National Grid Demand and Cost Representor

For A-level GCE Computer Science

Analysis

# Overview of The Problem

Firstly, I would like to outline the problem I have at hand. It is one of great complexity and cannot be simply explained within a single title. The project I propose to confront is a project that will ideally predict the demand for electricity within a given large-scale area so that an electrical company can arrange to inform the power suppliers that there will be more demand and therefore they will need to supply more. I hope to introduce an element of machine learning to optimize this process to the most efficient that it can be. To place it in the simplest terms what I would like to include is:

* An ideal algorithm to take large sets of inputted data and produce the most appropriate response. E.g.: high demand at 12pm, therefore more electricity must be produced, but not only this the national grid must compromise to meet this demand across their power lines and stations (the transmission of power must be calculated to a precise level at these points).
* All data sets will be able to be stored in a file format that another program can read and interpret to use the data in different implementations.
* Ideally it should be easily adaptable to any scenario of a power network and not just the national grid.
* It will be all included in a small GUI that will make interactions with the user the most simplistic as possible. However, all raw crucial data will be outputted as normal to the console such that a specialist can interpret it and reuse the data.
* Potentially it will include a visualization algorithm to show that the problem has been completed within the limitations.

Initially, I will outline the main factors that my project will need to introduce. This is the basis of what my algorithms will work off:

* An efficient file reading and writing algorithm that will interpret text data in table form into a variable that can be used within a calculation.
* In terms of machine learning, I would like to use this term lightly. This is primarily due to the fact I will try to introduce this via a method of taking large sets of pre-inputted data and interpreting it into a more reliable solution to my problem (the more data I can obtain the more I can refine my model for the ideal solution).
* I will need to introduce a custom algorithm that will interpret these data items and create a sort of average for that particular time/day/event and produce an ideal response.
* The GUI should be simplistic to use for an average user to visualize the distribution of power at a given time.
* Data should be printed to the console in an easy-to-interpret form so that a professional can efficiently analyse data.
* A ‘Bat algorithm’ which is a relatively new and complex optimization algorithm, which should produce quicker results and hopefully take less computation to process. This should introduce a more precise prediction for the ideal transmission of power and demand.
* So far this is what I would call the ‘easy part.’ However, the main piece of difficulty is using this bat algorithm to produce a valid response. The problem is that when the national grid produces a response based upon the data inputted, the response must be to an extremely high degree of accuracy. Any flaws within this process can cost the power supplier millions in losses and the national grid tremendous problems with the distribution of power. To combat this, I hope to produce:
  + A verification algorithm, is simply, a reiteration of the most complex piece of code so that the response roughly aligns.
  + An idea of the ‘concentration’ of power on a line to show how an increase/decrease will affect the line.
  + An index, which will be included on how to interpret the demand into current and voltage values. This will also include a value for a time scale on which to act on this, eg: apply more power to this area to x amount of time.
  + An idea of what areas require more power than others. Eg: areas of heavy industry require more power than a small residential zone. This will then be sent to the power supplier for that area and told the response.

# Stakeholders

This is a particular program that will be aimed towards a more real-world application. As I would hope to refine this solution in order to create a long-term method of reliably predicting the demand for energy at a given time. Therefore, the ideal customer is either a researcher or a power supplying company. This area of industry would be filled with professionals that can interpret my complex sets of data and send a response. Since I would like to drop in some sort of visualization algorithm/program, it could be used to interpret large data sets on this topic and be used to research this area further. This is critical for a supplying organization to have, as wasted input of power or deficit can cause people to be without electricity or be a waste and causing a loss in profits for a company or potentially increased, unnecessary carbon emissions. It could be adapted on a more powerful computer or even a distributed network easily to process outputs quicker.

My stakeholder, in this case, is Jonathan Jennings, who is the executive of a company called Meadean Limited. The demand for this originates from the potential, in the foreseeable future, to conserve electrical usage to a minimum, as gas and oil supplies become more valuable assets. This could even be potentially used in everyday households to conserve energy without stopping everyday life. Eg: It will automatically turn off lights and give more power to areas of buildings in need. I meet with Jonathan every Friday to catch up on my progress within his company, so I will take these opportunities to ask questions and make sure the overall product meets his requirements. Jonathan also has prior knowledge of what this task could take to develop, as he has before worked within the electrical industry and has knowledge of what electrical suppliers may require.

# Decomposing The Problem Via Computational Methods

Large, repeated pieces of data that need to be used within a calculation are repetitive and inefficient for a human to conduct. Not only this but it is prone to errors. This makes it perfect for a computer to solve and produce a more accurate result. The use of computational methods can outline this problem in more detail and with more simplicity. Although I have complex algorithms to introduce into this project, I have very little of them to do and a boat load of time. The following key computational methods should highlight the key elements of the problem and how a program of this sort will be perfect for this task.

## Thinking Abstractly

Within this project, I must apply the need for abstraction. This is because such a complex task like this can be far more over complicated without precautions. So, I will try to:

* Remove all details within the GUI and make it box basic as I can. For example: Just a Button to calculate, inputs for file location, and a visualization opener button.
* The visualization will only include representations of power distribution on an example map to show where power is flowing to at what time (represented via a time scale clock).
* Get file formatting to be produced in my own file format to make sure that data is not overlapped from other file types by accident (an error system will be in place to prevent this).
* Make sure data printed to the console will be minimum and printed separated by lines so that data is separated.
* Data will not be directly sent to a supplier as this is inefficient and pointless. Instead, the response will be either outputted in coloured text in the console or in a separate file.

## Thinking Ahead

I require this computational method within my plan to develop this program because it will allow me to think in advance and develop my program to a more efficient standard without thinking about the crucial structure of the program as I work through it.

* My initial plan is to use C++ with no or few external libraries as this is a raw task that will require my knowledge of how to manipulate my code into my final solution. I might use a small library for reading and writing files, but apart from this all libraries will be C++ standard eg: ‘iostream,’ ‘string.h,’ ‘windows.h.’
* All of the interactions from the user will be inputted via the GUI which should allow for appropriate inputs.
* All outputs from the program will be outputted through either a file format, visualization program, or console. This will allow me to give reach many types of audiences. Hopefully, this will make it easier for a particular person to interpret the data outputted.
* Ideally the more inputs we give it the more it will optimize itself to give a more accurate model which should indicate the key elements of the machine learning part of the program.
* The visualisation will be simple to understand and will show a sped-up or real-time scale of electrical transfer of energy. This can either be done via setting up a matrix of connections and assigning an area a part/value of the matrix. This is something I will address in more detail closer to the development.

## Thinking Procedurally

This will be the process of breaking down the main structure in order to see all the key elements we must develop. This will be especially useful as I have a lot of algorithms to develop, and the entire process is a lot to develop at once. To break down the problem further I will use the use of a hierarchy diagram which should show what elements I must develop as a part of that area of the project.

This project contains four primary problems that I will need to overcome. And within these problems, there are more problems that I will need to overcome. I hope to develop a larger hierarchy chart to show the algorithms in more detail with flow charts in the design phase of development. The darker segments represented in the chart are algorithms that I will need to approach. The others are problems that I will face as an element of the primary problems.

= Algorithm

= Problem

### GUI (Generated User Interface)

This is the segment where I give something to the user on runtime in order for them to upload their inputs. This will be mainly because a terminal is rather slow to input data into. Rather a GUI looks a lot better and is less likely to run into problems as it will only take the inputs placed in the boxes. Hopefully, this will generate a better user experience.

### Data Interpreter

The Data interpreter will be a main key feature of this type of program. This is generally due to the fact I will need to take large data tables from a readable file. After will include a series of processes including:

* Compilation of data into a single set of variables/array.
* Use of data to place it into a format that the response commands can interpret.
* Consistency needs to be crucial to make sure data is being read exactly as I set it to.
* Optimization needs to be to a maximum, to ensure minimum system requirements are kept to a minimum.
* Reading large sets of data should be done via multi-threading.
* Using a machine-learning algorithm to give better values for given data

### Response Calculator

This is an area that will become overly complicated at points. I am hoping that I can minimise the complexity by individually introducing algorithms as I proceed. It will include the bat algorithm that will use the process of ‘eco-location’ to find the most efficient solutions to many of my problems. However, this is the part that will take all the data from the variables/ array from the data interpreter and use it within calculations. The final value will be calculated by using an index or sort of dictionary of scenarios and getting some value based on this. The value produced will be taken and overapproximated. This is critical in the solution because an under approximation and incorrect prediction can cause a lack of energy supply. This is the only authentic way with the bat algorithm that we can make the algorithm the most optimal I can.

### Outputs

Outputs Are going to be a critical part of this project. Not only for the final product but for troubleshooting. I need to make sure that all my outputs are working up to the best standard so I can identify issues and fix them efficiently. But on the finalised product, I need something to prove my concept is working. If it is the visualisation as an output will show this. What is more the terminal output should show this too. File outputs will be a critical part too. I need to make sure that they can be interpreted easily so they can be reused to get the machine learning aspect to learn more.

## Thinking Logically

The main thing about my project to make it solvable is that I can use my current knowledge of applying my logic in order to hopefully produce a better response. An example that I will use of this within my project is: If there is an input of variables into the response calculator, start the response function and this will calculate all the ideal values. This will run through the bat algorithm and iterate producing slightly more accurate results until the condition is satisfied which is that it has either run out of data to analyse or has reached max iterations. For this reason, this will be in a loop. My other idea is that it will run forever until it produces a solution of precision close to other related results. For the machine learning I will need to define some values that will assess if a day is hot, cold, busy, weekends ect to determine what will be the final result. Eg:



Conditions: temperature: -3oC, Wet, Busy, Time = 5 pm.

Peak\_hour = 0.95

Coldness = 0.87

Current\_Demand = 0.55

Busy = 0.60

Work\_Hour = 0.13

Industry\_Hour = 0.31

Result

Demand = 1.5 x greater than current

Peak\_hour is when everyone leaves work.

Coldness Is how cold it is.

Current\_Demand is the current demand upon solution calculation.

Busy is how busy the area is.

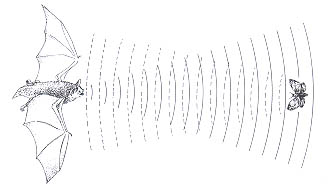
Work\_Hour is when people are working.

Industry\_Hour is when industry is working: eg: factories.

The results are 1.5 x greater than the current demand to increase production.

## Using Backtracking

Using a method called backtracking will allow me to make better calculations in my work. This is the method of taking a route/ specific path and then if it does not work go back and try another route. It will help me when looking into my bat algorithm. The basis of the bat algorithm uses echolocation which can be interpreted as a sort of backtracking. It uses the method looking for items ahead, if there is none it will check another location. For this I can use a diagram:



In the context of this diagram, if the bat were to try to locate the butterfly upwards, for example, it would not find anything, but it has looked to the right and something that is close to him has been located. Therefore, that will be the thing we schedule next.

## Thinking Concurrently

This is going to be a huge part of my project. My project will ideally have the ability to multi-thread. Meaning more operations can be ran at one time. This will especially come into play when we are thinking about the CPU rendering the visualisation, whilst the GPU does massive amounts of repetitive calculations. All this is going to be happening at 1 time. And for me to think about how this will work now will set me forward when I develop as I will know how and where to send the data.

A picture containing text, projector

Description automatically generated

GPU

Shape

Description automatically generated

Solution

Task Input

CPU

Processing Data

## Conclusion

All of the listed examples above of computational thinking have helped me to break down this problem as I proceed with this project. When I know how to think about how to solve the problem in smaller sizes and worry about one piece at a time, I know I can make a more efficient solution. It helps me to think about what I need to include in this project and what would be pointless in this situation. Since the project can be deconstructed via these methods, it is a suitable idea for a computer program. My overall final product will now have more structure as I know what bits I need to separate, and I know where to send them. The final product will be a tremendous success for not only my stakeholder but for the concept that a bat algorithm can be used to solve huge tasks more efficiently. Showing this will prove that this algorithm can be used anywhere if it can be placed in context. And the energy industry would have a more solutions on how and when to distribute their power.

# Research

The main problem with this stage of the analysis is that there are little to no resources for completing a task this extensive. Most of this sort of software is extremely closed source as it could be a security risk to expose it. However, the possibilities are endless, for example, the same context of algorithms can be used to predict the stock market even more precisely.

## Initial Chat with Jonathan

### Interview Plan: 14/06/22

As I am still in a bit of curiosity about what Jonathan would like from this project. I will need to follow up on some questions that I have about the project.

An interview is an Ideal form of collecting data for this program, although I would not directly call it an interview as he educated me on the topic and informed me of what this program could consist of. This form of data collection ensures that I get what he would like out of It, so I know what to stick to as the foundation of the program.

My questions are as follows:

What does an ideal output consist of?

* How would data be represented?
* Would data exportation be anything I can work with within this?
* What are the main calculations obtained from this program?
* How precise do the outputs need to be?

What is the Bat algorithm?

* What is the bat algorithm?
* Where is it used?
* How does it make a difference to a proposed algorithm?
* How can this apply to this program?

What is the need for this program?

* In what industry will this be used?
* Why will it be used as opposed to their current software?

What would you like to be the final goal of this program?

* What will it achieve in this industry?
* How will it affect the response of the power grid?

## Interview Overview: 14/06/22

### What does an ideal output consist of?

* How would data be represented?

Well, I could be anything you really want. As long as it represents the primary concept of the algorithm. For example, visualisation of distribution would be a great scenario to use. Or even basic text table outputs.

* Would Data Exportation be anything I can work with within this?

Yes definitely. Especially when a person would like to export the data into other forms for example a spreadsheet, it will become critically important to do so.

* What are the main calculations obtained from this program?

The main thing I will need from this program is to prove the concept calculation time is reduced. The outputs should consist of the output power for a given time and a sort of original power. It should be able to slope down/up to demand.

* How precise do calculations need to be?

Ideally, they need to be fairly precise. The industry that it is aimed towards is one that will lose millions if the wrong calculation is resolved. Most of the calculations overapproximate due to the fact that there may be an increase in demand at any one point.

### What is the Bat Algorithm?

* What is the Bat Algorithm?

The Bat algorithm Is used for optimization. It looks for the most efficient way to schedule commands. For example, in this project, it would be used to know when and where to do calculations. The overall idea is that it should be able to make a more efficient response.

* Where is it used?

Today it is used in very few open-source ways. It is still a very new concept and was only theorised in 2010. There are a few bits of evidence it has been used to solve simple problems, for example, sudoku problems, or chess algorithms. Your purpose would be to propose it as something that people could use more often.

* How does it make difference to the proposed project?

For now, all we need to do is get it into a program like this working and showing satisfactory results. However, we can improve the algorithm as we learn more about what results it produces to potentially reduce calculation times and improve the efficiency of what tasks are run and in what order.

* How can this apply to this program?

At the moment it is very unclear how you will be able to apply it to the program. That is an element you will need to figure out. Where you place the algorithm will determine how it will perform later on.

### What is the need for this program?

* In what industry will this be used?

This will be aimed at my clients across the industry. Because ultimately if we can prove the concept of the bat algorithm is working in this context, we should be able to apply it to any similar scenario. This should improve efficiency in any industry that requires large calculations. The energy is purely a scenario to get this concept to work on.

* Why will it be used as opposed to their current software?

If energy companies were to see if they could cut the times of their calculations, they could be more efficient in their work. Less calculation is required to complete a solution and ultimately more results can be produced as a result.

### What would you like to be the final goal of this program?

* What will it achieve in this industry?

Within this industry, it should be able to get more accurate results or at least present the benefits of using this algorithm within their company to get better results. Even if we cannot prove it, it would contribute to the little research there is on this algorithm.

* How will it affect the response of the power grid?

The power grid is essentially millions of power connections, and the power grid will respond in a chaotic response. It will be hard to show where everything is going but that is ultimately not the goal for now. Your algorithm should just output the information that will be carried to the power plant directors and used as a reference.

### Review Of the Interview: 14/06/22

The interview really gave me an insight into what actually this algorithm will do, so I greater improved my knowledge of how to solve it. But it gave me a baseline on what Jonathan would like from it. It showed me exactly what it will be used to prove too.

The main things that I know now is that; comparing the data to show, with or without is crucial; Making sure that the Bat algorithm is used and applied in this context; It will not be an easy task to solve and as long as I can briefly show some sort of results and optimisation it will be time well spent; Trying to output the data in a simple to understand for would be critical to show the concept of the program and algorithm.

So, my final list of things that I will include initially is:

* A visualisation algorithm to show data
* An option to calculate without the Bat algorithm
* Total processing times
* File reading and writing in my desired format
* Simple inputs for the user place a calculation

## Bat Algorithm Implementation on Economic Dispatch Optimization Problem

### Abstract

This is taken from an article written by Lili A. Wulandharia, Siti Komsiyahb, and Wisnu Wicaksonob, in2018 who are the heads of Computer Science and Mathematics at ‘Bina Nusantara University. They first theorised that it could be used in this exact use and that it would optimize the cost of running the power plant. Producing enough to demand and with appropriate cost would be ideal for this industry. It states that many people have theorised about the use of the algorithm in similar contexts, but it still needs work to work on the ‘computational efficiency’ of the program.

The article goes on to state that the current algorithms of powerplants use the use of the ‘firefly algorithm’ which can save up to 0.12% of the cost. Whereas compared with the Bat algorithm which can save up to 1.23%. In the context of this industry, we could be talking about saving millions a year on energy costs.

### Introduction

It states that there are two main types of power plant, the thermal and non-thermal power plant. And that in a thermal power plant most of the costs are towards fuels which is a non-renewable energy source eg: petroleum or coal. Since this is a huge cost towards the industry, optimisation is needed. This is so that a power plant can reach ‘maximum capacity.’ And the approach stated is called ‘economic dispatch.’

Graphical user interface, text, application

Description automatically generated

The article shows the need for the program, and it also shows the problems that may occur when trying to create something to show this. It also emphasises the need to reach demand, which will be a huge part of my project. And ultimately, they proved the same thing I am trying to prove.

A picture containing text

Description automatically generated

This states the uncertainty within our results that we could encounter. This will be massive when we are calculating results. As a result of underapproximation is power cuts and loss in profits for the power distribution industry.

It states the significance of seasonal changes, because a hot or cold day could mean less or more power consumption. This means that the algorithm should be able to calculate to this. It also shows how the bat algorithm works via a method of ‘echolocation behaviour.’ Echolocation in bats helps to find their pray in complete darkness.

Text

Description automatically generatedGraphical user interface, text, application

Description automatically generated

This is all the initial equations that I will need to do with the overall inner working within a power plant. It just shows how I can find the total power, or cost of generation.

### Economic Dispatch Problem

This is the section where the Bat algorithm is explained and how it is calculated through the equations 1-10. I have read over this text, and it seems to be the case that it uses a set of items to follow, step-by-step. Index:

* Pi = Power generated by a generator
* PD = Power demand
* PL = Power Loss

The first equation states the away we can calculate the economic dispatch. This is via taking the fuel price, and the power and 3 coefficients that are constant. n is the number of iterations and the number of generators in the system. And the equations gives the total cost of all the generators.

The power demand equation is stated in the second equation. It takes the power generated by all the generators, and then subtracts it from the lost power to get the demand for energy given by PD.

Graphical user interface, text, application

Description automatically generated

This part is a bit complex and would be hard to interpret the coefficients without using enormous amounts of data to calculate them. This is probably what I will do in my project. However, this uses a double iteration method, eg: a for loop in a for loop.

### Bat algorithm to determine optimum cost in economic dispatch

Text, letter

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

This part is the step by step to the Bat algorithm. In order to code my algorithm, I will need to be able to interpret this as best as possible and get the main principles of what the Bat algorithm works off. This article will give me an idea of how it works.

Now I will skip a bit, this is in order to get the outputs the algorithm and highlights the need for a bat algorithm. The tables below are compiled into a graph.

Table

Description automatically generatedTable

Description automatically generated

The coefficients are used in the equations above. I might end up using these coefficients for my algorithm as they seem to give precise results. I could also use machine learning to refine these coefficients too. However, the results from the tables seem to show that the Bat algorithm meets an extremely good demand. It does underestimate some of the time. However, it can be adjusted to overestimate slightly (eg: by the highest value that it was off by).

Table

Description automatically generated

This graph compares the amount that the bat algorithm was off by. This is used to plot into a graph below.

Table

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

This graph is very impressive. It shows that the results are extremely precise. In most cases it is less than 10 units off from the true value. So, the approximation is particularly good at that. It shows what my results should look like after I manage to get a working set of algorithms. Furthermore, the calculation of the cost is shown.

This shows that in some cases the Firefly algorithm is just as precise as the Bat Algorithm. However, this is compared with small data sets. And what is more, my task is to try to improve this. If I can or not is another question. However, it shows the Bat algorithm saves 1.23% and the firefly saves 1.16%. So, it is not far off, but over millions of barrels of oil a year it will take cost down a lot! We could be talking about near to 1-10 million in savings annually per company. If I can improve this to the predicted standard, it should be able to save up to 1.5x this compared to the firefly algorithm.

### How I Plan to Implement This into My Program

To apply this to my program I will need to keep in mind a few things:

* The algorithms and how to use them.
* How to use the bat algorithm step-by-step.
* How to optimise this algorithm further to hope to get better results.

I hope to get results like shown in this article. And if I can, I should be able to apply it to my own situations. The idea is that I can get to visualise it too, to express the program in a way that people can interpret visually.

The algorithms presented make huge impacts on the accuracy of the results. However, I would like to iterate the process many times to get more accurate results. The fact as well that I will need to repeat this many times so I can get results as the situation changes, will add to the hardware demand of the program.

## Gas Demand Forecasting Methodology by The National Grid

This document is especially important for me to understand how the power around the UK is distributed. It will help me to see what demand is like at different conditions and areas. This was all taken from an open-source document from the national grid’s documentation library at [Welcome to National Grid Group | National Grid Group](https://www.nationalgrid.com/). Firstly, the first snippet is on the zones of distribution in the UK.

### Distribution Zones of the UK

A picture containing map

Description automatically generated

It shows what areas the UK is split into. This will be useful when I am designing my project to see where I need to focus on demand.

### Factors that Affect Demand

Diagram

Description automatically generated

This photo above shows the main factors that affect the energy demand. This is what I will consider when I am calculating through the Bat Algorithm. As we can see the NTS which industrial demand is having varied factors to what effects the LDZ which is considered residential.

A screenshot of a computer

Description automatically generated with medium confidence

This part just explains the main factors represented in the diagram above. This will help me when use my equations too.

### Composite Weather Variable

Chart, line chart, scatter chart

Description automatically generated

This graph shows the basic negative exponential correlation such that if the temperature is higher the demand is lower. This is also modelled with the CWV (composite weather variable) such that this could be one of my coefficients for the weather factor and is given as a linear function. The idea is that my algorithm should be able to make something like this within the algorithm and predict the demand. It will help if I can use some calculations if I can predict the demand if the weather is a main factor on one day.

Text

Description automatically generated

This is used to explain the CWV. I read this before continuing so I knew exactly what the graphs were plotting against.

Chart, histogram

Description automatically generated

This shows the CWV that is just a linear version of the demand annually. We can see during colder months that energy consumption in the lowest in fact, which is opposing the other graph. Hotter months tend to have more demand. The historic range also shows this. It tends to be the same every year by the looks of things.

### Daily Demand

My project is more going to be predicting the daily demand although I will need to take in account for tendencies of temperature during the year.

Chart, scatter chart

Description automatically generated

Again, this graph has the composite weather variable on the x axis and shows the demand of energy as CWV increase. It seems the higher the CWV the lower the demand, even on holidays and weekends. This is against what I would expect. However, all the factors so far produce nice linear or predictably exponential graphs so I can predict it a lot easier.

Chart, line chart

Description automatically generated

A picture containing graphical user interface

Description automatically generated

These are all graphs showing how the composite weather function is used. It is an extremely useful variable to use for my calculations. And will allow me to relate factors to demand required.

Graphical user interface, text, application

Description automatically generated

These are some equations I may need to use in my calculations.

### Predicting For the Future

For my model to work, I need to look at the past and see how energy demand has changed and increased. Demand will only need to increase and If I can get an insight into how much I can predict within my algorithm a lot better.

Graphical user interface, chart

Description automatically generated

This graph relates all the way back to 1928/29 and shows the demand back then. This was monthly demand. And today a month’s demand has become a single day’s demand in most areas. And in some it is much, much higher than this.

### Calculating Demand

This is a step-by-step guide on how to calculate the demand. I will not explain any of it, as it is all explained in the extracts. Again, increadalby useful, all I will need to do is impliment this into the Bat algorithm intead of doing it all by hand.

A picture containing text

Description automatically generatedText

Description automatically generated with medium confidence

Chart

Description automatically generatedGraphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

This is what I need. It is a graph of demand against day number. It shows how the demand decrease over the threshold. If I can calculate if it goes over the threshold I know when demand will decrease.

## How I Will Implement This into My Program

## Again, this extract was extremely useful. This presented everything that I could have imagined seeing about finding the demand at a point. Hopefully, this will help me to calculate the demand given the variable conditions on a given day. I think the main things I got from this piece of research is to include:

* A composite weather function / variable as a coefficient of my equations.
* How I can calculate and represent the demand at a given time on a graph.
* What sort of values my dictionaries of values need to contain.
* The main factors that affect the demand for power at a given time.

Although it is not a lot I need to consider, it helped me to understand more about the demand and the key factors that affect demand. This will let me know what coefficients I need to get my value.

## Pre-development Chat with Jonathan

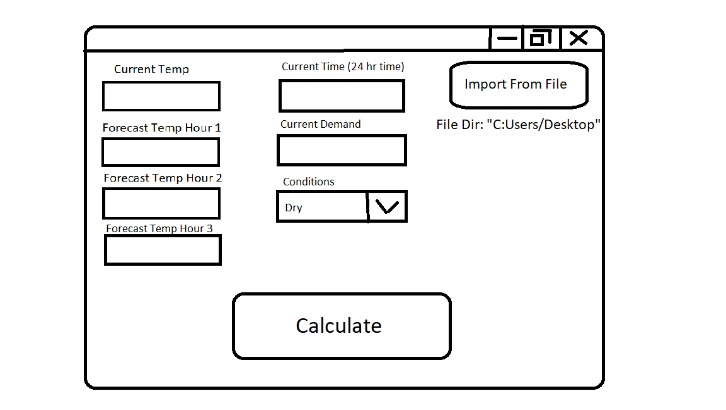
### Interview Plan: 17/06/22

How do I calculate the response?

* Where can I get data sets from?
* Can I use these data sets without limitation?

Would a GUI be an ideal way of taking Inputs?

* Would this be a good GUI layout?



* How would the user be able to easily verify where inputs want to go?
* Could a help file support the user to transverse the program?

Would machine learning be something to make better results?

* Where in the program will the machine learning improve?
* What will the main improvement from using machine learning?

Are there any additional things you would like me to include before development?

### Interview Overview: 17/06/22

### How do I calculate the response?

* Where can I get data sets from?

When I was working for an electrical company in Ireland, I obtained huge amount of data in a database. This should help to complete your task.

* Can I use these data sets without limitation?

Essentially yes, but they must not be shared with other people. It is confidential information to our company.

\*SELF-NOTE\*, I will not share the data sets on this document. I can show values they obtain, but the main database of values I will not be able to share.

### Would a GUI be an ideal way of taking Inputs?

* Would this be a good GUI layout?

That looks like a good input system. As long as it takes in the key values and it’s not too hard to use, it works with me.

* How would the user be able to easily verify where inputs want to go?

Something like labels will be ideal, just let the user know what is going to go in that box. You will need to make sure all fields are filled else it will cause problems later on.

* Could a help file support the user to transverse the program?

It could be ideal, it will really depend on how well you present the inputs on the main page, if they are not too clear you can explain them in a help file.

### Would machine learning be something to make better results?

* Where in the program will the machine learning improve?

It should be ideally used in the making of the coefficients stated from the documentation I sent you. Hopefully, this will take the best results and improve the coefficients.

* What will the main improvement from using machine learning?

It will be able to self-improve. For example, if we take 20 thousand entries from a time scale of 1 week, we can see consistencies within the time and effecting factors to give better predictions for demand.

### Are there any additional things you would like me to include before development?

Well, no, as long as you approach the main topic and manage to address in some sort of way that will give more optimal results it will achieve all of the main targets that I would like from this task. I would however like you to investigate distribution of the tasks, hopefully this can be used in my client's context to achieve results quicker.

### Review Of the Interview: 17/06/22

This interview indicated some key elements to how I can calculate the values given. It also showed me what sort of things would be ideal, eg: distributing the load over a network and allowing for this in my code. One of the things I was uncertain about is what he would like the GUI to look like, now I have a concept to what I could possibly look like in the future. It would also be relatively simple to develop too. I also got the main thing I was missing in order to proceed with the machine learning aspect, which was the vast amounts of datasets that I would need to even calculate accurate coefficients. The more entries are had, the better as my results become more precise.

## Main Features of The Solution

These are the things I need to make sure I include this, so I get to the main points of the problem:

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Reason | Why It is Essential? | Reference |
| GUI with input boxes and buttons | To make the inputs from the user easy to do and not overcomplicate this process. Makes it quick to input data sets | This will be the main input for the user to interact with the program. I have chosen a GUI as an alternative to a console as it makes it simpler to interact with the Program. | Interview: 17/06/22 |
| Reading data sets from files | To make it even quicker to import huge data sets that need to be worked with | This will make it a lot quicker to input values. When where talking about thousands of data items to input, it would take hours without a system to read the file. | Interview: 14/06/22 |
| Bat Algorithm | To make my program the most optimised it can be. | This will be the most important feature of the project. The whole point in this idea is that it is a proof of concept. It is to prove it can be used in the industry. | Bat Algorithm Implementation on Economic Dispatch Optimisation Problem |
| Data representation | Console outputs to see what is going on during calculation and view a particular solution that will be highlighted. | This will be a main feature for my piece in mind in debugging, and for a data scientist to be able to analyse the data with more accuracy and see where the data is tending towards. | Interview: 14/06/22 |
| Visualiser | To represent the solutions to compare them to the true values and show how much power will be distributed. | Comparing the data will be key in this program to compare it to other forms of optimisation algorithms and without the algorithms at all. It will hopefully outline that the concept works. | Bat Algorithm Implementation on Economic Dispatch Optimisation Problem |
| File exporting | To export data tables of solutions and basically gives a set of commands for the power plant to follow | Alike the file importing, it will be used to export the data sets. A single data set could consist of millions of outputs, and to copy and paste this hundreds of times would be inefficient. | Interview: 14/06/22 |

I can get all of this done within the time given. I hope to address all the little bugs as well after development.

## Optional Extra Features

|  |  |  |
| --- | --- | --- |
| Feature | Reason | Limitation |
| Time Limitations |  |  |
| Mini map of the national grid to visualise | Shows a real scenario of the reaction from the grid compared to the real value. Will show if it meets demand ect… | Is exceptionally long to code. And would require a separate GUI. It can be intensive to run and what is more, it requires information on the national grid I do not have. |
| More calculations for distinct types of power plant and not just thermal power plants | Will give more adaptability to the industry and opens it to a wider range of possible clients. | Again, is longer to code and I would have to make up my own equations. I will investigate this at a later date. |
| Allow it to be used in a distributed network | Would use a central controller and little stations that could calculate solutions individually. Calculations are quicker. | It is again hard to code. Would require precise data packet transfers across a local network. I just do not have the time for it. |
| Hardware Limitations |  |  |
| Server Compatibility | Would allow for it to easily be installed on a server network and be calculated there. | I do not have a server I can use directly and reliably give results. |
| Raspberry Pi Distribution | To use a central hub of a Raspberry Pi and use separate ones in a server configuration to communicate and calculate. | The raspberry pi’s at school are just not powerful enough. I could do it, but we were talking about all 30 or so of the Pi’s together to produce a solution a computer with a simple GPU can complete in 30 seconds. |
| Software Limitations |  |  |
| 2D rendering of visualisation | It would help to produce a cleaner visualiser. Also, will allow for demand to be taken off the CPU. | However, it is hard to use even one GUI library in C++, let alone two. I will leave this for now. |

# Prerequisite Requirements

## Hardware Requirements

In advance of my requirements, these are optimal specifications. This is due to the fact this program will be ridiculously hard to run due to the large, repeated recalculation it will require. Under specs will only cause an increase in time for the solution.

|  |  |
| --- | --- |
| Hardware | Reasoning |
| 32GB of DDR4 RAM / Memory | When loading all the data tables onto the program they will all get stored in memory. Many will be duplicated, saved, and recalculated so the memory address used will be used up very quickly. |
| GPU With At Least:   * 8000 Base Cores * A Base Clock of 1.2 GHz * 16GB of VRAM | The repeated calculations will be run primarily on the GPU, eg: Response Calculations. This means a hefty GPU is required to run millions of operations a second in minimal time to produce a response that will take upwards of a billion calculations to complete. It will also take charge of the machine learning aspect. |
| Data Bus with a Capacity of 64-Bits | This is generally available in most machines today. This is because a 64-bit operating system requires a 64-bit bus. And most people run a 64-bit computer. |
| A CPU With At Least 8 Cores and 4GHz | This will take the bulk of the commands in terms of converting data. However, all multi-threading commands will be sent to the CPU so if someone wants to visualise something they must demand on the CPU instead of the GPU. |
| Ideally A Supercomputer / A tonne of Raspberry Pi’s. | Most companies that this product is advertised to will have immensely powerful computers with hundreds of CPU cores and 10s of powerful GPUs. This is all to ensure that they can produce reliable responses. |
| A Display | In order to show results visually, although is not required. |
| Data Logger/ Inputter | Will allow for data to be directly inputted into a file and interpreted within the project. Reduces the change of human error on transfer. |

Again, this is a lot, and my target audience is someone that will have a lot more than this. Even with these specs, I expect calculation times for a single set of results to take upwards of 10 minutes. Ideally, a company should have the hardware that will be able to calculate this within seconds. And what is more, my Bat algorithm is the main testing feature of this, If the algorithm works it will skim predictably up to half the time. This type of software is **not** for your average user.

## Software Requirements

There is a very minimum amount of software requirements, this is because there are little to no external libraries that will be used. The only one I may plan to use is GLFW which is an open-source graphical processing library.

|  |  |  |
| --- | --- | --- |
| Software | Reasoning | Download |
| Windows Operating System (10 or greater) | The only way to run this code, for now, is on windows servers since it will take a lot of extra libraries and instalments to get it to work with other OS like macOS or Linux. | [Download Windows 10 (microsoft.com)](https://www.microsoft.com/en-gb/software-download/windows10) |
| Visual C++ Redistributable 64x/86x | This is the standard runtime for C++ files that allows for compiled C++ files to be ran. It also imports all the standard libraries for C++. | [Latest supported Visual C++ Redistributable downloads | Microsoft Docs](https://docs.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170) |
| Development Tools: |  |  |
| Windows SDK | Allows for the development of the program if development is required. | [Windows SDK - Windows app development | Microsoft Developer](https://developer.microsoft.com/en-us/windows/downloads/windows-sdk/) |
| GLFW Library | This is used for graphical rendering. In this case, I might use it to develop the GUI. It would be also particularly good at representing the visualisation. | [An OpenGL library | GLFW](https://www.glfw.org/) |

# Success Criteria

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Requirements | Reasoning | ☒ | Reference |
| 1 | GUI opens on runtime. | So that I can get an image that will be used to take user inputs. | ☐ | Interview: 17/06/22 |
| 2 | GUI renders Inputs. | Makes sure users can interact with the program. | ☐ | Interview: 17/06/22 |
| 3 | GUI allows taking inputs into variables. | Makes sure inputs are placed into the program for calculation. | ☐ | Interview: 17/06/22 |
| 4 | GUI has an exit option. | To get out of the program without having to go into task manager. | ☐ | Interview: 17/06/22 |
| 5 | File opening and reading system. | To read large data sets quickly instead of typing them in one by one. | ☐ | Interview: 14/06/22 |
| 6 | Variables are interpreted into the Bat equations. | To make sure variables are passed. They will be ready to be calculated on. | ☐ | Bat Algorithm Implementation on Economic Dispatch Optimisation Problem |
| 7 | The Bat algorithm correctly calculates the results I need. | Uses equations to get results and give an idea of the result in need. | ☐ | Bat Algorithm Implementation on Economic Dispatch Optimisation Problem |
| 8 | The Bat algorithm gives a more optimal result. | To compare to the other results. And makes sure the Bat algorithm works to the best of my ability. | ☐ | Interview: 14/06/22 |
| 9 | The data interpreter uses comparisons of other data tables to give better more accurate results. | Makes sure that the data inputted is compared with other items to see if they have equivalent properties. | ☐ | Bat Algorithm Implementation on Economic Dispatch Optimisation Problem |
| 10 | The response algorithm uses machine learning to get more precise coefficients for the equations so that we get better results as we feed it more data. | Takes these comparable properties and compares them to get an average and a new average coefficient. | ☐ | Bat Algorithm Implementation on Economic Dispatch Optimisation Problem |
| 11 | The console outputs the variables/ solutions I want it to. | Makes sure no unnecessary items are printed. And gets it to show all the working of the calculations and steps. | ☐ | Interview: 14/06/22 |
| 12 | Solutions are highlighted to make it clear they are solutions. | Makes sure the person reading from the console can see what a solution is. | ☐ | Interview: 14/06/22 |
| 13 | Solutions get written into a file in a table format so they can be used to prove my concept. | So, they can be used on other programs. Or if a power plant wants to use that as an input to their system they can. | ☐ | Interview: 14/06/22 |
| 14 | Over approximates the answer depending on a calculated uncertainty constant. | Makes sure that results never go under and always go over to always reach demand. | ☐ | Interview: 14/06/22 |
| 15 | Solutions are copied into the visualiser. | So, they can be used and interpreted by the visualiser. Will use new variables. | ☐ | Interview: 14/06/22 |
| 16 | Visualiser opens a GUI/ console window and displays a graph. | Makes sure something opens to show the graph | ☐ | Interview: 14/06/22 |
| 17 | Draws two graphs, one for the true value, the other for the calculated value. | On graph GUI runtime the graphs will be drawn to make sure we have something to plot the data against. | ☐ | Gas Demand Forecasting Methodology by The National Grid |
| 18 | Visualiser plots graph and shows the comparison of true value. | Ensures that the data I am getting is precise and will be a good algorithm to use in a real application. |  | Bat Algorithm Implementation on Economic Dispatch Optimisation Problem |
| 18 | Plotted in a bar chart to compare the values. | Helps to see the difference in the two values. Will also be drawn after the graph. | ☐ | Bat Algorithm Implementation on Economic Dispatch Optimisation Problem |
| 19 | The visualiser shows how far it was off from the true value. | Let’s us see how much we were off by and if this value decreases as we add more results, we have a fully working algorithm. | ☐ | Bat Algorithm Implementation on Economic Dispatch Optimisation Problem |
| 20 | The visualiser shows a mini-scale map and represents the power exiting the plant. | To visualise what is going to happen to someone who does not know what all the data items mean. | ☐ | Interview: 14/06/22 |
| 21 | GUI will not allow you to exit immediately during calculation. | Stops data from leaking and sudden stop in large calculation can cause damage to your computer. | ☐ | Interview: 17/06/22 |
| 22 | The project will not be subject to memory leaks. | Stops extremely high memory usage on the computer and makes sure the program is instead using that memory for the calculation. | ☐ | KEY FEATURE |
| 23 | An option on exiting the visualiser will display. | Ensures on exit of visualisation that the GUI is opened, and you do not have to reopen the program. | ☐ | Interview: 14/06/22 |
| 24 | Opening the visualiser will close the GUI. | Makes sure that resource usage is lowest it can be at one time. Reduces hardware requirements. | ☐ | Interview: 14/06/22 |
| 25 | Visualiser has options to switch between Bar chart, Map, and graph | To make sure we can see what way we would like the solutions to be interpreted as. | ☐ | Interview: 14/06/22 |
| 26 | Use of the composite weather function | Allows me to predict demand relative to the temperature. | ☐ | Gas Demand Forecasting Methodology by The National Grid |
| 27 | Dictionary of values equivalent power value | Hopefully will make it easier to get a value for power instead of using a function that could be inaccurate for that value. It will be like following a table |  | Gas Demand Forecasting Methodology by The National Grid |
| 28 | The Data interpreter machine learning aspect is proven to give better results. | Will show that the machine learning is working for my scenario and is efficient and not pointless | ☐ | Interview: 17/06/22 |
| 29 | The Response calculator is proven to give better results. | Again, to prove that the program is better with it than without it. So, I know I am getting better results with it. | ☐ | Interview: 17/06/22 |
| 30 | The Bat algorithm is proven to work in this scenario. | Ultimately, this is what I am trying to prove. If I can get all of the above to work and for calculation times to be less or solutions to be more accurate | ☐ | Interview: 14/06/22 |

Design

# Overview Of The Design Phase

For this section of the project, I will need to address how I plan to tackle the problem, and ideally what it’s going to take to complete the problem. In this current moment, I have a few brief ideas on what I can do, and after the interviews and reading up on articles about the algorithms that we would require, I have really got to grips with what I will need to develop in the development phase. However, that is later a task, and for now I will focus on key aspects, for example, what the project will look like in the graphical user interface, decomposition of algorithms, i.e.: flow charts, systems diagrams, and pseudo code. The more I’m able to decompose now into smaller abstract pieces the easier it will make the final problem. I will also mention all of the mathematical equations and coefficients I will be using in the project here too. I will also do some brief research into how I am going to be able to implement a GUI into C++ (which is a heck of a task by the way), to make user interactions more efficient.

# Programming Language

Within this project I have proposed to use C++ as my programming language of choice. There are many reasons and benefits for using C++ for this type of project and I will try to overview the main aspects of why. Firstly, let me introduce you to C++, and the critical features of it. It is both a procedural and object orientated programming language, you can easily swap between elements of both and combine contents of each to make a more effective program. All of the family of C offer a use of this. However, unique to C++ is the use of namespaces. Namespaces allow a developer to specify what scope the items are accessible through. This allows me to make certain functions ‘invisible’ to other structures in different namespaces. Making effective use of this can lead to more efficient code as the program ignores certain components on compilation, and what’s more less memory is used as a result. As we will be initialising thousands of items and iterations of items, this will be very effective when it comes to the execution and practicality of the program, if I can develop it in the most productive way possible. And finally, I would like to expand onto the expanse of external libraries available for C++. Since it is a widely used programming language upon the open source and corporate community, there is no end of libraries to make use of. For example: If I wanted a reliable library to draw boxes on my console screen, I’m sure with a bit of research and time, I can obtain a library that will do this for me within a few lines of code. Hopefully, the use of libraries will shorten development time and reduce the amount of code I will need to write in total. It will also allow me to focus on the main problem rather than focusing on developing tedious string declaring and console printing libraries which are already very well refined by other, more skilled, developers.

# What I Plan To Achieve From The Design Phase

The idea for this part of the documentation is to overview what I will need to focus on and how I plan to develop it, as mentioned above. However, from this, what I can gain is experience in the practice of decomposing and preparing a program for development. I am keen to see how I can develop this project and have a mind full of potential ideas, and I should be able to address all of them within this phase. I can also propose if some of the programs will be solvable and if I can find solutions beyond it. I hope to expand my logical and problem-solving ability too, bringing me a better set of skills for later potential projects.

# Systems Diagram

= Algorithm

= Problem

The method of breaking down the key elements into smaller elements within this program is a method of decomposition. It is extremely good in this scenario, as large tasks that can be difficult to solve on their own can be broken down into smaller sub-procedures to develop and link into a central engine for that segment of the program and would be brought together into the final project. I can now have a good idea of how I am going to develop the program and what I should develop first to make the program as simple as possible. I have also split the diagram into segments of algorithms and problems. Problems overlay the things that will be a product of algorithms and could develop into smaller algorithms later on. Algorithms show the main key procedures that I will need to develop to make a whole segment of the code. Some of the algorithms could be split into even smaller sub-procedures.

# Distribution Of Computation

A thought I have come across whilst within this phase of the development is how we could approach one of the primary problems ahead of ourselves. For this, as stated in the title, I have proposed a solution of ‘distribution of computation’. All this means is that large segments of the program instructions are posted to other machines for processing. In this we reduce the load computation on the primary central ‘hub’. Ideally, I will use a cluster of raspberry pi’s and either a switch and small-scale network, this is slower than my second option but is easier with the equipment I have available. Or, my second option, to connect all the pi’s via a SATA or data connection cable. This is faster, but it will be harder to code initially and with little knowledge of the raspberry pi’s circuit structure, is not realistic for this segment. Within the first option I will need quite a few raspberry pi’s and a switch. However, ultimately, I should be able to just leave it, and it will process within a reasonable time, which makes this theory well worth the time investing in. For this set up I will use a diagram:

Central Hub

PI 6

PI 7

PI 8

PI 9

PI 10

PI 5

PI 4

PI 3

PI 2

PI 1

Switch

# Segmentation Of Individual Systems

## National Grid Demand And Cost Representor

### GUI (Generated User Interface)

#### Find a library to render GUI’s

* + This segment will require research into the libraries available to C++ for rendering and generating GUI’s.
  + It will be imported at the top of the program and will likely be what I declare at the very top, with the rest of my libraries.
* Render A GUI
  + A main skill I hope to achieve is to render a GUI and get it to open/run at run time.
  + The overall idea is that I can make user interaction and inputs/outputs of a higher standard and accessibility throughout the program.
  + The main GUI menu should have error limitations to stop it from demanding too much from the computer to render/ causing memory leaks.
* Render Output Boxes And Buttons
  + In order to take inputs and run commands.
  + Should be able to take inputs and pass them into a function.
  + Is a key aspect of basic GUI’s and is generally easy to do with open-source libraries.
* Placing Algorithms In Subclasses In Inputs
  + Functions will be placed into the input/ button subclasses to ensure the running of the function at activation of the object.
  + May require a small library to make sure clicks are detected as of when pressed.
* Button with algorithm that activates upon click
  + Alike the aspect mentioned above I would like to include this to ensure the process of the routeing declared/ mentioned in the button click subclass.
  + The button/component will reliably call into section.
* Shows/prints data into readable and comparable style
  + Will need to show output textboxes/input boxes to show the calculated data.
  + This could be the solutions to a particular problem, or it could be a segment/single solution to a problem with multiple solutions.
* Button That Activates the data interpreter algorithm
  + A single problem that I will arrive upon is that the data interpreter algorithm is far too large to fit reliably and comfortably in a single subroutine in a button.
  + My plan to combat this is to have multiple sub procedures that will be called into the program.
* File Validation Check
  + This will be essential to make sure that the inputted directory location of the file is actually there. It should return an error message if no file was located.
  + Problem is without validation checks it will crash and cause the program to quit if nothing is implemented to stop this.
* Input File Path/Name Box
  + In order to achieve the ability of opening a file we need a directory/name.
  + This means that a file is referenced to where to find it, in which this will be a used as an input to locate the file we want to read from.

### Data Interpreter

* Format File into readable form
  + Formatting a file will be needed to make sure that we are interpreting the correct set of data.
  + It reduced the likelihood of data corruption and overlapping with other data.
  + It is also easily identifiable to what file is to do with the program and what it does.
* Read Line By Line Expanding Into An Array
  + The data will need to be taken as an input
  + Since I need data to be taken individually, I will take it all into an array, which isn’t a very efficient way, but it should be quick and means I will only need to call the function that does this a single time.
* Telling where data should be directed and how it is interpreted into later algorithms
  + All this essentially does is placed all the input data into the array (as mentioned above), but then passes all the values we need (byRef) into the functions we need. (byRef will be important for later).
  + Just passed and directs the data into the function locations where needed. Will hopefully reduce code needed to organise all my data later on.
* File opening algorithm
  + As It wasn’t mentioned above, we will need something to read the file in the first place and place it into an array.
  + Ideally, we will need this to be something that opens without causing new memory leaks, which also tends to be a problem with this.
  + Errors should also cause the file to close upon exit.
* View current data and if current is better than previous then set new best
  + Apart of the machine learning algorithm would be to detect if we are improving towards the new solution. This should identify if we are actually getting better results and where to nudge the bats to (mentioned in later topics).
  + If we have a new best the best data set should be set to the current found best. This will be used later to compare and improve.
* Iterate multiple times to improve many times over
  + In order to determine if the data is getting better in the first place, we will need to iterate as many times per data item as possible.
  + The more iterations we complete the more precise the data is as we converge to the final solution/ best solution.
* Improve values and bat properties.
  + Regarding the bats coefficients, i.e.: initial velocity, frequency etc… I hope to improve these, by taking averages of the best initial values. The reason for the average is I get most of the best elements of the best of each solution.
  + A better coefficients will be determined by shorted calculation times. And should be ran independently to the main solution/ answer generating code.
* Make sure values are optimised for starting of the bat algorithm
  + The values are placed ready for the next iteration, the idea is that if we getter better coefficients per iteration we can improve the overall response/ calculation time.

### Response Calculator

* Use algorithm sequentially
  + This means that the algorithms is referred to in segments and each part of the algorithm is completed one after another for simplicity of breaking down the bat algorithm.
  + This is a key aspect to making the development of the algorithm easier.
  + It will then reiterate through the algorithm for self-improvement.
* Places used data into an array
  + All the data that has been used and their responses are placed into a new array, so we can print and plot the data that has been provided by the output. (Used in a later step).
* Use of previous statistical tables to average this
  + Previous tables that have proven and actual solutions for the load demand and cost will be used to reference to. The idea again is that I can provide more precise values if I use this as a starter test set to improve the values initially then place it on real data values.
  + Overall data should be improved and will ideally provide more precise and quicker results.
* Take current global best
  + We can initiate a current way of taking a new global best that has been found. We can also take a best solution that the bat has ever found and one of the bats will contain the global best as this.
  + To save memory I may reference to this bat instead of placing the global best into a new array.
  + Is all declared and stated in the research mentioned above, more detail provided later on.
* Generator load up time
  + A later set of problems will be to consider for engineering and problems with mechanical limitations.
  + I.e.: a generator cannot initiate and generate energy instantly, so we need to factor this when taking in the price. If the generator is not on, we need to make it costly to turn on, or decide if we can get it turned on by the time we get to a later solution.
* A\* / Dykstra’s to determine shortest path
  + Will be useful to determine where the electricity is going and weather it will be efficient to send it there. If it is too far away its unrealistic to send it all that way and will add to the limitations
  + The A\* / Dykstra’s will be used to find the shortest path to get the minimum distance between the two points
  + Grid is placed/ referenced to as a matrix that has weighted nodes.
* See where paths load shortest demand
  + We should only consider the areas that the demand of the generations will meet the local demand and other locations should use their electricity to provide for their area.
  + If an area can provide more this should be considered for.
  + And if an area is of deficit it, we should be able to compromise and supply from other areas.
* Load a single day onto a single unit
  + Each unit of the distributed computer should refer to a single day to calculate.
  + This means with 6 raspberry pi’s and a single control hub we should provide 6 periods of solutions per complete cycle, Ideally….
  + The reality Is that we will get solutions completing at different time and this is something I will need to factor in.

### Outputs

* Renders map
  + Will show where power is distributed to.
  + To be honest this was just to increase the complexity of the program but may be an ideal feature to show in an easier format where power is going. Also, will be a good way of debugging my shortest path code.
* Shows links between locations
  + The rendered maps will show the paths between the locations, and dependant on distance will have a weight to them, this will be showed either visually or as a number above vertices connecting them.
* View where plants should be lit and lines too
  + The lighting of power plants and lines show the flow of power.
  + If the line is lit it shows power flowing, and not lit means no power.
  + For simplicity I will show the idea path to a location and providing of an area.
  + It will be scaled down a fair amount to show only a few nodes as a large number of nodes calculates very inefficiently.
* Shows path of demand
  + The path of demand is shown to direct the direction of the path.
  + It will show where power is flowing too and what plant is providing to what area.
  + It should show what plants provide to what areas and what overlap to partially provide an area.
* Shows Distribution And connections from Power station
  + As mentioned above this is very similar except a line will be resolved and reliant on the power plant to provide power. And power flows under the rule of flow is in the direction of power plant to location. In exemption to overlay of line powering.
* Shows final response / solution
  + The final response is printed into a console with a suitable direction to what it’s the solution to.
  + This will be printed straight from a solutions array.
  + And will only print the main aspects.
  + Should be identifiable from the other printed data in the console, e.g.: is coloured different to show.
* Shows Repeating Calculation Data
  + This should be optional and referenced to at the GUI.
  + Shows all the improving iterations per period of time.
  + Helps to debug into showing if the solution is converging into the final solution/ a better solution.
* Print with variable separation
  + In a table format in the console outputs the separations should distance the data items.
  + columns will be shown at the top and will CLS to show where data is directed to.
* Formatting Into an interpretable form
  + The file output system will need to able to write to a file and will have to be in a different file type/ name to that as the original. This stop data overlaying.
  + This should also be readable from a text file and importable into a spreadsheet software for data analysis.
  + Would also help for a later algorithm to tell if our results are the most reliable. Its not on the criteria, so I may not attempt this.

# GUI and Graphical Displays

## GUI (Graphical User Interface)

The GUI will show where to input data in a simple and understandable format. The picture I will attach and annotate below is a second draft in reference to the interview conducted in: ‘Interview: 17/06/22’. For which I proposed this solution:

## Diagram Description automatically generated

Under more careful consideration from the requirements of the project I will need more inputs and more options to select. This is my new concept for the main greeting GUI:

Diagram

Description automatically generated

Shows the inputs of demand predictor, and what is predicted to be the demand. All these are factors that affect demand. As referenced to by: Gas Demand Forecasting Methodology By The National Grid. I chose this placement because it is in a neat format that is directly in the way.

Shows the solutions of the current calculated forecasts. It will show in a separate navigable GUI as shown below. Will not activated until calculation button is pressed and calculation is complete.

Shows the input calibration data, if it is the correct format, it will read the file location and use it to improve coefficients. Does not output data, rather improves the machine learning and bat algorithms. Directory also shows the location and name of the file as also show above.

Will show a separate GUI to show the visualisation of the solutions, again will only activate upon completion of calculation.

Shows the inputs of the file directory in which we want to take preloaded conditions and load forecasts in.

The checkbox below determines If we want to show if we want to show all the data in the iteration of each calculation. As referenced in the systems diagram. File buttons are placed in the corners to make sure that it is out of the way and is to be completed.

Calculates the data and places outputs/ will not show outputs if we are calibrating. Where main procedures are placed and executed. All the buttons are placed at the bottom because it is the las t thing we complete.

This shows the main breakdown of the GUI. The idea is that now I have visualised the inputs and things I will need to take in an output I can develop this a lot quicker and focus more on the main algorithms. However, there is two more GUI aspects I need to cover, the solution GUI and the Visualisation GUI. I will show representations of this below.

## Solution GUI

This GUI is to show all the solutions that have been generated as a result of our data. Ideally this means that the data should all be shown in an easy-to-read format. I will represent my idea below:

Closes the current GUI and goes back to the calculation GUI. The corner of the box shows the common location of getting back.

Diagram

Description automatically generated

The period is selected to show the solutions to that. All the outputs will change to the solution of that selected period. Selection is at the top as it will need to be selected initially and is a requirement for starting.

Shows all the solutions/ demands required from each generator to meet the demand to the most optimal cost. Output boxes are in this format because it is neat and allows for selection at the top period

The properties of the bat that found the global best solution are shown. This may be useful to self-improve at a later point.

The fitness and total cost is shown. These are the most important outputs. They show how much it will cost and if its ideal to the situation. If we are getting a fitness of high magnitude, it is probably wrong.

## Visualisation GUI

The visualisation GUI will be completely different from the others. It may be the case that I use a graphical library to render this as it will be easier to show in a game like format rather than a GUI WinForms format. I will show the design nether the less:

Is the map of powerplants and location of generators. Will affect the distribution of power.

Diagram, schematic

Description automatically generated

Shows the key to which areas are powered so if red it has no power/ is not providing power. If yellow has power and is supplying. I chose the colouring of this because it stands out and yellow is a common colour of light/power.

Shows the key to what is a power plant and what is an area node that needs to be powered. Is like a map of lines. The keys are in the corners as it is out of the way but easy to see. It can be a reference to the map.

Shows the buttons to go back to the calculation screen and the show graph button takes you to the graph GUI which is also shown below. This piece plots the data in a visual format alike this. The buttons are out of the way and easy to access like the keys.

## Graph GUI

The use of the graph GUI will mainly be for the aspect of data analysis. It skips the step of plotting the data in an interpretable format by doing it within the program. All it will do is show expected compared to reality and the improvement on the calculation:

Chart, bar chart

Description automatically generated

Shows the plotted data in the selected format. Would ideally be shown to compare the data of each generator as shown below. May include multiple graph x and y axis. The graph is centralised to make sure it is easy to see and is the primary aspect of the GUI.

Simple back button to go back to the previous GUI. IE: the visualisation GUI

The period and graph type we want to plot is show. Multiple plottable graph types will be available. And the period is shown so the data will change as the period is changed. The selection boxes are at the bottom and easy to access. Can be changed quicker too.

All this being shown I can now jump straight into development upon the introduction of the development phase as this is the first aspect of the program.

# Program Classes

Classes are a critical point to my program. Since C++ allows for diverse use of classes I can choose what sections I need from each. I will try to keep mostly to the aspect of OOP however slipping into procedural may be essential for this program.

Collective Class

Current\_Section

Current\_Period

Total\_Runtime

Max\_Itterations

Next\_Task()

Global\_Is\_Found()

Global\_Best Class

Fitness

Gen[genmax]

Frequency[genmax]

Loudness[genmax]

Pulse[genmax]

Velocity[genmax]

Bat\_Property Class

Frequeny\_min

Frequency\_max

Pulse\_min

Pulse\_max

Amin

Amax

Fitness

Gen[genmax]

Frequency[genmax]

Loudness[genmax]

Pulse[genmax]

Velocity[genmax]

GUI Class

Is\_active

Current\_State

Current\_Menu

Input\_State[7]

Close()

Open.NAME()

Bat\_Collective Class

Bat\_Is\_Active

Current\_Bat

Bat\_Array[maxbats]

Bat\_Initate()

# Flowcharts and Algorithms

For this I will ignore the rendering of the GUI for the most part as it is a tedious part that doesn’t really require any logic and just know phrases of code that are included in the documentation of the GUI rendering library. I will include aspects like if button is press for example as it is key logic to continuing a process. Somethings I may represent as pseudocode and as a flow chart. Especially the larger aspects of the program ie: the bat algorithm. I would take these algorithms with a pinch of salt; they are nowhere near the final aspect and are just there to give me an idea of the structure of the algorithms and the program. Without further or do, here we go…

## Data Interpreter

### Format File into readable form

Diagram

Description automatically generatedFlow diagram: Pseudocode:

Text

Description automatically generated

Starting off super simple. Most of the format functions are prebuilt into libraries and just come under different titles.

### Read Line By Line Expanding Into An Array

Text

Description automatically generatedDiagram

Description automatically generatedFlow diagram: Pseudocode:

Again, a simple one, just reads each line of the file and places it into the array. The flow chart is different to the code because the use of for loops is not really permitted in flow charts.

### File opening algorithm

Diagram

Description automatically generatedFlow diagram: Pseudocode:

Text

Description automatically generated

A small file opening algorithm, would be a part of Openfile() function shown above and referred to above. Is more in depth to detect for errors and open a specific directory.

### View current data and if current is better than previous then set new best

Diagram

Description automatically generatedFlow diagram: Pseudocode:

Text

Description automatically generated

|  |  |
| --- | --- |
| No. | Array[x] |
| 0 | 50 |
| 1 | 21 |
| 2 | 43 |
| 3 | 80 |
| 4 | 10 |

Trace Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Array[x] | Array[x+1] | Temp | Swapped |
| 0 | 50 | 21 | 21 | Yes |
| 1 | 50 | 43 | 43 | Yes |
| 2 |  |  |  | No |
| 3 | 80 | 10 | 80 | Yes |
| 4 | NULL | NULL | NULL | NULL |

Trace Table shows that the algorithm alike a bubble sort.

Just sorts the data into the order of the best solutions, the best will be defined as the last item in the list ie: array[array[].length].

### Improve values and bat properties

Diagram

Description automatically generatedFlow diagram: Pseudocode:

Text

Description automatically generated

All this does is sets the properties of the current bat to the best properties found. This will not be used as much in the final algorithm as it will be fully integrated into the final bat algorithm. But for now, it separates the algorithm into separate pieces that I can use later.

### GUI Page Switching

Text

Description automatically generatedFlow diagram: Pseudocode:

A picture containing text, sign, worktable

Description automatically generated

This just page switches, so when a button is pressed to open a new menu the old menu unrenders (to save resources) and the new one renders. The selected menu is the class name of the current pressed button. Calculate button is stated to run the routeing too at the end. Just to start the calculation.

### Bat Algorithm

Text

Description automatically generatedDiagram

Description automatically generatedFlow diagram: Pseudocode:

Trace Table:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Iteration | Bat | G-A | G-B | G-C | G-D | G-E | Skack | calcost | totalcost |
| 1 | 10 | 10 | 68 | 27 | 2 | 4 | 37 | 5020 | 41983 |
| 2 | 1 | 53 | 49 | 28 | 10 | 12 | 4 | 5288 | **9284** |
| 2 | 2 | 1 | 65 | 77 | 24 | 1 | 20 | 9692 | 29672 |
| 2 | 3 | 41 | 73 | 75 | 13 | 14 | 68 | 10622 | 78554 |
| 2 | 4 | 87 | 5 | 90 | 12 | 8 | 54 | 10991 | 64937 |

Diagram

Description automatically generated

I have taken a segment of a worked solution for my trace table. It shows the output of our algorithm,

and also has all the total costs at those points.

The blue text also show an idea solution that could work

The reality of the algorithm is it is far larger than this. This is just an abstracted version of the actual thing. I will expand the function calculate fitness as the next algorithm in the section.

### Calculating Fitness

Flow diagram: Pseudocode:

Text

Description automatically generatedDiagram

Description automatically generated

|  |  |
| --- | --- |
| P1  (€/MWh) | Q1 (MW) |
| 20 | 50 |
| 5 | 10 |
| 90 | 55 |
| 2 | 25 |
| 4 | 15 |

Trace Table:

|  |  |  |
| --- | --- | --- |
| Gen iteration | Fitness | Guessed demand |
| 1 | 1000 | 50 |
| 2 | 1050 | 60 |
| 3 | 6000 | 115 |
| 4 | 6050 | 130 |
| 5 | 6110 | 145 |

The reliability of this trace table is slim as many calculations are completed per line / iteration.

But the expansion of the fitness does in fact work under the pseudocode I generated.

Calculating the fitness is important to decide which solutions are the most valid and closest to the required solution. It is how we decide if we have a better scenario of where power should be forecasted to. The total forecast is compared to the price of a single gen. If the gen is greater than the demand needed, then we move on another gen band. It will iterate through all the end of the genmax.

## Generator load up time

Text

Description automatically generatedDiagram

Description automatically generatedFlow diagram: Pseudo code:

This double if loop shows that we are detecting to see if a generator is initialised. If it is we can comfortably say we can take away from that generator without having to wait or for it to be boosted in price. Else, we initiate the other option which is to decide if the generator limitations are actually viable and a good one to go for.

### Rendering Nodes and Connections On Map

Text

Description automatically generatedFlow diagram: Pseudo code:

### Diagram Description automatically generated

This is an algorithm I have used to determine if a node has been rendered. It will render the connections and then the nodes on top of them. The function render will be a collection of library functions I will call earlier in the program.

### A\*/Dykstra’s To Determine Shortest Path

I think for this project I have settled on Dykstra’s algorithm as it will be effective at finding the shortest path of a simple tree scenario like this. If I was much larger, I would represent it as a binary 2D matrix and use A\* to transverse it. It is a typical algorithm used in google maps and optimisation algorithms to determine the quickest way of getting to that solution. It might come in use later if I decide to optimise the bat algorithm further.

Text

Description automatically generatedDiagram

Description automatically generated Flow diagram: Pseudo code:

Diagram

Description automatically generated

False

As we can see with this algorithm, I can get a little complex. But all we do essentially is test every path and add a weight to it, provided the path is connected or is a reasonable overall distance it will connect the nodes. It then searches through all the total path lengths and takes the shortest one to the node. It is pretty efficient for this sicario as a I will only be rendering a few connections. But for a later more scaled up prototype of the project this algorithm will need to change or be optimised to deal with thousands of connections and paths.

### Lights Up On Power Intensity

Text

Description automatically generatedDiagram

Description automatically generatedFlow diagram: Pseudo code:

All this code does is takes the percentage of the max power to see how much we light the node up by. Provided the node contains power in the first place. It then renders the colour on the node. Again, this is library functions I can express in flowchart form.

### Output Into File Format

Flow diagram: Pseudo code:

Diagram

Description automatically generatedText

Description automatically generated

This just follows through and writes the solution line by line into the file. More formatting will be added in the final project, however for now this is a good start. And error detection is added to see if we have a valid file name and location.

## Connection Of The Algorithms

For some of this I have decided to include things that I haven’t coded directly ie: the GUI rendering states, but I need to show these as they outline a key aspect in the program. However, flow diagrams nor pseudocode does not justify the code required.

Render GUI

Error Corrections

Load GUI Library

Render GUI

Render Buttons

GUI page switching

Outputs

Decision Engine

File Opening Algorithm 

Format File 

Read line by line into array 

Render Connections and nodes on map

General Purpose

Lights Up On Power Intensity

View current data and if current is better than previous then set new best

File Closing Algorithm 

Graph Determination

Calculate Fitness

File Format Output

Improve values and bat properties

Response Engine

Bat Algorithm

Generator Load Up Time

Dykstra’s to determine shortest path

The path is rather simple, but it does the job in showing where data should go. The simplicity of it will make it much simpler when developing anyway. Each darker box represents a algorithm that will become a module, and then all called into a larger module, ie: Decision engine is the larger module, and Format File is a smaller scenario of a module. Everything in general purpose, will be declared either on a global or wide range scope. This is because they may be called multiple times time throughout the program. And rather than coding it again I will just call it and pass my parameters in instead.

# General Bat Pseudocode

Some of the aspects of the bat algorithm can be quite complex. If we can simplify it with a simplified piece of pseudo code, I can improve my understanding of the algorithm.

## Subscripts

nBats = 1

fmin = 1

fmax = 5

pulsemin = 0

pulsemax = 1

Amin = 0

Azero = 0.9

maxIter = 5

## Start

Initialise Bat (random between min and max):

G-A 56

G-B 43

G-C 11

G-D 32

G-E 8

V-A 0

V-B 0

V-C 0

V-D 0

V-E 0

For this example, load forecast (demand to be met) is 155

A B C D E

G

(generation) 56 43 11 32 8

V (velocity) 0 0 0 0 0

F

(frequency) 1.0 2.0 2.5 3.7 4.3

Pulse (r) 0.2 0.5 0.5 0.6 0.8

Loudness (A) 0.3 0.4 0.25 0.1 0.7

## Step 4

Fitness value = total cost

G-A (generation A guess for bat 1) = 56 >>> cost = (50\*20)+(6\*30) = 1180

G-B (generation B guess for bat 1) = 43 >>> cost = (10\*5)+(33\*40) = 1370

G-C (generation C guess for bat 1) = 11 >>> cost = (11\*90) = 990

G-D (generation D guess for bat 1) = 32 >>> cost = (25\*2)+(7\*12) = 134

G-E (generation E guess for bat 1) = 8 >>> cost = (8\*4) = 32

Slack adder = 5 \* 99999

Total fitness for Bat 1 (iteration 1) = 3706

## Step 5

Say we have 10 bats, and the best fitness (bat 6) = 2540 [current GLOBAL BEST LOCATION], which has generation values of:

G-A 70

G-B 40

G-C 34

G-D 20

G-E 5

## Step 6

Adjust frequency of the bat population (for all 10 bats)

F(i) = random value between Fmin and Fmax

## Step 7

V(i)(t+1) = V(i)(t) + [Fitness(i) – GLOBAL\_BEST] \* F(i)

Where:

· F(i) is calculated at step 6

· GLOBAL\_BEST is the best fitness

· Fitness (fitness for the bat in the current iteration)

· V(i)(t) is zero from initialisation (for iteration 1)

Generation values for each unit and each Bat are updated

GEN(t+1) = GEN(t) + V(i)(t+1)

For our first bat:

GEN(i)(t+1) = 56

43

11

32

8

GEN(i)(t+1) value for unit I = 56 + (56-70)\* F(i) [say 0.5 in this case] = 49

Do for every unit

## Step 8

Pick a random number (rand) – if rand > r(i) [for G-A in Bat 1 for iteration 1, the value is 0.2]

Let’s say that rand = 0.4 > 0.2 >>> goto step 9, else goto step 11

## Step 9

Take all units in the GLOBAL\_BEST solution and create a new local solution:

G-A GLOBAL\_BEST for this unit + (epsilon \* AverageLoudness)

G-B ditto

G-C ditto

G-D ditto

G-E ditto

Where:

Epsilon is a random number between -1 and 1 (-0.2 in this case for example)

AverageLoudness is the average of the loudness for this iteration for all units (0.35)

G-A 70 + (-0.2 \* 0.35) = 69.93

G-B continue as above

## Step 11

Calculate fitness for the rest of either step 7 or step 9 (if step is completed), in a similar way to step 4

## Step 12

Generate a random number = rand (let’s assume 0.25 for this example)

If rand < Loudness(i) (UP TO HERE )

For G-A this is not the case (0.3) but for G-D it is (0.1) and cost of current bat < GLOBAL\_BEST

This becomes the new GLOBAL\_BEST (step 13)

## Step 14

Increase the Pulse r(i)

R(i)(t+1) = R(i)(t) \* (1 - exp(-1 \* gamma\* t))

Where gamma is > 0 and is a constant for all iterations, say 1.5

Decrease the Loudness A(i)

A(i)(t+1) = A(i)(t) \* alpha

Where alpha is between 0 and 1

## Go back to step 4

# Data Structures and Variable Declaration

I am highlighting the key variables and structures I will use in the program here. It can be good to refer to when analysing the code to see where I have used variables and how I have used them. I would also like to make clear my notation of ‘long’ and ‘static’. Long in C++ is a variable that has the capacity to hold more than 32 bits of data. Static is a data structure that cannot be changed at runtime and is a constant sometimes referred to as const.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method | Name | Data Type | Explanation | Justification |
| GUI |  |  |  |  |
| Variable | Is\_Active | Boolean | Determines if the current GUI menu has been rendered and is active | To see which menus are active and to make a on/off switch for each menu |
| Variable | Current\_State | Integer | Will determine if there was an error or if launch was successful | To detect errors on runtime and if we are switching/executing GUI’s correctly |
| Variable | Current\_Menu | String | Determines what menu we are currently on. | Is needed to decide where to navigate to and what options are open to the GUI to switch to |
| Array | Input\_State[] | Integer | Takes the array of buttons in the current GUI and sets a value ie: 1 or 0, 1 meaning pressed, 0 meaning not pressed | Is needed to determine if buttons are pressed or not. If they are, they will be activated due to the change in input state. |
| Procedure | Close() | N/A | Closes the GUI when the exit button is pressed or a critical error halts execution of the program | Will be needed to make sure the GUI closes successfully without causing any dormant programs and GUI render not closing. |
| Procedure | Open.NAME() | N/A | NAME is a super class of GUI’s available to open, but all it does is open the menu that we want to open. | Is needed to unrender the current menu and load a new one. Creates a smooth transition between GUI’s. |
| Decision Engine |  |  |  |  |
| Variable | Filename | String | Gives the input of the file name. Is taken from the input of the file location | Used to locate the file we want in a particular directory |
| Variable | Filelocation | String | Represents a file location that has been interpreted by the file\_loc() procedure | Is needed to find the location so we can pass it into interpretable and smaller forms for later algorithms |
| Variable | DIR | String | Shows the physical location of the file in the explorer according to the operating system. Is the directory of the folder the file is in. | Needed to input the location into the File\_Loc() array, and is the physical location on the computer of the folder. |
| Procedure | File\_Loc() | N/A | Is an array used to convert the directory of the folder into a interpretable way that the program can understand. | Will used as a pointer for later algorithms and will produce an output that is essential to opening a file. |
| Variable | Fmin | Float | Is the minimum frequency a bat can obtain during all iterations | Used to set a limit to the lower bound and not generate absurd numbers that would be unrealistic |
| Variable | Fmax | Float | Is the maximum frequency that a bat can obtain during iterations | Will be used as the upper bound to again make sure numbers are realistic to the scenario and not becoming too high/low |
| Variable | Pulsemin | Float | Is the min pulse rate a bat can achieve. | Again is to set limits to reduce the number generation and make sure numbers don’t tend to a single number out of the range. |
| Variable | Pulsemax | Float | Sets the upper limit to the generation numbers for pulse again of a bat | Makes sure the numbers again don’t go too high/ out of realistic range. |
| Variable | Amin | Float | Is the lower limit of the loudness. | Is usually to make sure that the loudness of a bat stays > 0 and doesn’t dip into negative numbers |
| Variable | Azero | Float | Is the upper limit to the loudness | Makes sure that the loudness doesn’t exceed this again. |
| Variable | Fitness | Integer | Is the validity of the current bat iteration and how ‘good’ the results are. | Is used to make sure that results that have an overall better fitness will be accepted as they are a better solution. |
| Variable | Genmax | Integer | Is the maximum number of generators per plant | Used to set upper limits to the size of arrays and how many times to iterate to. |
| Variable | Gennum | Integer | Is the number of suppliers that are available | Used to set an upper limit to the width of the price and quantity arrays as to how many times we should look for items |
| Procedure | CalcFitness() | N/A | Used to calculate the fitness of a scenario and give a value to a bat of fitness. | Is used to convert all the gen inputs, quantities and prices into a fitness and determine if it meets the demand required. |
| Array | QTY[][] | Integer | An array to store the max quantity we can take from each generator, is taken from the inputs and scenario data set. | Makes sure that we are obeying a limit to the supply we can take from that particular generator |
| Array | PRICE[][] | Integer | Is the price per unit of electricity that we can have per generator/ is directly related to the quantity | Used to determine the price (set by a supplier) of a generators demand wanted. |
| Array | Period[] | Integer | Selects the current period of demand we are calculating for. | Splits the time of demand up into equal segments called periods. |
| Array | Gen[] | Integer | Is all generators and what each bat has in terms of demand per generator | Is used to store the demand from each generator that a bat a guessed/tried |
| Array | Frequency[] | Long Float | Is the frequency of that bat per generator that has been generated | Used to calculate new values and set properties to that generator on that bat. |
| Array | Pulse[] | Long Float | Used to set the pulse of the bat per generator which has been determined | Is used for later calculation and assigns the property to that bat on that gen. |
| Array | Loudness[] | Long Float | Is the loudness per bat of the generator selected | Used again for calculation on that bat on a generator in the bat algorithm |
| Array | Velocity[] | Long Float | Is the velocity per bat of the generator selected | Finally, is used to assign a velocity property to that bat on each generator |
| Array | Best\_Gen[] | Integer | Takes the best generation the bat has found and sets it in an array | Is used to compare with all the later values of global best and current bests. |
| Procedure | Bat\_Initate() | N/A | Initiates all the bat properties to random values between the max and min for each bat. | Makes sure that the numbers start of non-null and are between the values we need them to be. |
| Response Engine |  |  |  |  |
| Variable | nBats | Integer | Is the number of bats in total that we will initiate properties to | Is used to set the number of bats we will need to initiate. |
| Variable | Itterations | Integer | The total number of iterations per bat we complete. | Determines the max amount of times we can ‘self-improve’ the algorithm |
| Array | Batcollection[] | BAT (class) | Is the declared array in which we will assign all the properties of the bat to. Each item in the array represents a single bat | Makes sure that we can call each bat easily In a loop like format. |
| Variable | Epsilon | Static Float | A random value generated to give random numbers between the limits -1 and 1. | Is used to calculate the new best solution. |
| Variable | Gamma | Static Float | Random value that is > 0. | Is used making a new pulse for the next iteration |
| Variable | Alpha | Static Float | Another random values that is between 0 and 1. | Used to calculate the new loudness of a bat. |
| Procedure | Bat\_General() | N/A | Is the launching of the general bat algorithm and everything calculating in the main segment is stored here | Makes sure that the bat algorithm is stored modularly so we can iterate outside of the procedure |
| Procedure | Bat\_Improve() | N/A | Self improves all the initial bat min, max ect values for the next period we want to iterate to. | Makes sure that all the values are optimised and are better to provide better results. |
| Procedure | Bat\_account() | N/A | Uses the limitations of a generator to take in account a result or if the fitness is below the demand ect… | Is used to make sure our results are realistic or reliable. Makes sure the bat algorithm converges into the correct solution that meets the demand. |
| Global Best |  |  |  |  |
| Variable | gb\_Fitness | Integer | Sets the best ever found fitness a place in memory to be stored | Makes sure that we are storing our best solution so we can print it later. |
| Variable | Foundby | Integer | Shows which bat found the best solution | Just for debugging and allows for output to find possible trends |
| Array | Gb\_gen[] | Integer | Takes the values of each generator from the global best as our solutions. | Makes sure that we have the correct solutions, and these will be our final solutions |
| Array | Gb\_F[] | Long Float | Takes the values of the frequency of the bat at the time finding the solution. | For debugging and makes sure we are getting correct values of frequency |
| Array | Gb\_A[] | Long Float | Takes the values of the loudness of the bat at the time finding the solution. | For debugging and makes sure we are getting correct values of loudness |
| Array | Gb\_R[] | Long Float | Takes the values of the pulse of the bat at the time finding the solution. | For debugging and makes sure we are getting correct values of pulse |
| Array | Gb\_V[] | Long Float | Takes the values of the velocity of the bat at the time finding the solution. | For debugging and makes sure we are getting correct values of velocity |
| Outputs |  |  |  |  |
| Array | Path[] | PATH (class) | Is the properties of the path we want to flow for the map | Sets all the properties of the path class to the total number of paths available |
| Variable | Desired | Integer | What node we need to get to in that scenario | Is used as a reference to see where we need to get to so we can find if we are at a location |
| Variable | Weight | Float | Is the weight of a path and how much it affects the path ie: how far it has to go. | Makes sure that we are taking into account the fact that more may be longer than others. |
| Variable | Nextright | Integer | Is the node to the next right that we will transvers to | Used as a pointer to reference and needed to transverse into the next node. |
| Variable | Starting | Integer | Is the initial node we want to start at | Makes sure we have a starting point for the algorithm |
| Array | Node[] | Integer | Is the properties of the node we currently lie on eg: if it is a power station/destination | Makes sure that power is radiating from the correct place later in the program |
| Procedure | Render() | N/A | Renders the nodes that connect the connections | Makes sure that nodes render on calling at location passed in. |
| Procedure | RenderConnection() | N/A | Renders the connections between nodes | Makes sure that connections render from the centre of a node to the other connecting node |
| Procedure | Transverse() | N/A | Transverses into the location passed in, adds the value of weight too. | Makes sure that the current pointer is set to that of the transversed node. |

# Test Data

I will use the test data in this project to determine if my algorithms are producing the correct results. It ensures that I am producing the correct and desired result. For this my stakeholder has produced some documentation on the correct values I should be getting. This I will use as my test data for the bat algorithm. However, there is other data I will use to test the GUI switching, rendering and outputs. I will need to test my data because all the data that I will produce will have to be valid and produce valid and no valid errors. I will need to expect if an output is valid or not as I know what to improve and account for in the development of the project.

## GUI Testing

### Buttons

|  |  |
| --- | --- |
| Test Data | Type |
| Calculate Button Pressed | Valid |
| Exit Button Pressed | Valid |

## Files

|  |  |
| --- | --- |
| Test Data | Type |
| File loaded without directory | Invalid |
| File loaded with directory | Valid |
| File loaded with corrupt file | Invalid |
| File loaded with no data | Invalid |
| File loaded in format data | Valid |

### Inputs

|  |  |
| --- | --- |
| Test Data | Type |
| High Temp (eg: 99999) | Invalid |
| Mid Temp (eg: 30) | Valid |
| Large Fluctuation In Temp | Invalid |
| Small Mean Temp | Valid |
| Conditions N/A | Invalid |
| Current time > 24hours | Invalid |

## GUI Swapping

|  |  |
| --- | --- |
| Test Data | Type |
| Show solutions is pressed | Valid |
| Calculation is in process whilst solutions is pressed | Invalid |
| No calculation has occurred and pressed visualisation | Invalid |
| Calculation is completed and visualisation is pressed | Valid |
|  |  |

## Bat Algorithm

For this section I will not use a conventional method of testing like above. Since I know some of the solutions to the exact values. I will use this as reference to test against. I will add all the data below. The correct solutions are provided to compare. This is needed to make sure that the algorithm is producing the correct response. In an ideal scenario all periods will produce the correct result. The dataset is used to make sure that I have obtained the correct solution. Since I know all of the expected data I know the outputs too.

|  |
| --- |
| **Introduction** |
| For this example, there are 48 half-hour time periods covering a whole day |
|  |
| **Demand (Load) Forecast** |
| The Demand (or Load) Forecast is provided for each time period in the day |
| LF is in Megawatts (MW) |
| Data is found in the input\_data sheet |
|  |
| **Generation Units** |
| Generation Units can be considered to be power stations |
| Each Generation Unit has a limit in terms of generation |
| Prices for generation units are provided as Price-Quantity Pairs |
| Each Price-Quantity Pair reflects the price for each MW from this quantity back to the previous quantity |
| Prices must be non-decreasing (i.e. P1<=P2<=P3 ….) |
| Quantities must be increasing (i.e. Q1<Q2) |
| Price Px applies from quantity Qx-1 (exclusive) to Qx (inclusive) |
|  |
| **Merit Order** |
| This sheet provides a stack by generation and cost, in ascending order of cost |
| The cost column in this sheet is the cumulative cost of 1MW values to that MW level |
|  |
| **Correct Answers** |
| Contains the "correct" cost for each time period and the total cost for the day |
|  |
| **This is the SIMPLEST version of the problem** |
| There are lots of additional elements of the problem to be added once this initial version is working |
| >> Price-Quantity curves changing by time period |
| >> Startup Costs and Times |
| >> Inertia Optimisation |
| >> Transmission Constraints |
| >> Generation Unit Technical Constraints (e.g. ramping) |
| >> Generation Reserve Requirements |
| >> Machine learning for tuning |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Unit | P1  (€/MWh) | Q1 (MW) | P2 (€/MWh) | Q2 (MW) |
| G-A | 20 | 50 | 30 | 100 |
| G-B | 5 | 10 | 40 | 85 |
| G-C | 90 | 55 | 110 | 95 |
| G-D | 2 | 25 | 12 | 34 |
| G-E | 4 | 15 |  |  |
| gen\_slack | 99999 | 99999 |  |  |

|  |  |  |
| --- | --- | --- |
| Time Period | Test LF (MW) | "Correct" Cost |
| 1 | 296 | 5842 |
| 2 | 229 | 2812 |
| 3 | 132 | 997 |
| 4 | 147 | 1222 |
| 5 | 150 | 1267 |
| 6 | 224 | 2712 |
| 7 | 203 | 2292 |
| 8 | 101 | 562 |
| 9 | 288 | 5402 |
| 10 | 296 | 5842 |
| 11 | 185 | 1932 |
| 12 | 276 | 4852 |
| 13 | 255 | 3907 |
| 14 | 201 | 2252 |
| 15 | 188 | 1992 |
| 16 | 207 | 2372 |
| 17 | 235 | 3007 |
| 18 | 191 | 2052 |
| 19 | 226 | 2752 |
| 20 | 191 | 2052 |
| 21 | 169 | 1612 |
| 22 | 227 | 2772 |
| 23 | 167 | 1572 |
| 24 | 135 | 1042 |
| 25 | 303 | 6227 |
| 26 | 209 | 2412 |
| 27 | 232 | 2872 |
| 28 | 217 | 2572 |
| 29 | 145 | 1192 |
| 30 | 278 | 4942 |
| 31 | 254 | 3862 |
| 32 | 230 | 2832 |
| 33 | 128 | 937 |
| 34 | 204 | 2312 |
| 35 | 165 | 1532 |
| 36 | 119 | 802 |
| 37 | 310 | 6612 |
| 38 | 275 | 4807 |
| 39 | 305 | 6337 |
| 40 | 203 | 2292 |
| 41 | 313 | 6777 |
| 42 | 308 | 6502 |
| 43 | 277 | 4897 |
| 44 | 213 | 2492 |
| 45 | 137 | 1072 |
| 46 | 232 | 2872 |
| 47 | 130 | 967 |
| 48 | 310 | 6612 |

The correct total cost is shown, this is the most optimal cost given the table above

This is the required demand at that time, and using the table above you can show the optimal cost.

This is the section of period, it is our control for this table

# Datasets For Deep Learning

For this section I will attach the example that I have been provided and also a few of the data sets I have to use machine learning and decide what would become a better response/ quicker response to a solution. Needed to ensure the development of coefficients in the machine learning algorithm.

Table

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidenceTable

Description automatically generated

# Acceptance Testing

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Requirements | Input | Expected Outcome |
| 1 | When program is executed, GUI is presented to user | Valid: GUI entered with buffer switching  Invalid: GUI entered without buffer switching | Valid: GUI opens and renders the screen and keeps on a steady screen  Invalid: The GUI freezes on the front buffer and does not switch back to the next buffer |
| 2 | Buttons and input boxes are rendered | Valid: Buttons are rendered in range of screen  Invalid: Buttons have undeclared render location | Valid: Buttons show in their correct position with defined location  Invalid: Buttons overlap at pos(0,0) due to undefined position. |
| 3 | Buttons and input boxes are interactable | Valid: Click the button and place characters into the input boxes  Invalid: Button doesn’t click, and input boxes won’t interact. | Valid: The button will return 1 and input boxes allow characters to be written.  Invalid: Button will most likely freeze, and input boxes will not allow input |
| 4 | Buttons take inputs from user | Valid: Place 2 into all inputs and select correct conditions  Invalid: Place “Hello world” into all boxes | Valid: Will return a value and take correct data.  Invalid: Will return critical error of string to integer conversion. |
| 5 | Redirect GUI button open new GUI | Valid: Click button and wait  Invalid: Press the button multiple times whilst loading | Valid: GUI opens and closes old GUI  Invalid: GUI opens multiple times and overlaps data. Old GUI may return critical error. |
| 6 | When exit button pressed, the GUI closes | Valid: Pressed a single time  Invalid: Pressed multiple times | Valid: Program Closes out after memory has been cleared  Invalid: The program closes with memory leaks as calculation is still stored in memory. |
| 7 | File can be opened, and directory is correctly shown | Valid: File is selected of correct format  Invalid: File is selected of invalid format | Valid: Opens file directory and places file into the algorithm ready to execute  Invalid: The file opens and causes an error on execution as file cannot be read. |
| 8 | File does not respond with error | Valid: Correct directory is entered  Invalid: Incorrect directory is entered | Valid: The file opens without no error  Invalid: The directory cannot be accessed and returns and error |
| 9 | Data set is interpreted into algorithms | Valid: Data set is of valid format  Invalid: Data set is of invalid undetermined format | Valid: The data set is passed into the algorithms  Invalid: The data set returns an error/ or passes nonsense values into the algorithm |
| 10 | Dropdown combo box presents all the conditions and periods. | Valid: Dropbox selected value is in the combo box  Invalid: The selected item is NULL | Valid: The calculation will proceed  Invalid: The calculation will return error with no response initiated. |
| 11 | Calculate Button Creates response on console screen | Valid: The boxes are filled out correctly  Invalid: Some is incorrect | Valid: The responses print out onto the console screen  Invalid: The response is invalid or nothing is represented |
| 12 | A Response Is Box Missing Data | Valid: The boxes are all filled with correct data  Invalid: Some boxes have correct data, some have NULL data | Valid: A response is calculated correctly  Invalid: Some initial responses are null and some don’t return, / possible error messages from logical errors at runtime. |
| 13 | Periods are created in output GUI | Valid: Periods are placed in data set  Invalid: Periods are not placed in data set | Valid: Periods show in combo boxes and calculates all periods  Invalid: Calculates a single period and combo boxes show no periods except the initial |
| 14 | Bat algorithm creates correct response | N/A | Response for test data is the same as the model correct answers |
| 15 | Datasets are referenced to, to create improved coefficients. | Invalid: Dataset is not referenced to | Invalid: The dataset provides error messages and doesn’t improve coefficients. |
| 16 | Map renders on GUI opening | Valid: Calculation is complete  Invalid: Calculation is not complete | Valid: The map will render with data  Invalid: The map will not render as no data is provided |
| 17 | Nodes render | N/A | Valid: Nodes render  Invalid: Nodes don’t render |
| 18 | Connections render | N/A | Valid: Connections render  Invalid: Connections don’t render |
| 19 | Key renders | N/A | Valid: key renders  Invalid: key doesn’t render |
| 20 | Correct colours render | N/A | Valid: colours render  Invalid: colours don’t render |
| 21 | Buttons to go back and show graph render | Invalid: No data is provided to calculate | Invalid: No graph is shown as data cannot be represented |
| 22 | On button click for graph, GUI is closed, and graph GUI is rendered | N/A | The GUI renders and unrenders in order of each other alike above. |
| 23 | Renders the correct graph type with correct data points | Valid: The input graph type is bar  Invalid: The input graph type is NULL | Valid: graph is shown and rendered  Invalid: graph is not shown, and no data is plotted |
| 24 | The correct period is show and renders each sequentially | N/A | Period Shows in order with graph rendering for each |

Development

# The Initial Preparations

This section is fairly self-explanatory. I will bring you though my development phase with each step I took to develop and release project. It is a fair task at hand and my previous sections (‘Analysis’ and ‘Design’) will help me to program this development formally and efficiently, with insurance that I keep my work organized and modular. This initial part of the project, it will outline everything I needed to install, and research before even typing a line of code.

## GUI Preparations (Day 1)

I have opted to use WinForms within the .net framework and SDK (software development kit) of C++. This means that provided the user has a windows operating system, the .net framework is always included and thus no further installation is required other than that of the program. The [Microsoft official](https://docs.microsoft.com/en-us/windows/win32/learnwin32/your-first-windows-program) website has some amazing documentation on how to get started and creating windows, boxes, buttons, ect… Graphical user interface, application

Description automatically generated

However, I already know how to do all of thus I will not bore you with all the details of these basic pages. I will however show you the additional dependencies that Microsoft require you to install to develop programs with the .Net framework. This just consists of a simple google search of ‘.Net SDK’ and you will get the first option which is also the official Microsoft website where you will be greeted with the following page.

Graphical user interface, text, application, email

Description automatically generated

It shows the different SDK’s that are available to download, however it is crucial you download the correct one, as the additional libraries that will need to be included in the project will differ from different CPU’s as the instruction set is different. I am running on a x64 CPU so I will download that one. However, the next thing to consider is do you want the ‘Long-Term Support’ (LTS) Version or the Preview for the latest version. For this project I will use the LTS because it is likely my client will need to use this with little maintenance, it just means that the initial release of the project will be compatible with most computers for the foreseeable future. And finally, I will select the ‘Visual Studio 2022 SDK’ column. This is because firstly, my IDE I will be using is Visual Studio, and secondly it is easier to set up with their official IDE rather than an outsourced one that could take hours to import basic libraries that Visual Studio import by default.

Next…I will jump into the IDE and start preparing that for development. I find it is always nice to get all the header and Cpp files in a nice, organised manor so that we can jump straight into development. The other thing I will need to do too is set all of the default runtime includes and libraries. This just makes it so I’m not wondering why I’m getting errors suggesting functions don’t exist.

Graphical user interface, application

Description automatically generated

So, this is the start of my project. This is completely blank and has absolutely nothing in it. I will first jump to including the libraries and dependencies that we installed with the .Net SDK.

Graphical user interface, text, application, email

Description automatically generated

v

This is the properties tab within the IDE for the project. I have opted to delete some extensions on runtime. All this will do is clean the cache and RAM of all of the extensions listed so that we don’t clog up our memory with random stuff we aren’t using anymore. It is also a good idea to set the character set to Unicode, this is because when rendering the items, the character set of the compiler will need to be more complex to allow for any unexpected inputs of characters. So thus, I will use this.

A screenshot of a computer

Description automatically generated with medium confidence

This is the organisation system opted to use. I have used this for many large projects before and the system seems to work very well. The header files includes all of the custom made includes that I made for my project (which is where 90% of the main code is), and the Source Files, which are the entry points for the compiler, and what the compiler actually ends up compiling. Resource files just contain all the images or sounds for example, that I might wish to use. The filters in the ‘Header Files’ filter help me to deduce what is what; ‘ENGINES’ control a large portion of a project to complete a particular task. ‘BANKS’ hold all the includes in a single place or store large amounts of global and local variables, this means we can change constants easily without having to scavenge through the complex depths of the code looking for a single variables. This also leads to quick debugging too if I need to test different values. The next logical thing to do is state all the files I created in the filters. So within ‘BANKS’; ‘Globals.h’ holds all the global variables that will be used across the program;’Halfincludes.h’ Includes all the external dependencies in a way that won’t overlap with entry points, this is because if

two Cpp files compile with the same variables the compiler confuses them and mixes them; ‘Solutionbank.h’ stores all the information and details on the solutions to the problem, Is useful to make sure that code isn’t repeated throughout the project, and it is stored and returned to that .h file. Now, ‘ENGINES’; ‘BAT.h’, this is where the majority of the machine learning, multithreading, Bat algorithm all takes place. I decided to place this in a single .h due to the fact I will not need to worry about including loads of external .h files and if decide later in another iteration of the .h to create this into a library I can do so more easily; ‘GraphPlot.h’ This is where I do all of the graph plotting calculations and place this onto the GUI; ‘ShortestPath.h’ All of the main calculations for the visual representation will be placed in here, everything from the shortest path of a generator to a unit is stored here and then the lighting of connections ect;’SolutionOutFormat.h’ is where all of the file writing operations will be performed for the main solution. All debugging writing will be performed in the main algorithm for simplicity. The rest of it is where all the GUI’s that I’ve included with my program.

## Solution and Main Form Setup (Day 2)

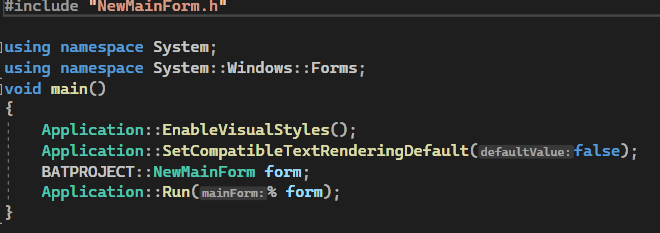
Firstly, I need to make sure I can initiate a GUI and make sure that I can get consistent closing, minimising, and functional testing, i.e.: buttons work and execute functions and reference to other classes. I will test with a blank form with a few buttons and things first. Then I will create a table to declare what is working as of this moment.

This is where stuff can get a little creative. I get to design and assign locations to buttons, include events, and set up the main entry point (through MainForm.Cpp). This initially will be very difficult; this is because the method making sure that all the GUI’s are linked throughout the program so we can open and close seamlessly. Firstly, creating a form, and design view…

For this I created a header file to ensure that all my code in this section is neat and out of the way of the main entry point. The next thing I made was a .cpp for each GUI, and included the default code to ensure that when executed the program references to the mainform to open, this makes sure that a form cannot be opened apart from each other at different times (more on that later).

The .h file for this .cpp is called so the main function (void main) knows the definitions and what to define inside the GUI.

From what we referenced from including earlier we are now just importing the scope and classes included with the namespaces listed. This will give us access to the functions needed below.



The first line allows the WinForms library to realise that we want to attempt to open an instance of a GUI. Within the application scope is from WinForms made namespaces.

Next line if for GDI+ based graphical interfaces. We use this to declare how we want to render characters on the GUI.

The ‘BATPROJECT’ namespace is one I created in all the GUI’s to make sure that all have the same properties, and we can open them by just calling that GUI under that scope and the form name and calling a reference name / group of pointers.

Lastly the line at the bottom runs the program by opening it and passing the pointer to the properties of MainForm. If I were to change line 3 to BATPROJECT::SolutionFrom form; it would open the solution form instead, provided we included the solution form in the same area as NewMainForm.h or newmainform.cpp.

Now for the .h file. This defines all the positions of text boxes, styles, colours, text, events, ect…. This ensures that what we want to be rendered is actually rendered and also applies functionality to the item. I will annotate below; however, it is a lot of code. This is fairly straightforward and there is little room for errors, this only really took me about an hour, with it managing to run first try. Despite this I will show my progress of the GUI with a diagram of code and as mentioned above a table.

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Now I will explain the things I was trying to test with this iteration of the GUI. This should make it clear how I made the GUI and what I did to ensure that we wouldn’t run into problems created now, further into the program. I hope this stage will ensure a smooth transition from my light GUI shown above to what is shown later on with all the details.

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Type | Pass/Fail | Improvement |
| GUI | GUI loads and runs without fail to a blank screen | Pass (Shown by Working!!! Message box that runs on execution) | N/A |
| GUI | Label/ Button renders | Pass (Shown on GUI by the button and Working!! Text in the label) | N/A |
| Debug Config | Make sure that we are getting any warnings with the GUI. This ensures that we know where limitations may occur | Pass (All warnings are displayed when entered a simple diagnostic eg: conversion of Boolean to integer) | N/A |
| SDK | Make sure all SDK’s are loading all the prerequisite requirements of the WinForms library. | Pass (Evidence) : | N/A |

Text

Description automatically generated

Public protected and private parts are used throughout the program. Protected will be used such that another form can access the items. Public so the whole form can, and private so nothing can access it accept the class its in.

The main classed used to distinguish the MainForm from other forms. And also allows for all textboxes, sliders etc to be made and rendered.

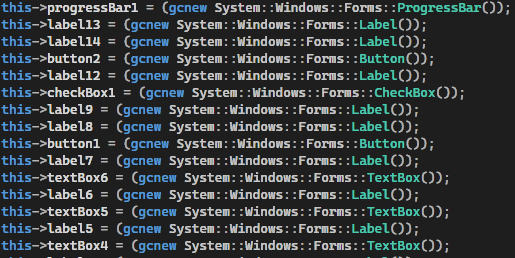
The next bit is for placing this class into use, as WinForms does not know that we want to use it until we call initializecompnent(). The rest is as it says in the commenting.

This is the main area for declaring everything we will need to make winforms items. Specifically, refer to the last line, which references to ‘Drawing2D’. This is what I will use for all of my graphical rendering of lines, graphs points, ect.

Text

Description automatically generated

I won’t annotate this bit and bore you. All this section does is allow for items to be declared. For example, if we want a label on the form then we will need to declare it here. I have declared one of these for every component in this form. I also made sure it renders the correct thing. Ie: if I want a Button, I will need to use the reference Button^ (as a pointer). And this is private to make sure if a component is named the same in another form it will not alter that one too. It is just a variable type of label/ button ect that contains all the properties that it needs to be that.



This is fairly much the same as before, except now we are just making the objects into manipulatable items with all the WinForms functions and definitions. Not much to say here, just note that if I assigned a progress bar a label property it would crash the program and cause all sorts of weird errors. And finally ‘gcnew’ creates a new instance of that object ie: can be used to redefine the properties in this scenario. This repeats basically forever too.

Text

Description automatically generated

This is all the main properties of a button for example, if we wanted it to be bigger, we could draw it to be bigger (line 5). Or we can assign a new font by replacing the quotes, and this is all referenced in the drawing namespace we used earlier. This allows for character rendering. Or we could change the text.

The main events handles that we want to happen under this object are referenced to here, we refence to that if it is clicked that we activate/ call the button2\_click function that is referenced through from the main form class.

This again just repeats for every item, this took most of the time through my day and also making sure that when I ran it, that everything would align with where I wanted it to be. My plan is to compare this with my main GUI, after a few runs and deciding where everything should be, my first iteration was this rather basic and I also faced the problem that if I wanted to make a conversion visualisation in the main program, I would need to make sure that all of my items where in the main form scope. That is the one limitation of my code, is that I cannot create variables that travel across all GUI’s. It is fixed to a single GUI. This makes a hassle of making functions that I can use multiple times across programs. I have a solution; however, I will mention this later on. The first iteration screen shot is attached below.

Diagram

Description automatically generatedGraphical user interface

Description automatically generated

If this is compared to the original design, it is very similar. However, it highlights a few issues that I will face when developing. For example, I have no method of debugging to a file for ease of access. Debugging to a file is very intense with large amounts of this occurring within a short span of time. So, to counter this is will need a toggle to debug with all the details or none at all. Next of all, I have decided to allow the user to just input the demands into source code, this is susceptible to change later on, but this will change for a box that will take all the demands for all 48 hours at once. Now I need a way to enter gen values for each gen. I will probably replace the forecast hours for this. And ‘self-improvement’ via the coefficients needs to be accessible and readable. I will attach change this too. Finally, I would like something that will show the convergence of the algorithm. Hopefully this will show how the algorithm is working and how the bats are moving, this will later include a lot of vector geometry in 2D.

Graphical user interface

Description automatically generated

This I where I’ve added all the new inputs for gen info like costs and max generation. And this is the new box for the demand. This just makes it a lot easier when released to input the data required. More boxes may appear as I develop.

The box I created here is where we will draw graphical dots to represent the convergence of the bats as they get closer to the solution. This again makes it easier to debug in some cases. And if not else creates a neat little visual gateway the algorithm I’m developing. The labels will show information about the current iteration of the algorithm too.

This is a toggle button that should take in the input of the user to if they would like to print out all the debug information or not. And it also gives us the option to specify a bat that we would like to track in the algorithm. And I’ve added a number of calibrations we can use to get better coefficients. This should ideally limit top how long it takes.

This will allow the uses to see all the best coefficients used. Will also be used as an input. The number of bats we want to use can also be specified here. This will hopefully make the algorithm more adjustable to a problem.

This is the new design that I came up with, I have annotated it to show the changes I made and mentioned above. Ideally this will make my life a lot easier when coding the main modular algorithms. This has also introduced a few new ideas of going about solving my problem.

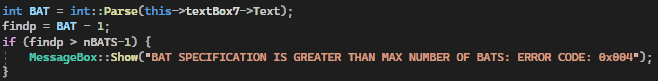
### Testing The Main Form Functions

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Type | Pass/Fail | Improvements |
| GUI | Renders | Pass (As shown) | N/A |
| GUI | Switches forms after button press | Pass | N/A |
| GUI | Takes valid inputs and looks for invalid inputs | Fail | Crashes on input of invalid text type. Need detection of the text type inputted |
| GUI | Progress bar counts | Pass | N/A |
| GUI | Has limitations to when a button can be pressed before calculation | Pass | N/A |

We can see that one of the test failed. I will need to fix this before continuing. I will fix this by making a few if statements to ensure that It will check before using the input. This is shown below:

This just switches whatever was inputted to what it needed. This is the same but similar for other types of input. This was my work around.

This is just used to exert an exception and make sure that the user is entering ‘dodgy ground’. It just checks inputs.



After I did this modification, the failed test now passed, this means the bug is fixed. I will not need to worry about false inputs for now unless I would like to refine this further. This is repeated for many input values, and it applies on the event of a change in text input.

Next, let’s talk about how I want to output the solutions I get. I have been introduced to one of the key errors with multiple GUI’s in a single program with WinForms though, that Is that the global variables I initiated will not be accessible to all forms, just a single one. Therefore, I cannot carry values to different forms. I have an idea for later though. And this should fix this with a few compromises though. I will tackle this later today and make sure I can refine this before debugging anything major.

Graphical user interface

Description automatically generated

This is the back button to go back to the main form, this means that we can just navigate back and to the main and solution form.

This shows all the generator values to place on how much each generator should be outputting in Megawatt hours to provide that cost/fitness. This should make it easier to visually see where the generators need to be placed.

This prints all the bat properties for that particular period of which found the best solution. This will be good for debugging, but also finding some key patterns between bats and periods.

This Is the combo box that will list all of the available periods and will assign the associated values to that value of the combo box. This ensures that I don’t have to create loads of boxes to print a few periods that may not even contain values.

This is the main bits were looking for. This shows the total cost, which is how much the provider will need to pay for that output of gens and the fitness is the suitability of that period. The higher the worse it is. You want a fitness approximately equal to the total cost.

This Is my initial design, and the code is very much like the main form, except I will attach the base of the code to distinguish the differences in defining the GUI’s. But compared to my initial design in the design section of this documentation, it is very similar to what I wanted to produce. This is a good indication that my initial ideas where a good direction to where I’m developing now.

Diagram

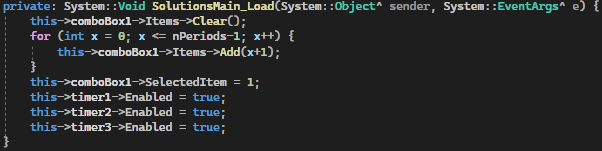
Description automatically generated

The code that I will show, will show the main aspect of declaring the new ‘SolutionForm’. This file only contains a .h in it’s series. This means that It just links into the main form. No .cpp is required to initiate this GUI. All we do instead is make the GUI to the from on the dialog for the BATPROJECT namespace and then just get it to make that the new entry point. This will just place the old mainform onto a stack with all the information it held, which is the reason why I can’t have the same global variable on the same form as I would be initialising it twice, in which c++ does not allow. I will also attach a diagram of the BATPROJECT namespace I created in terms of class diagrams and accessibility to other parts of the program.

Next the code for the Solution Form, this bit gets quite complicated so I will try to explain it at my best. What I did essentially is prepared to read directly from a file. And this is read on a timer so that it will execute every tick of the timer. This runs about every 10ms or so. This allows for the solutions to be consistently updated, and when we change the index, it immediately changes the solutions.

This section of code adds all of the periods to the combo box. This allows us to make sure that we create an index for each of the periods in terms of the position of the combo box. Clear also just makes sure that we get no overlapping and duplication of indexes.

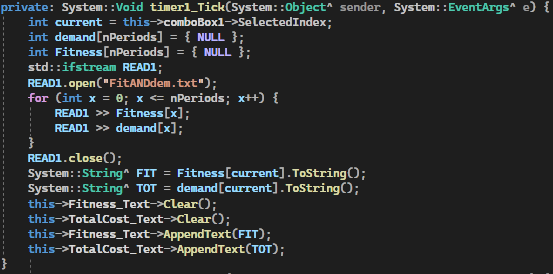
This enables all the timers the moment we load up the form. This also ensures the case that if we switch forms, the timers stop and run after it opens again. This allows for more efficient use of resources.



The next bit of code I made was the timers and using the reading function, that I will make later, to read the solutions. I made this first before any heavy logic code so I could debug my outputs easily. The files I read from is the files I will create to output debug to. I used 3 separate timers to isolate each column. This is because too much execution of a single 10ms timer will overload the stack on a single core and cause it to either crash, run really slow, or finally mess up a lot of answers. So, I will use this to make sure that I get consistent reading from each execution of the timer.

### Timer 1

All this does it take a variable to set it to the index that will reference to the place in the file. Also allows use to know how far we need to go down to get to the index.



I declared a file read object called READ1 that contains all the properties of files. And then opened a file from the resources file called ‘FItANDdem.txt’. This then iterates through the file reading every 2 lines for Fitness and demand inputs. I called it ‘FitANDdem.txt’ because it stores fitness and demand.

This sets an array to the fitness or the total. This can then be used to set the textboxes to display our solutions. This is used as a System string to make sure it is consistent with the input of a label/ textbox.

### Timer 2

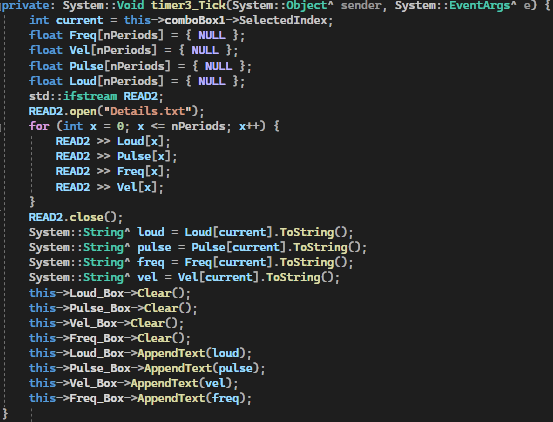


Finally we just set all of the text in the boxes to g1,g2,g3,g4 ect and make sure it is clear beforehand.

This will read through the HELP.txt file. It was initially called this because it was to debug and help me get over the reason why I could read multiple items at a single time. I now remains as that as it was too hard to change. After this it just reads through all the gens and outputs them to our second file ie ‘help.txt’. It then sets all of the current items g1,g2,g3,g4,g5 to all of the places of the current gens for input to a textbox.

This reads from the solutions main text file to get all the gen values. This is referenced by the index alike before. This gives it a position. And it switches every 5 or so (which is the set value of gens). This is placed into a 2D array of period against gen number. The double for loop iterates through this.

### Timer 3



We then read in the loudness pulse etc for each period of the current into the new values and then clear and set them equal to the textbox that controls the output of the corresponding values.

I opened a read2 object for reading from details.txt. This is then looped through to get loudness, pulse, frequency, and velocity. These are all values that I will need from the bat algorithm to indicate a valid solution.

### Testing The Solution Form

This part will entail ensuring that all of the functions I made here work and read in correctly. This will make sure that I won’t need to worry about this section when implementing it into my main algorithm and other forms.

## Map Form And Graph Form (Day 3)

## File Reading And Writing (Day 4)

## The Bat Algorithm (Day 5-10)

## Displaying Solutions (Day 11)

## Adding Constraints (Day 12-14)

## Multi-Threading (Day 15)

## Improving Convergence (Day 16-20)

## Implementing Convergence Testing (Day 21)

## Graphical Map and Graph Displaying (Day 22-24)

For reference to my future self the map form is a static system and nodes are non-customisable, however I plan to implement this feature soon…