to the target Cell Tower. This process will then be repeated to assign frequencies to all Cell Towers. In the case where a Cell Tower cannot be assigned a frequency, it will be reported to the operator for further action. This manner of assigning frequencies to Cell Towers is the core idea of this project. The why correctness was ensured will be discussed in a subsequent section.

To implement this idea, additional research was necessary, including how distance between towers will be calculated and which coordinate system to use?

### Coordinate System

The first consideration was the coordinate system. As previously mentioned, initial information provided easting and northing coordinates (Universal Transverse Mercator (UTM) system), as well as longitude and latitude coordinates (geographic coordinate system) (Langley, 1998:46-47). Determining the coordinate system to use is a core part of the proposed solution.

The main difference between the UTM (Universal Transverse Mercato) coordinate system and the geographical coordinate system is that the UTM coordinate system simplifies the spatial calculations and mapping by using a grid-based projection to transform earth’s curved surface into a two-dimensional plane (Langley, 1998:47). This transformation abstracts complexities, such as angular measurements, making distance and area calculations simpler compared to the geographical coordinates system (Langley, 1998:46-47).

With this information, it would be logical to use the UTM coordinate system, as the task has towers stationed in close proximity to each other; however, to improve the project’s applicability to more complex scenarios, the geographical coordinate system will be used (Langley, 1998:46-47). In simpler words, the longitude and latitude coordinates of each Cell Tower will be used in distance calculations, allowing these calculations to account for the earth’s curvature and other factors (Langley, 1998:46-47).

### Distance Calculations

With the coordinate system chosen a manner of calculating the distance between towers needs to be established. The distance metrics considered where, the Euclidean distance, the Manhattan distance, and the Great-Circle distance using the Haversine Formula (Young & Householder, 1938:19; Krislock & Wolkowicz, 2012:1-2; Malkauthekar, 2013:1-2; Dauni *et al.*, 2019:3).

Of these three metrics, it was determined that the Great-Circle distance (Haversine Formula) was the most suitable, as it considers earth’s curvature and provides acceptable accuracy for long distance calculations (Dauni *et al.*, 2019:3). This makes it suitable for this project, as it improves the applications applicability in more complex scenarios.

### Programming Language

With the proposed solution in mind, it was necessary to choose the appropriate programming language for implementing the solution. For this project, C# was chosen primarily due to the author’s familiarity with the language, enabling the use of existing expertise (Microsoft, 2024).

#### Libraries

In addition to the main programming language, research was conducted on potential C# libraries to use for simplifying aspects of the project, mainly graph creation, to ensure that the project meets the proposed deadline. During this research, QuickGraph and Msagl were considered when researching options to effectively create the Cell Tower network graph visualisation (KeRNeLith, 2022; Microsoft & Lamb, 2023).

Both libraries offer a range of capabilities; however, initial coding using the QuickGraph library, revealed limitations, particularly in graph visualisation and creation, which led to the use of the Microsoft Msagl library for all graph creation and visualisation needs (KeRNeLith, 2022; Microsoft & Lamb, 2023). All of the mentioned aspects will be discussed in subsequent sections.

On a smaller note, the Ookii library was used when creating a dynamic file selector as the built in Windows library had consistency problems on the host machine (Ookii, 2021).

# Application Development

This section will detail the creation of the console application to address the problem statement described in the Background section. This section covers the entire development process from the creation of the application’s input functionality to the assignment of Cell Tower frequencies.

## Input Functionality

Before the application can accomplish its task, input is required. To gather the information on Cell Towers and the frequency range to use when performing allocation, dynamic functionality was implemented.

However, before the data can be loaded, it needs to be in a standardised format. To accomplish this, the Cell Tower information needs to be stored in a text file in the format of Cell Tower ID, Easting, Northing, Longitude, and Latitude, each separated with a semicolon. This approach was chosen due to the low setup time required. After this preprocessing step, users can select the text file through the application to load the data. The format is illustrated in Figure 1. Input Data.

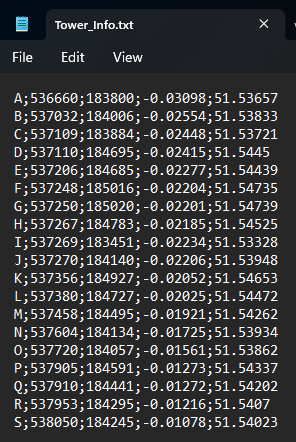


Figure . Input Data

This data was loaded into a list of type Cell Tower. The Cell Tower class is a custom class created to contain Cell Tower’s information and simplify the future creation of graph nodes, which will be discussed in a later section. The Cell Tower class is illustrated in Figure 2. Cell Tower Class.

A screen shot of a computer program

Description automatically generated

Figure . Cell Tower Class

Using a list along with objects enables extreme flexibility, as the list can continuously grow along with the input, and encapsulating Cell Tower information in an object allows standardisation, enabling simplistic data management and scalability (Liang, 2015:322, 366).

Along with the input for Cell Tower information is the selection of the possible frequency ranges to use. This in accomplished using the terminal and asking users for the lower bound (the first frequency that can be used) and the upper bound (the last frequency that can be used). This input method is illustrated in Figure 3. Frequency Range Input.

A screenshot of a computer program

Description automatically generated

Figure . Frequency Range Input

This method is simplistic to set up correctly, which is the main reason for its selection.

Along with the input for the Cell Tower information and frequency range, the user can also adjust the number of closest towers to a target tower to consider when assigning frequencies. This is illustrated in Figure 4. N Closest Towers.

A screenshot of a computer program

Description automatically generated

Figure . N Closest Towers

This input is important for several reasons; however, the main reason is to enable hyperparameter optimisation, allowing the user to select the most optimal number of closest towers to a target tower for their given situation (Bischl *et al.*, 2023:1).

## Graph Creation

With all the necessary information gathered it is now possible to implement the proposed solution to the Cell Tower Frequency assignment problem.

The first step conducted was the creation of a graph to store all the Cell Tower information in the form of vertices and the distance of each tower to the others as a weighted edge connecting the towers. It may not be necessary to create a graph at this stage in the project, as it is possible to solve the problem without creating a graph and using lists instead; however, as the original task requires an illustration of the solution in a graph format, it was necessary to create a graph at this stage not only for the author to familiarise themselves with the graph creation library but also to determine the library's suitability for the task, which did prevent future problems as described in the Libraries section. The code for the graph creation is illustrated in Figure 5. Graph Creation Code.

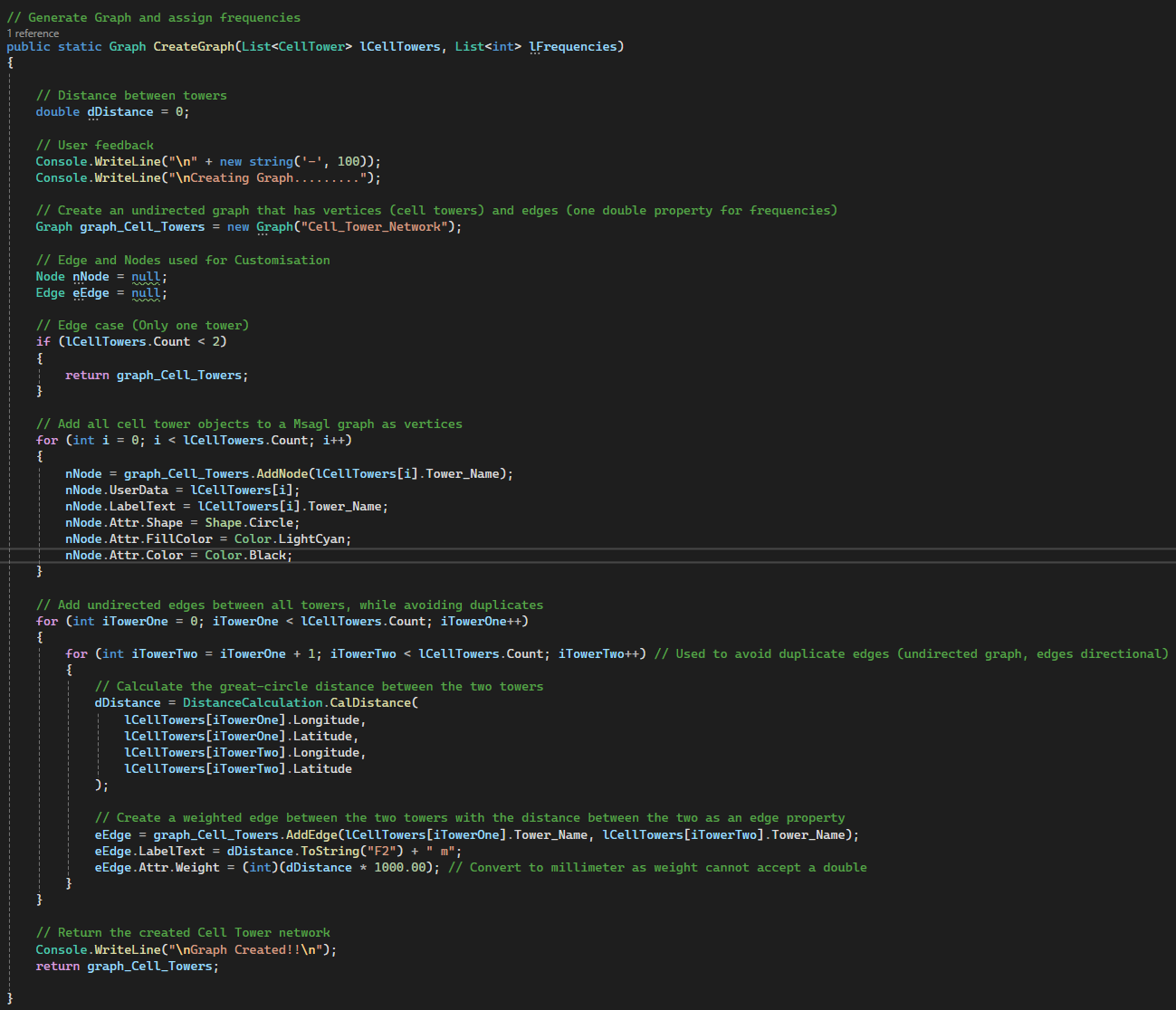


Figure . Graph Creation Code

As indicated in Figure 5. Graph Creation Code: A custom distance calculation method was used when calculating the distance between two Cell Towers and subsequently stored as a weight on a graph edge. It is important to note that the graph library (Msagl) does not support any decimals as weights; therefore, the weight was stored as an integer but converted from meters to millimetres to retrain an acceptable precision level for the task (Microsoft & Lamb, 2023).

### Distance Calculation

As mentioned previously, the haversine formula was used when calculating the distance between towers (Dauni *et al.*, 2019:3). The function in which this formula is applied accepts the longitude and latitude of two towers as input. These inputs are subsequently converted from degrees to radians, which is another angular measurement used in trigonometry and for defining trigonometric functions. The converted coordinates are then used to in the haversine formula as, illustrated in Figure 6. Haversine Formula (Dauni et al., 2019:3), to calculate the distance between two towers while accounting for the Earth’s curvature (Langley, 1998:46-47; Dauni *et al.*, 2019:3).

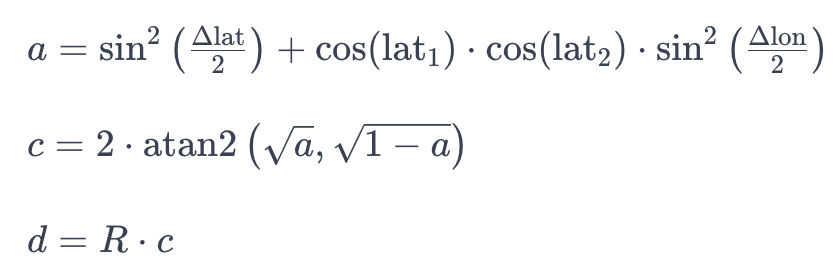


Figure . Haversine Formula (Dauni et al., 2019:3)

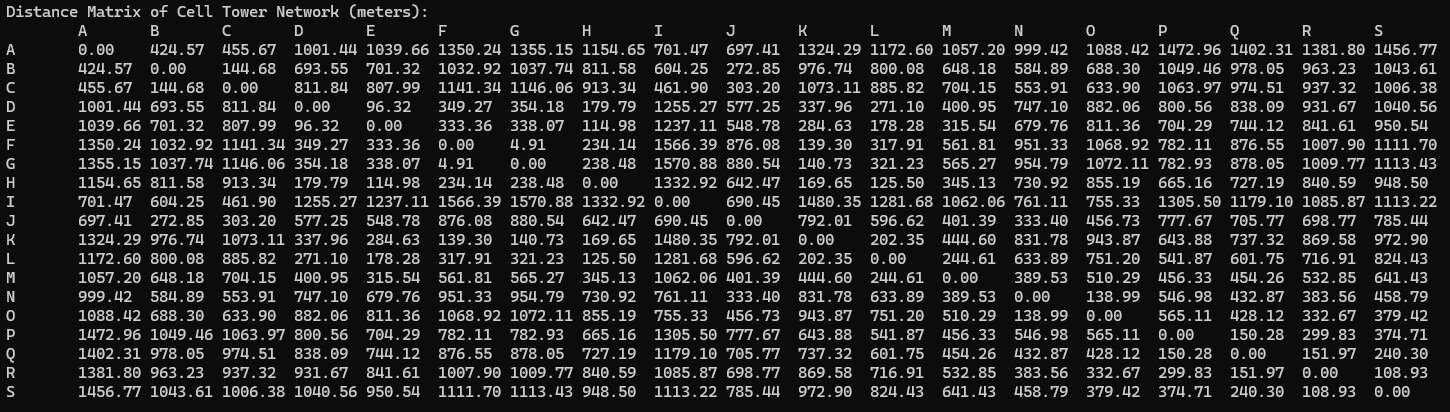
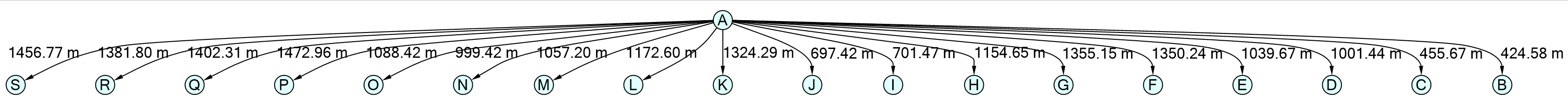
Using this formula all the distances, in meters, between towers where calculated, as illustrated in Figure 7. Distance Matrix.

Figure . Distance Matrix

To ensure the correctness of the calculations, a random selection of towers was selected, and their distances to other towers were calculated by hand and using an online calculator, which confirmed their correctness (MTL, 2024).

### Graph Illustration

With all the Cell Towers and edges created, Msagl’s graph visualisation capabilities were utilised to create a full graph visualisation. The following image illustrates one of the Cell Towers with weighted edges, containing the distance from the target tower to the other towers, connected to all other towers (Microsoft & Lamb, 2023).



Unfortunately, due to the many edges that are created the full illustration of the graph is complex, as seen in Figure 7. Full Graph Network.

Figure . Full Graph Network

This graph illustration can be improved in future projects by leveraging geometry nodes, a feature of the Msagl library that enables the plotting of nodes based on their coordinates. However, due to technical difficulties and time constraints, this feature could not be implemented in the current project (Microsoft & Lamb, 2023).

## Frequency Assignment

With all the necessary information calculated and gathered, the proposed method of assigning frequencies to Cell Towers, as mentioned in the Initial Decision Making Section, was implemented as an algorithm. The code is illustrated in Figure 9. Frequency Calculation Code.

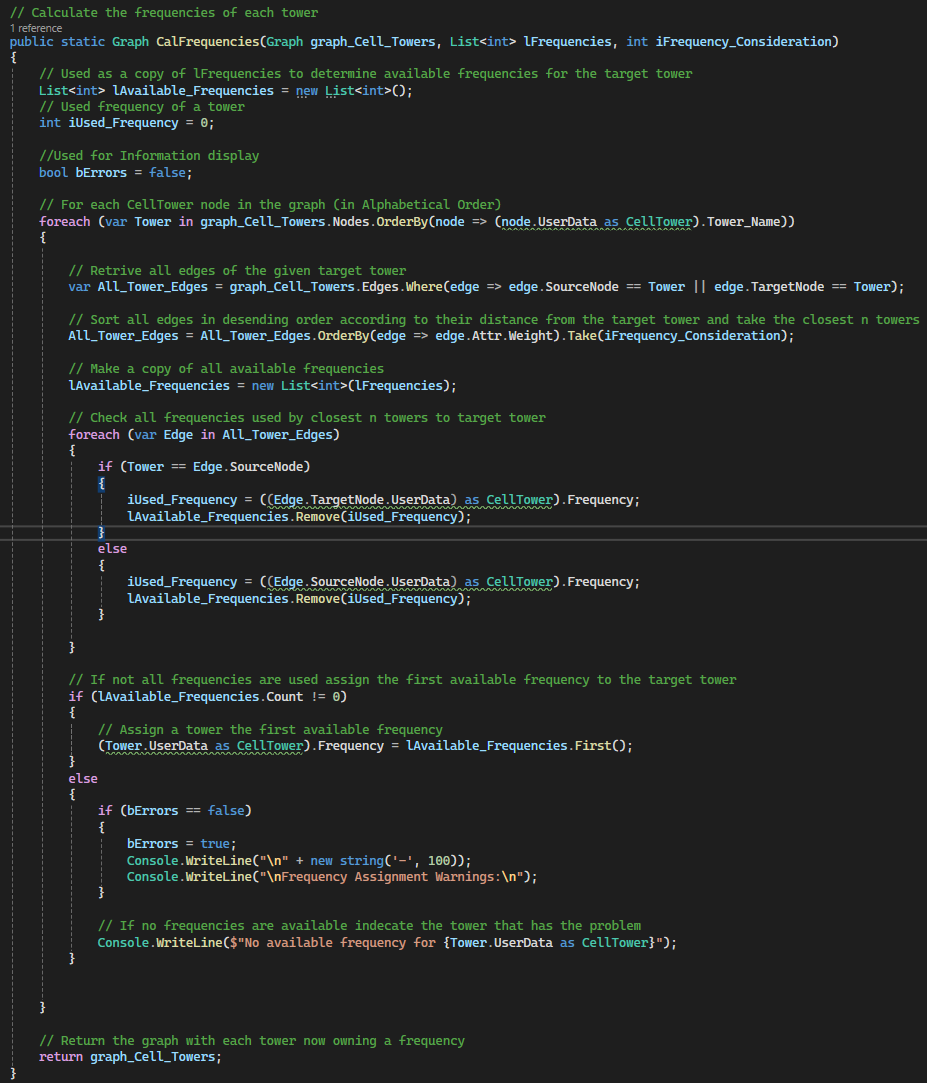


Figure . Frequency Calculation Code

In this algorithm, it accepts the previously created graph as input along with a list of valid frequencies and a number indicating how many closest towers to a target tower to consider before assigning a frequency.

For each node (Cell Tower) in the graph, all its connected edges are added to a list, effectively creating a list containing the distances of each tower to the target tower. This list is then sorted, and the closest N towers are selected based on the previously specified number.

Using this list of the closest towers, the frequencies already used by these towers are compared to the list of all valid frequencies to identify the frequencies that have yet to be assigned. From the unassigned frequencies, the smallest frequency, or the frequency closest to the lower bound of all available frequencies, will be assigned to the target Cell Tower.

This process will then be repeated to assign frequencies to all Cell Towers. In cases where a Cell Tower cannot be assigned a frequency, it will be reported to the operator for further action. The completed frequency assignment is illustrated in Figure 10. Frequency Assignment on the list of initial Cell Towers provided, while considering the closest five (N = 5) towers for each target tower.

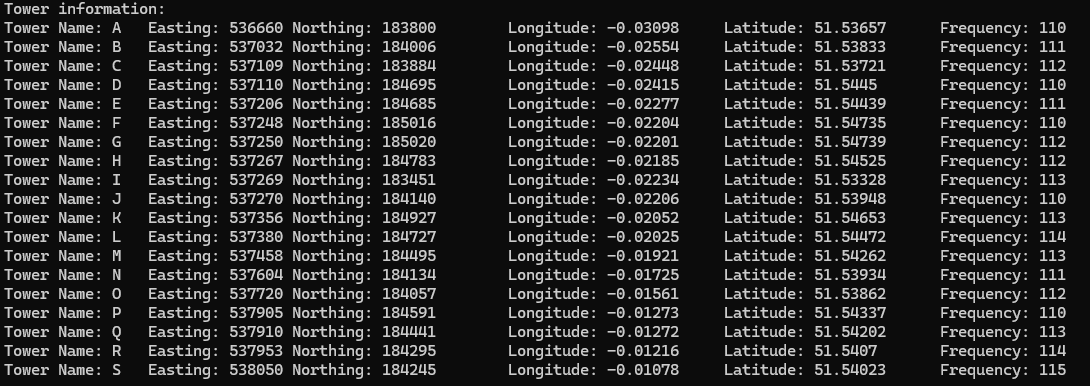


Figure . Frequency Assignment

This algorithm was run considering the closest four (N = 4) and five (N = 5) tower and resulted in the frequency assignments, illustrated by Figure 11. Closest 4 and 5 Towers.

A screenshot of a graph

Description automatically generated

Figure . Closest 4 and 5 Towers

### Frequency Assignment Correctness

The most difficult part of the problem space was the evaluation of the frequency assignment’s correctness. This was challenging due to the absence of labelled data to compare the results to (Nasteski, 2017:4; Shyam & Chakraborty, 2021:3).

To address this challenge, the correctness of the algorithm was evaluated manually. Distances were calculated, and the algorithm's results were reviewed by hand to ensure consistency. This approach aimed to verify that frequencies were assigned as optimally as possible.

Despite the difficulty of trying to evaluate the algorithm’s correctness, the calculations of the distances and the algorithm were done by hand to ensure that the results are consistent and that the frequencies were assigned as optimally as possible. However, it is important to note that the optimality of frequency assignments remains subjective. The lack of predefined distance metrics to indicate interference ranges, along with the absence of results to compare the frequency assignments to, introduced uncertainty. Additionally, the manual evaluation process was susceptible to human error.

Nonetheless, the frequency assignment for each cell tower was performed as accurately as possible.

A screen shot of a computer

Description automatically generated The following image is a excel sheet representing the manual calculations conducted.

Figure . Correctness

Additionally, Figure 13. Manual Frequency Verification illustrates the application of the frequency assignment considering both the closest four (N = 4) and five (N = 5) towers.

A screenshot of a chart

Description automatically generated

Figure . Manual Frequency Verification

Using these manually calculated frequency assignments the consistency of the algorithm was further confirmed.

# Conclusion

# References

An, T., Chen, X., Mohan, P. & Lao, B.-Q. 2017. Radio frequency interference mitigation. *arXiv preprint arXiv:1711.01978*,

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., ... Boulesteix, A.L. 2023. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(2):e1484.

Chou, C.K., Guy, A.W. & Galambos, R. 1982. Auditory perception of radio‐frequency electromagnetic fields. *The Journal of the Acoustical Society of America*, 71(6):1321-1334.

Dauni, P., Firdaus, M., Asfariani, R., Saputra, M., Hidayat, A. & Zulfikar, W. 2019. Implementation of Haversine formula for school location tracking*.* In. Journal of Physics: Conference Series. IOP Publishing. p 077028.

KeRNeLith. 2022. *QuickGraph*. <https://www.nuget.org/packages/QuikGraph> Date of access: 13 January.

Krislock, N. & Wolkowicz, H. 2012. *Euclidean distance matrices and applications*. Springer.

Langley, R.B. 1998. The UTM grid system. *GPS world*, 9(2):46-50.

Liang, Y.D. 2015. *Introduction to Java Programming, Comprehensive Version*. Tenth Edition. Upper Saddle River, New Jersey, USA: Pearson Education, Inc.

Malkauthekar, M. 2013. Analysis of euclidean distance and manhattan distance measure in face recognition*.* In. Third International Conference on Computational Intelligence and Information Technology (CIIT 2013). IET. pp. 503-507.

Mawrey, R.S. 1998. Radio Frequency interference and Antenna sites. *Unisite, sitio web:* [*http://www*](http://www)*. unisite. com*,

Microsoft. 2024. *C# language documentation*. <https://learn.microsoft.com/en-us/dotnet/csharp/> Date of access: 10 October.

Microsoft & Lamb, A. 2023. *Microsoft.Msagl*. <https://www.nuget.org/packages/Microsoft.Msagl> Date of access: 13 January.

MTL, M.T.S. 2024. *Calculate distance, bearing and more between Latitude/Longitude points*. <https://www.movable-type.co.uk/scripts/latlong.html> Date of access: 13 January.

Nasteski, V. 2017. An overview of the supervised machine learning methods. *Horizons. b*, 4(51-62):56.

Ookii. 2021. *Ookii.Dialogs.WinForms*. <https://www.nuget.org/packages/Ookii.Dialogs.WinForms> Date of access: 13 January.

Porko, J.-P.G. 2011. *Radio frequency interference in radio astronomy.* Aalto University.

Shyam, R. & Chakraborty, R. 2021. Machine learning and its dominant paradigms. *Journal of Advancements in Robotics*, 8(2):1-10p.

Technologies, W. 2025. *Developing solutions that address real-world telecoms complexities*. <https://www.wimtechnologies.com/> Date of access: 13 January.

Young, G. & Householder, A.S. 1938. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3(1):19-22.