Adam Whitter-Jones

0924607

Interim Report for project 87: Network design

Module 343

Supervisor: Steve Hurley

Moderator: Paul L. Rosin

Table of Contents

Adam Whitter-Jones
0924607

Abstract

This project aims to help with planning and testing the positioning of telephone masts, referred to as BSTs within the telecommunications industry. The application that has been designed utilises raw data obtained from Vodafone that is used to help depict the differing strengths of signals (in decibels) given their (x,y) coordinates. After plotting this given information, the end result of the application is to permit users to interact with the software to be able to compare the results of positioning these BSTs in differing locations. Further, incorporating an algorithm to recommend the most effective place to establish these BSTs is the main objective. This will assist the places a BST is located so that it has minimum cost to the network operator and maximum coverage to the users of the mobile network.

Introduction

This project has been undertaken in order to aid the planning process that is undertaken to decide the most appropriate location for Base Station Transceivers (BSTs) when deploying a mobile phone network. BSTs are one of the key components required in a mobile phone network and permit end users to connect via their mobile telephone to other resources or people around the world. The issue that is encountered when planning these networks is both an issue of cost and also of signal propagation. Simply, the key question asked is how can we place the minimum number of BSTs and serve the maximum number of customers as possible?

Through creating this application, the beneficiaries are both network providers and the customers which use their networks. Enabling a network service provider to plan and deploy a network that is as efficient as possible reduces time and cost. This would hopefully reduce corporate expenditure on some of the most costly operations which would hopefully filter down to the users of the network, reducing their charges incurred for using the services provided. In addition to this reduction in cost, effective planning of a network is essential to serve as much of the population as possible. As we can see from several recent news articles[1,2,3], the digital age is creating fragmentation between urban and rural areas. Communication and the Internet are key factors in enabling businesses to grow and develop.

To assess and hopefully assist this issue, raw data relating to these BSTs has been obtained. We have several different plain text files from network operator Vodafone. These are as follows:

1. Reception Test Points (RTP) listed by their (x,y) coordinates.
2. Propagation Losses Matrix (PLM) which contains propagation losses (in dB) from each potential site to every RTP.

[1] http://www.bbc.co.uk/news/technology-15679101
[2] http://www.bbc.co.uk/news/technology-15854582
[3] http://www.bbc.co.uk/news/technology-15717913

3. Service Test Points (STP) to which a known service is required (dBm) listed by their (x,y) coordinates.
4. Traffic Test Points (TTP) in which traffic demand is known (in Erlangs) and is listed by their (x,y) coordinates.
5. Mobile and Base Station Antenna Information (NET) in which the names and locations of sites are specified.

The data set provided relates to a mobile network, that isn't necessarily beneficial to the planning of Internet enabled BSTs. However, the foundations of this application could hopefully be adapted in the future to assist with the planning of mobile Internet.

Once the application is completed it should be able to read data files similar to above and be able to plot them to a grid that is both visually aesthetic and easy to understand. In addition to this, it is a vital requirement that user interaction is permissible so that assessing various changes in the power or the location of specific sites renders different results to assist in planning. For this to happen, I hope for the data sets to be populated in amendable text fields permitting the user to adjust the (x,y) coordinates and the power of each mast so that they can see the difference in the amount of users served and any changes in signal strength. To do this, an algorithm needs to be created to compute the highest signal strength of a site given the (x,y) coordinates and varying power. At this stage, the use of a simulated annealing (SA) algorithm has been planned for optimum site selection. Through using this algorithm, it is hoped that the most important part of the network planning stage can be computed with little to no interaction from the end user. The application should in its entirety require the user to input specified powers and current data sets and then enable the algorithm to calculate the most viable places to locate sites, striking equilibrium between network cost and users served.

Background

Recently we have seen a global increase in the technology and subscribers of wireless technology. Mobile network providers have realised the necessity in efficient design and planning of these networks and the importance of a stable and efficient network. Hence, services offered to assist in this planning and development of said networks has increased. As the use of wireless technology grows year on year, the necessity for efficiency is something network providers strive for.

The problem that we face is that if we place a BST in a specific area, although it may service many people within its locality, it may not serve the optimum number of people if it was placed in a slightly different place. As it is obviously not feasible to move the physical BSTs around and test the results, developing an algorithm to calculate and predict these results will help us to visualise the required information during the planning stage. This permits for a reduction in costs for mobile network providers whilst providing minimum disruption for service users.

The data set that I am using relates to 250 real sites, which is fraction of the true amount of telephone masts that exist in the United Kingdom. To check the amount of BSTs in your area, you can visit the Ombudsman for Communications (OFCOM) website[4]. The site information I am working with is two dimensional and is plotted on a simple grid within Visual Basic .NET that spans to the maximum size of the dataset. The maximum size of the grid I am working with is determined by reading the RTP file and calculating the highest and lowest values from the data. Within the grid, BSTs, STPs and TTPs are plotted relative to the aspect ratio. Taking the real life (x,y) coordinates of the BST, STP and TTP data, I have then manipulated these through the calculations

```
x = ((x - BaseStation.minimumX) / (BaseStation.maximumX - BaseStation.minimumX))
y = ((y - BaseStation.minimumY) / (BaseStation.maximumY - BaseStation.minimumY))
```

Figure I

Where x and y relate to the location of the sites in pixels, x and y are the (x,y) real world coordinates of the sites read from the RTP file and minimum/maximum x/y are the real world (x,y) coordinates calculated from the RTP file.

It is important to realise at this stage that plotting the data in a two dimensional plane may be slightly unrealistic in that the lack of visualisation of terrain and physical objects in the x plane does not help us to determine any factors that may be reducing the strength of signal and coverage area when moving the BST to a new (x,y) coordinate. As we are currently unable to see objects on the map we are using, the decision as to where the best place to move a BST to will only be determined through amending the coordinates; this may be extremely inefficient as the interaction on behalf of the user could be reduced if they we able to see something on the map they knew would reduce the capabilities of the BST.

Approach

I have chosen to create this application using VB .NET as I have sufficient knowledge of this language as it is an advancement of VB6, a language that I worked with during secondary school and college. Although using this language limits the application to being run within a Windows environment, Visual Studio allows for easy code writing and enables quick and professional formatting and indentation. Further, the ability do debug in real time along with detailed explanations as to what has gone wrong was a key driver in choosing this language over others that may not necessarily provide an environment through which debugging is as effective.

One of the key aspects to ensure that the project goes in the right direction was to ensure from the start that there was some clear understanding and direction relating to the project. One of the simplest ways for me to visualise the project was through creating an activity diagram (Appendix VII) outlining some of the basic aspects and key functionality I wanted to incorporate. Within this activity diagram I have shown at the end the option to export the results. The purpose of this is so that once the optimum site locations have been determined they can be exported for further analysis in more

[4] http://www.sitefinder.ofcom.org.uk/

readable format. This feature has not yet been included within the application and hence the format in which the data is exported is yet to be decided.

At the beginning of the project, I set about designing some key aspects of the application that I felt were important. The key parts of the application are the design of the interface and the method through which the data would be visualised. My initial design of the graphical user interface (GUI) shows text fields with related buttons that enable the user to amend aspects of the data for their investigations. At the current stage in the project, these fields are not incorporated as the ability through which the user is able to manipulate the data has not been completed. Further, the plotting of the data within this interface is a key aspect of the application as without being able to visualise the data, the application serves no purpose. As a result, I set about designing some simple pseudo code to assist me when the implementation of the code was required. This pseudo code can be seen in Appendix III. The pseudo code details the process of drawing and visualising the data; its implementation follows in Appendix IV. Further, sample code I have written to extract the data from the .NET file can be seen in Appendix V.

So far in the project, I have only achieved one of the main objectives specified for this report that was set out in the initial plan; extracting the data for the application. The objectives

1. … design algorithm (include NPM) for applying the data.
2. Design the environment and how I expect to use the data.
3. Design and create a testing schedule.

One of the aims for the final report, *"Create the GUI and apply the testing schedule to obtain data and permit further analysis of the data"* has been partially completed in that the GUI has been successfully created but needs further enhancements to permit interaction from the user. Hence, the ability to visualise the data on a grid is in place, as shown in Appendix II.

When reviewing my initial project plan, it seems worthy to note that some of the dates and objectives specified for them were in the wrong order. For example, it was said that I would design the algorithm before the GUI; however in retrospect, without visualising the data first, it would be difficult if not impossible to create and assess the success of the algorithm. Further, some of the objectives that were set out in the initial plan, as specified above, have not been completed. At this stage I was hoping that it would have been possible for the user to interact with the data and have the basis for the algorithm completed; unfortunately I am slightly behind schedule.

One of the final key parts of the application that is yet to be completed is the reading of the PLM file. This file details the path loss at specific coordinates for each site. This data will be used later on in the application to support the SA algorithm to help determine the best location to place a site. Through the assistance of my supervisor we have deduced a logical way as to how to read the file as its structure differs somewhat to the other files used.

As specified previously, data is fed in to the system through a series of data files stored relative to the application. These files are read in to the application using VB .NET function StreamReader. One of the files, as mentioned before, is not read line by line as the others are. The PLM file in this instance is too large to be loaded in the memory all at once. Instead, I have chosen to read the file in byte by byte. Using the information provided in Appendix IV we can see that the data is ordered in specific bytes. Having this information has greatly helped in writing an effective method for reading such a large file.

Once these files have been read, they are stored in the application using a series of different objects and object instances. Further, encapsulation has enabled me to add and data to these objects when reading in the data from the files and subsequently has made accessing the data for plotting and analysis easier and more efficient.

Conclusions

To date, I feel I have made sufficient progress on the project itself. Whilst facing some issues relating to the programming aspect of the application I have been able to research required information effectively on the Internet whilst getting sufficient support from peers. Further, the support from my supervisor has been excellent. Unfortunately I am not as far along in the project as either my supervisor or I had hoped; this is due to reasons beyond my control.

When reviewing my initial project plan, the timeline that I had initially created before embarking on the programming stage of the project doesn't accurately reflect the sequence of in which the programming and development happened. Hence, some of the future targets have already been completed and tasks that should have already been done are yet to be fulfilled. To ensure that I have a more structured approach going forward in development, I have updated and amended the initial project plan timeline to more accurately reflect the stages in which the application will be created. The timeline has only been updated from week 11 onwards and this can be found in Appendix VI and is subject to further change depending on how the development of the software progresses. It is hard to assume at an early stage which aspects need to be created first to enable a more fluid development process.

From this revised timeline of targets, the main objects for the second half of the project are to implement the algorithm and network performance measure (NPM) and complete the design and interaction with the interface. The NPM is to be used within the SA algorithm and is determined through the following calculation:

NPM = % coverage (max) + %traffic/service (max) + cost (# of sites) [min]

Figure II

When we are calculating the NPM for each STP on the graph, whichever returns the highest NPM is the server we will chose. This measure will help to determine if increasing the power to two sites is more efficient than using reduced power at three sites and so forth, resulting in an optimised result for planning site locations.

The main aspect of interaction that needs to be completed is the ability for users to interact either with individual sites on the grid, which may be hard due to the amount of sites. If this isn't feasible to incorporate, the second option I have chosen is shown in my initial GUI design (Appendix I). This is that fields and buttons relating to specific sites are displayed to the right of the grid.

Once I am completed with the GUI and my supervisor agrees that it is satisfactory, a series of bug tests and optimisation will be carried out to ensure that the application is as effective as possible and meets the targets specified when the project was undertaken. At this stage, the function used to draw the sites and grid to screen isn't as efficient as I had hoped; when loading the application there is a delay whilst drawing occurs, giving a less than satisfactory user experience. To assess if the optimisations have been met, regular meetings with my supervisor will need to be undertaken to demonstrate the application and gain opinions and advice.

Glossary

Activity Diagram - an analysis model that shows a dynamic view of a system by depicting the flow from one activity to another; similar to a flowchart. (Process Impact)

Algorithm - a process or set of rules to be followed in calculations or other problem-solving operations. (Google)

Base Station Transceiver - equipment to assist wireless communication between a mobile network and user equipment, such as a mobile telephone.

dBm - power ratio in decibels (dB) of the measured power referenced to one milliwatt (mW).

Decibel - the unit used to measure the intensity of sound or the power level of electrical signals through comparing it with a given level on a logarithmic scale. (Google)

Erlangs - a unit of traffic intensity in a telephone system. (WordNet Search)

Propagation - the travel of an electrical signal through a medium such as air or free space. (Aeronomy and RadioPropagation Laboratory)

Regular Expressions - a method to describe how to match a text string to a pattern. Some regular expressions can look rather complex (and some are) but this gives them great abilities. (Clemson University)

Simulated Annealing - an optimization algorithm that chooses random changes that improve the desired trait. (Nicholas, Derek, Jonathan, David, & Nipam)

StreamReader – A class within VB.NET that implements a text reader that reads characters from a byte stream in a particular encoding. (Microsoft)

Visual Basic 6 – Visual Basic 6 is an object-oriented computer programming language that creates applications designed to run within an Windows environment.

Visual Basic .NET - Visual Basic .NET (VB.NET) is an object-oriented computer programming language that can be viewed as an evolution of Microsoft's Visual Basic (VB) which is generally implemented on the Microsoft .NET Framework. (Wikipedia)

Visual Studio - an integrated development environment (IDE) from Microsoft. It is used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight. (Wikipedia)

Vodafone – a mobile network provider. This report specifically relates to Vodafone UK.

Table of Abbreviations

dB – decibels.

GUI – graphical user interface.

BST – base station transceiver.

NET – mobile and base station antenna information.

NPM – network performance measure.

PLM – propagation loss matrix.

RTP – reception test point.

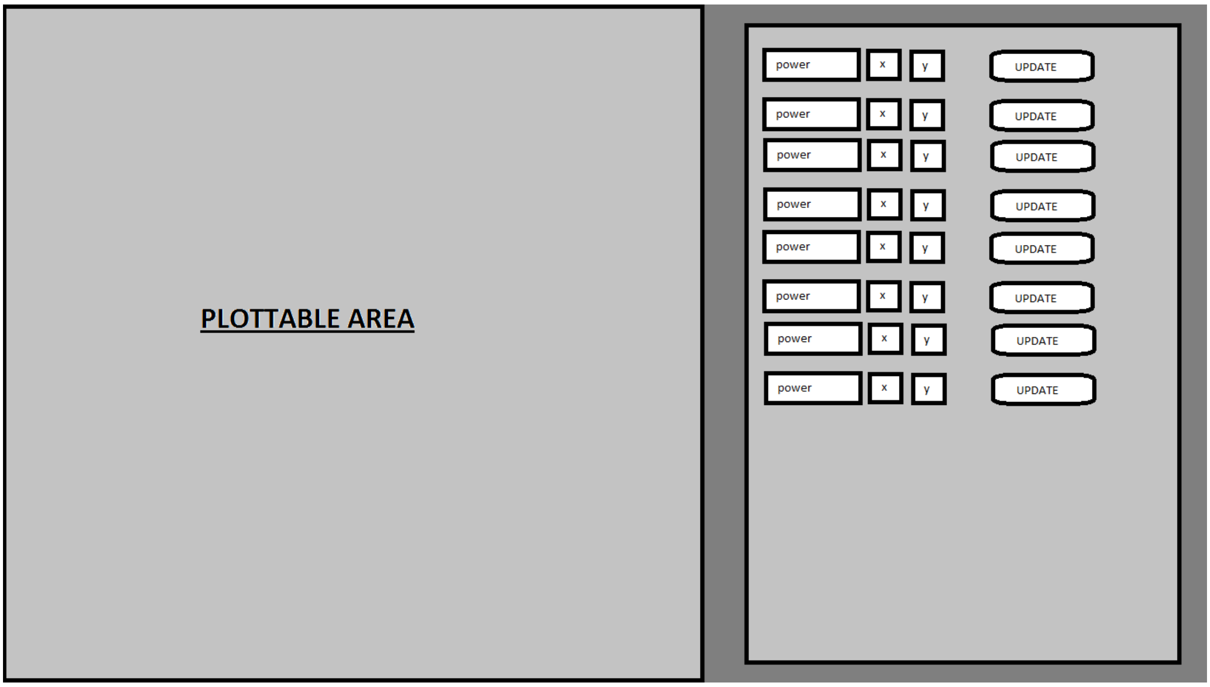
STP – service test point.

TTP – traffic test point.

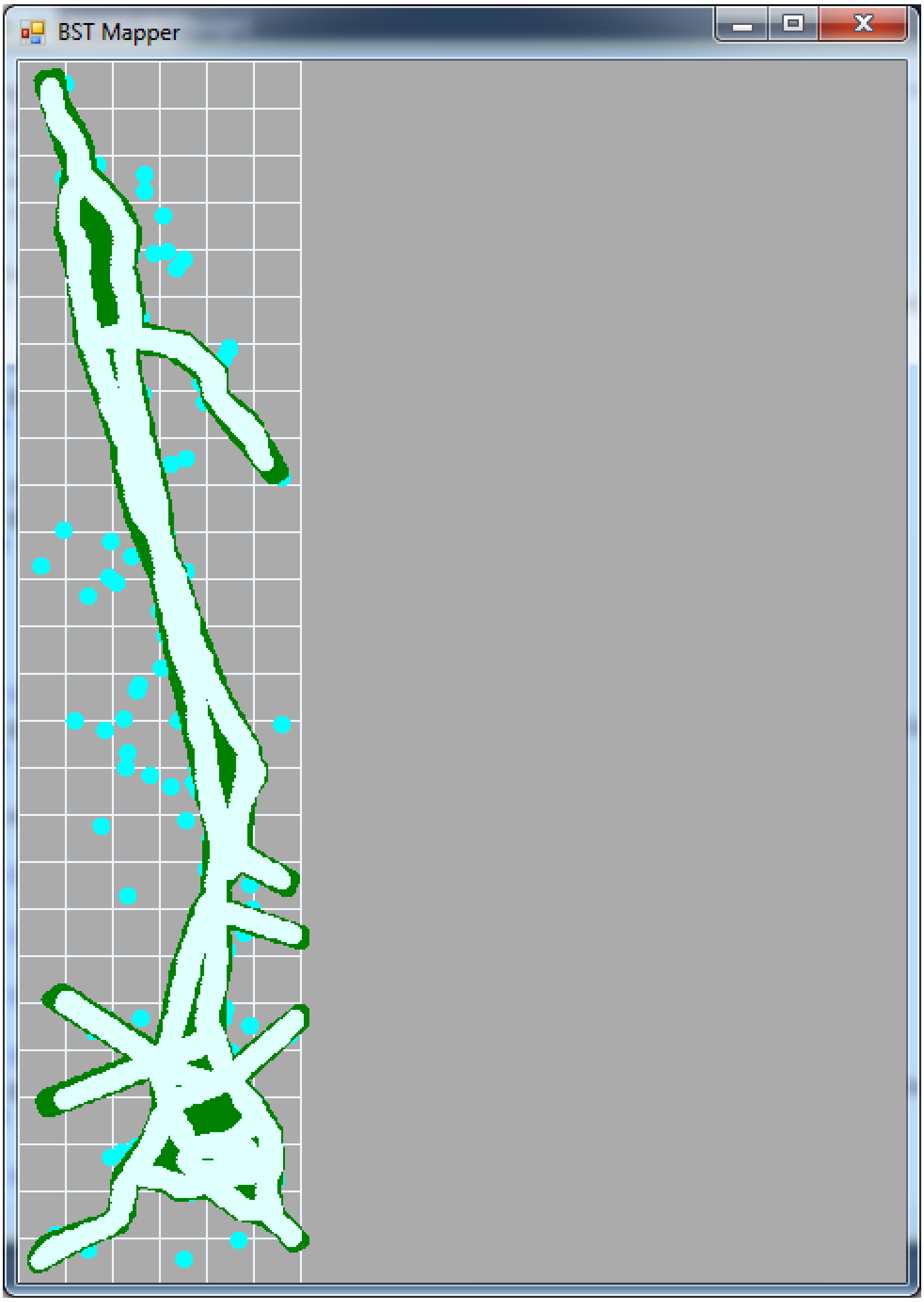
VB – Visual Basic.

Appendices

Appendix I



Appendix II



Appendix III

Drawing Function

Declare graphics object

Declare pen object

Declare xAxis and yAxis as integers

Declare N as our chosen graph size (in pixels)

Set our pixel offset mode to high speed (to enable faster rendering)

Set spaceCounter for X to 0

Increase spaceCounter by 25 in each iteration stopping at our maxmimum (n)

    Draw lines after each space

Next

Set spaceCounter for y to 0

Increase spaceCounter by 25 in each iteration stopping at our maxmimum (n)

    Draw lines after each space

Next

```
For each basestation item in basestation list
       Draw an ellipse representing basestations
Next

Declare myColour as a brush of any colour

For each servicethreshold item in the servicethreshold list
       If the threshold < -90 then
              Change brush to red
       Else if the threshold >= -90 then
              Change the brush to green
       End if

       Draw an ellipse representing servicethreshold
Next

For each capacity item in capacity list
       Draw an ellipse representing the capacity
Next
```

Appendix IV

```
    Protected Overrides Sub OnPaint(ByVal paintEvent As PaintEventArgs)

        Dim g As Graphics = paintEvent.Graphics
        Dim pen As Pen = New Pen(Color.AliceBlue)
        Dim xAxis, yAxis As Integer
        Dim N As Integer = 150
        Dim ratioY As Integer

        'attempt to optimise painting by adjusting offset mode.
        g.PixelOffsetMode = PixelOffsetMode.HighSpeed

        'calculate y axis relative to x to provide correct aspect ratio
        ratioY = (BaseStation.maximumY - BaseStation.minimumY) / (BaseStation.maximumX
- BaseStation.minimumX) * N

        'draw x axis
        For xAxis = 0 To N Step 25
            g.DrawLine(pen, xAxis, 0, xAxis, ratioY)
        Next

        'draw y axis
        For yAxis = 0 To ratioY Step 25
            g.DrawLine(pen, 0, yAxis, N, yAxis)
        Next

        'plot the bast stations
        For Each base_station As baseStationInfo In BaseStation.bstList
            g.FillEllipse(Brushes.Cyan, CInt(base_station.X * N), CInt(base_station.Y
* ratioY), 10, 10)
        Next

        'change brush to black
        Dim myColor As Brush = Brushes.Black

        'plot service threshold
        For Each service_threshold As serviceThreshold In BaseStation.sthldList

            If service_threshold.threshold < -90 Then
                myColor = Brushes.Red
            ElseIf service_threshold.threshold >= -90 Then
                myColor = Brushes.Green
            End If
            g.FillEllipse(myColor, CInt(service_threshold.X * N),
CInt(service_threshold.Y * ratioY), 10, 10)
        Next

        'plot capacity
        For Each capacity_erlang As capacityErlang In BaseStation.capacList
            g.FillEllipse(Brushes.LightCyan, CInt(capacity_erlang.X * N),
CInt(capacity_erlang.Y * ratioY), 10, 10)
        Next

    End Sub
```

Appendix V

```
    Public Shared Sub readNET()
        Dim name, x1, y1 As String
        Dim x, y As Double

        Try
            ' Create an instance of StreamReader to read from a file.
            Dim sr As StreamReader = New
StreamReader(System.AppDomain.CurrentDomain.BaseDirectory & "\network1_0_adam.NET")
            Dim line As String

            ' Read and display the lines from the file until the end
            ' of the file is reached.
            Do
                line = sr.ReadLine()
                'if the line contains NOM, take the first 8 characters and trim from
char ";"
                If line.Contains("NOM = ") Then
                    name = line.Substring(8)
                    name = name.Trim(";", """")
                End If
                'if line contains x=, store x value in same way
                If line.Contains("X = ") Then
                    x1 = line.Substring(5)
                    x1 = x1.Trim(";", """")
                    x = CDbl(x1)
                End If
                'if line contains y=, store y value in same way
                If line.Contains("Y = ") Then
                    y1 = line.Substring(5)
                    y1 = y1.Trim(";", """")
                    y = CDbl(y1)
                    'once y is calculated, we have all the required information we
need.
                    'calculate x values relative to data size
                    x = ((x - BaseStation.minimumX) / (BaseStation.maximumX -
BaseStation.minimumX))
                    y = ((y - BaseStation.minimumY) / (BaseStation.maximumY -
BaseStation.minimumY))
                    'add data to base station object
                    BaseStation.addBST(name, x, y)
                End If
            Loop Until line Is Nothing
            sr.Close()
        Catch E As Exception
            ' Let the user know what went wrong.
            Console.WriteLine("The file could not be read:")
            Console.WriteLine(E.Message)
        End Try
        BaseStation.printBST()
    End Sub
```

Appendix VI

Work Plan

Week 11 12/12/12

1. Finalise PLM file reading.
2. Complete interim report.

- **16/12/2011: Submit interim report for project 87: "Network design"**

Week 12 19/12/12

1. Work on completing the GUI to permit user interaction with sites relating to power.

Week 13 26/12/12

Week 14 02/01/12

1. Finalise GUI.
2. Research algorithms- specifically Simulated Annealing.

Week 15 09/01/12

Exams

Week 16 16/01/12

Exams

Week 17 23/01/12

Exams

Week 18 30/01/12

1. Discuss algorithm and its implementation with supervisor.
   a. Discuss NPM.

Week 19 06/02/12

1. Implement algorithm.

Week 20 13/02/12

1. Test algorithm and ensure it works effectively.
2. Meet with supervisor to demonstrate and obtain opinion.

Week 21 20/02/12

1. Work on GUI once algorithm completed to see if it needs changing/improving.

2. Create testing schedule.

Week 22 27/02/12

1. Test data and document results.
2. Assess if any optimisations are required.

Week 23 5/03/12

1. Optimise application if required.
2. Arrange to meet with supervisor.
3. Meet with supervisor.

Week 24 12/03/12

1. Begin writing final report.
2. Write report.

Week 25 19/03/12

1. Write report.

Week 26 26/03/12

1. Write report.

Week 27 2/04/12

1. Submit draft copy of report.
2. Arrange to meet with supervisor.
3. Meet with supervisor.

Week 28 9/04/12

1. Amend report
2. Resubmit report.

Week 29 16/04/12

1. Arrange to meet with supervisor.
2. Meet with supervisor.
3. Make report changes.

Week 30 22/04/12

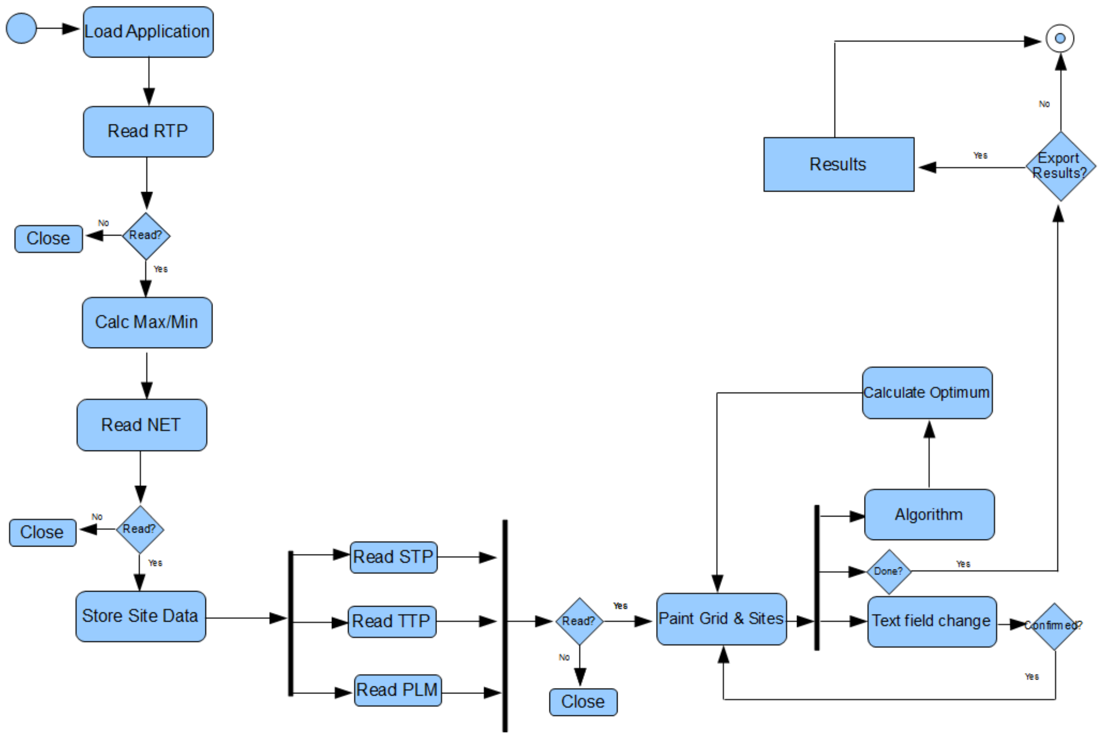
1. Finalise report and additional documents.

Week 31 30/04/12

1. Submit report.

- **4/5/2012: Submit final report for project 87: "Network design"**

Appendix VII



○

Bibliography

Aeronomy and RadioPropagation Laboratory. (n.d.). *Glossary*. Retrieved 12 15, 2011, from Wireless@ICTP | Aeronomy and RadioPropagation Laboratory: http://wireless.ictp.it/school_2003/docs/radio/Glossary.html

Clemson University. (n.d.). *Linux Definitions*. Retrieved 12 15, 2011, from Linux 101: http://learn.clemsonlinux.org/wiki/Linux_definitions

Google. (n.d.). *Google Dictionary*. Retrieved 12 15, 2011, from Google: http://www.google.co.uk

Microsoft. (n.d.). *StreamReader*. Retrieved 12 15, 2011, from Microsoft Developer Network: http://msdn.microsoft.com/en-us/library/system.io.streamreader.aspx

Nicholas, H. B., Derek, E. B., Jonathan, A. E., David, B. G., & Nipam, H. P. (n.d.). *Glossary*. Retrieved 12 15, 2011, from Evolution Textbook: http://evolution-textbook.org/content/free/glossary/glossary.html

Process Impact. (n.d.). *Glossary*. Retrieved 12 15, 2011, from Process Impact: http://www.processimpact.com/UC/Module_3/data/downloads/glossary.html

Wikipedia. (n.d.). *Visual Basic .NET*. Retrieved 12 15, 2011, from Wikiapedia: http://en.wikipedia.org/wiki/Visual_Basic_.NET

Wikipedia. (n.d.). *Visual Studio*. Retrieved 12 15, 2011, from Wikipedia: http://en.wikipedia.org/wiki/Visual_Studio

WordNet Search. (n.d.). Retrieved 12 15, 2011, from WordNet Search: http://wordnetweb.princeton.edu/perl/webwn?s=erlang