LIONS (Team #3)

Maximizing Utilization of Surplus Interconnection Service

April 26, 2021

Team Members

Anastasios Kolovos, Magaly Osorio, Jessie Rice, Justin Thakar

Mentor

Jacqueline Vogel - Tri-State Generation and Transmission Association

Executive Summary

A software tool to maximize the utilization of surplus interconnection service is needed due to the passage of FERC (Federal Energy Regulatory Commision) Order 845 on April 19, 2018 [1]. The software will evaluate the existing generation facilities point of interconnection, recognize any surplus interconnection service that already exists and then come up with the most cost effective option to more fully utilize the surplus interconnection service. This is done in four steps with the first being obtaining the difference between the existing generation and transmission line capacity it is connected to. This difference is then used to calculate what additional resources can be used to maximize the utilization of the transmission lines capacity. An economic analysis is then performed on the possible resources identified in the previous step and finally the information is presented to the user. Development of the above described tool has been completed and described in detail below.

Table of Contents

Introduction

We are creating a software tool to maximize the utilization of surplus interconnection service based upon a utility's needs. This tool will analyze an existing generation facilities point of interconnection to identify any surplus interconnection service that exists and then determine the most cost effective option to more fully utilize the surplus interconnection service. The genesis of our project was the NOPR (Notice of Proposed Rule Making) issued on December 15, 2016 as part of FERC Docket No. RM17-8-000. Without the passage of FERC Order 845 on April 19, 2018 and the subsequent clarification issued in FERC Order 845-A on February 21, 2019 our team would be working on another project idea as surplus interconnection service would not exist in Title 18 of the CFR (Code of Federal Regulations) [2]. FERC Order 845 adopted many of the recommendations contained within the NOPR on FERC Docket No. RM17-8-000 including those regarding surplus interconnection service. (EM a, o, q)

FERC Order 845 revised the *pro forma* LGIP (Large Generator Interconnection Procedures) to define surplus interconnection service as any unneeded portion of Interconnection Service established in a LGIA (Large Generator Interconnection Agreement), such that is surplus interconnection service is utilized, the total amount of Interconnection Service at the point of interconnection would remain the same [3]. A requirement that transmission providers create a process for interconnection customers to use surplus interconnection service at existing points of interconnection was also created in FERC Order 845 [1].

Now that surplus interconnection has been unlocked for utilization, the question is how can electric utilities put it to use in the most economical way. Our tool will maximize utilization of surplus interconnection service by performing the following: determining the amount of surplus interconnection service that exists at an intermittent renewable resource point of interconnection, identifying what additional resources can be added at that point of interconnection to more fully utilize the surplus interconnection service, and finally performing an economic analysis of possible resources identified in the previous step. This analysis will be performed over an industry standard 20 year planning horizon. There is not currently a single tool that we have been able to find that performs all of the tasks listed above. EPRI (Electric Power Research Institute) has developed the StorageVET® tool which can be used to analyze battery storage solutions but is limited in its time frame and is more designed for peak shaving opportunities at the distribution level. In attempting to solve this problem we had to adjust our proposed solution due to constraints placed on us by our project sponsor. Below you will find our approach to solving this problem, results we have obtained so far, a discussion of how we plan to implement our solution, and the timeline for doing so. (EM a, c, k)

Project Scope

The scope of our project includes the development of a software based solution that allows a utility to analyze colocation opportunities for existing resources and potential additional resource hybrid combinations to arrive at the best possible resource additions to maximize transmission use at an interconnection while also maximizing the use of the most economical power in a given hour. This includes the ability to store energy that would otherwise be curtailed as well. To determine the most cost effective way to maximize the existing transmission available the cost of the resource, characteristics of the resource, and the economic opportunity cost all need to be considered. The resource cost consists of design and construction of the resource and all related auxiliary equipment required as well as the ongoing operations and maintenance cost of the resource. The resource characteristics include its efficiency, heat rate for a thermal unit or charging efficiency for battery storage, and weather limitations for renewable resources. The economic opportunity cost of the resource includes fuel costs, ancillary service capabilities, and LMP's (Locational Marginal Price) of where it is located. Taking all of the above into account in addition to the existing generation profile for the already installed resource many different types and sizes of resources must be analyzed to determine what the most cost effective way to utilize the surplus interconnection service is. (EM a, o, q)

Project Requirements

The project requirements given to us by our project sponsor are to create a functional tool utilizing commonly available business software to determine the most cost effective way to maximize surplus interconnection service at an existing point of interconnection. This tool must allow the user to enter the nameplate capacity and technology of the existing resource, an 8760 hourly profile, location of resource, transmission interconnection rights, and resource technology dependent characteristics that affect its operation. From a dataset of possible resources the tool will determine the best resource or set of resources for co-location and output the hourly energy profile, hourly and total cost, and the hourly transmission utilization profile. The output will be available to the user both in graphical and text formats. (EM g, o)

Desired Project Attributes

Some desired project attributes include the ability to add additional resource locations and upload the data associated with that resource and point of interconnection. A user guide that describes how to use the program, how it works, and describes the text and graphical outputs is also a desired attribute of our project. In discussing how the program works the sources of data, methodology used to determine the least cost option, and any constraints that are present in the software should be discussed. (EM j)

Technical Description

The technical work of our project can be broken down into four distinct sections, research results, design activity results, prototype implementation and testing, as well as specification evaluation. The research results and design activity results were largely completed over the first semester of this course while the specification evaluation and the prototype implementation and testing were completed over the second semester.

Research Results

Our research conducted to complete this project covered a wide variety of topics and information. We spent time reading and understanding FERC orders, NERC Reliability Standards, and the United States Code of Federal Regulations to grasp what the genesis of our project was and how it affected utility customers across the nation. There was significant research conducted to understand the different parameters of generator and battery technology and how they affect the operational efficiency and costs of running each of them. Some time was also spent researching energy markets both electric and gas to understand their pricing structure as well as obtaining both historical and futures data for each. Yet the greatest amount of research done was on how to write out the code we did as not a single team member has any experience with writing software for a large project.

FERC Order 845 was predated by the NOPR issued on December 15, 2016 as part of FERC Docket No. RM17-8-000. This NOPR proposed the changes to the *pro forma* LGIP and *pro forma* LGIA that was ultimately passed in FERC Order 845. The changes proposed in this NOPR were proposed due to “a number of developments that impact generator interconnection, including the changing resource mix, (and) the emergence of new technologies” [3]. FERC Order 845 also modified the *pro forma* LGIP “ to allow interconnection customers to request interconnection service that is lower than full generating facility capacity” [1]. This is to allow for the implementation of battery technology where energy from a generation resource may be directed to a battery storage system for later use.

To understand the cost associated with each generator technology we had to research both the capital costs to install each resource type as well as the operational and maintenance costs associated with each. We developed a spreadsheet with both capital and operational and maintenance costs associated with each type of generator. The results of this research is depicted in Figure 1: Economic Research. To further understand the operational costs and how that would affect the capacity factor of each unit some economic research was required. This economic research centered around gas and energy markets. For the gas markets the WAHA hub was selected due to its proximity to the Tri-State Generation and Transmission Associations physical footprint. We used the gas futures trade prices from the NYMEX (New York Mercantile Exchange) and the unit heat rate to determine an operational cost for each fossil fuel based generation option. (EM c, g, h)

Design Activity Results

The design activities were successfully implemented during the spring semester however some adjustments were required as the programming of the different modules in Python started and again when the modules were integrated as described below.

An independent module for the user interface was added to be able to work simultaneously with other modules. The initial user interface was redesigned several times to improve the user interaction with the program, allowing more options to shape the final simulation result. For example, the user can select from seven different stand-alone renewable resources that the resource analysis module combines to simulate a hybrid combination of renewable resources for resource size, and the stand-alone and hybrid combinations are also passed on to the economic analysis module. The user interface also allows the user to select among thirteen options to simulate a resource size based on a nameplate percentage. Additionally, the user interface provides the user six options to select a type of plot to show the resource size. These adjustments were discovered and made while developing and testing the program to offer the expected result to our customer Tri-State Generation and Transmission Association.

The design of the forecast analysis module was accomplished and modified to improve the program flow of information, for example all CSV files for weather data, nameplate capacity, market price, and forward gas curve are read on the forecast module and integrated into an array that is the input to the resource analysis module. This change improves the flow of information on the program.

The design of the resource analysis module was accomplished and modified. The major changes are the options to calculate the resource size, the first option is based on selected renewable resources and the other option is based on the percentage of transmission line installed capacity and renewable resources. Additionally, the modules consider the combination of renewable or hybrid resources as capacity for wind and solar or wind and batteries, or wind and combine cycle and so on. Also, a plot with six different options to display the resource size was added. As mentioned on the forecast module, the profiles of the renewable resources are captured on the previous modules and passed on to this module on an array.

The design of the economic analysis module was accomplished and modified as needed to provide resource sizes and prices on any of sixteen selections to calculate resource size based on renewable resources or any of thirteen options when calculating resource size based on nameplate percentage. Creating twenty-nine options of resource sizes and prices made considerables changes on the initial design of this module although they were successfully accomplished.

The design of the co-location or last module was successfully implemented, adjustments were made to provide plots for the solution and the recommendation of the program is saved on an excel file.

Additionally, an interface program was added to retrieve the user inputs from the user interface and passed to the forecast and resource analysis modules. The final program architectural design is shown on Figure 2: Modified Program Architecture. (EM a, b, c, d, e, f, g, h, i, m, n, o, p, q.)

Prototype Implementation and Testing Results

The main main goal of this project is to assist the user to select the best sizes of additional resources that will accommodate the existing resource and increase utilization of the transmission interconnection, with the least cost of energy as possible. Therefore, the primary objective of the code is to select the best resource/generator size for the location. The display of cost in [$/MWh] for the use of the resource and the utilization of the line in [%], is essential output for this project. In addition, the code outputs some other useful information, such as energy [MWh] production by the additional resources, energy lost as curtailment [MWh], and the size of the additional resource labeled as 'Size1' and 'Size2' respectively in Figure 3: Python Data Output Table. On top of that, the team decided to allow the user to test any resource or combination of resources, by selecting the size of their preference. Here the user has the opportunity to compare the values of the code recommended sizes versus their own preferred sizes.

The way that the code selects the ideal size of the tested resource, is by evaluating the utilization versus the size data which is an array of 60 different sizes for each location site. The moment that the utilization versus size plot bends becoming more horizontal is the point that adding more energy, by increasing the size of the additional resource, does not increase the utilization of the line proportionally, and therefore there is a lot of generation in curtailment. At this point, this is the size of the resource to be added, that the code recommends to the customer. The example in Figure 4: Utilization of Added Resource vs. Resource Size Plot depicts the utilization vs size, after the addition of a solar resource of size array 0 to 300 MWh, in the site Carousel of nameplate 150 MW. The code recognizes this point in the utilization versus size plot, by analyzing the second derivative of utilization versus size. When the absolute value of the second derivative of utilization versus size is the biggest, this is the resource size that the code recommends to the customer as ideal. This is where the utilization of the transmission line is high and the cost remains low, since not much energy has been lost due to curtailment, up to this point. Continuing with the same example, based on the second derivative, the code recommends a solar resource of size 95 [MWh] in this case as can be seen in Figure 5: Second Derivative of Added Resource vs Resource Size Plot.

Overall the code is capable of generating 34 outputs, but not all at once. At the most, in one trial the customer can choose to check the code recommended sizes and all their viable combinations, with all data values provided by the table, and on top of that they can choose to check the data of three resources, of their size preference and the combinations that these resources produce with the ones recommended by the code all at once. The data in python is best displayed by the python printed table Figure 3: Python Data Output Table, and by the eight matplotlib bar-graphs that the code generates. The bar graphs colored in blue depict energy [MWh] and the second curtailment in [MWh] of the additional resources used. The bar graphs colored in light blue show utilization of the line in [%] and the additional resources used. And finally, the bar graphs colored in red indicate the cost in [$/MWh] for the use of these additional resources. Due to the limitation of matplotlib graphs in producing clean and labeled graphs of 34 outputs, the team decided that the data will be better persistent by eight rather than four bar graphs of “Energy”, ‘Curtailment”, “Utilization” and “Cost”. However, splitting each category in two or three graphs makes the matplotlib graph output a little less than optimal. An example of this is provided in Figure 6: Bar Graph Display. This is an example of the Energy vs Resources bar graph. In this example all the resources, except battery added, at the sizes suggested by the code, and their combinations, are displayed. Here the most energy is generated by the addition of a solar in a combination with a combustion cycle gas unit. Their combination generates an output of energy per year close to 250,000 MWh.

A very important feature in presenting the outputs is an excel file generated by the code. In this file there is a detailed table that displays the same outputs, as it does the table in the python console window. However, the table in excel is more clearly labeled and defined. This is in part, due to the fact that the excel platform is better suited for tables. In addition, the team wanted it to be easy to read, since the excel file is the one that the staff of the Tri-State Generation and Transmission Association will primarily be used for analysis. An example of this is provided in Figure 7: Data Output for Existing Resource. This table is generated by the site Carousel, which has a pre-existing wind resource in place. As a result, Carousel is populating this table in the excel spreadsheet, while it leaves blank the table that is designed to be populated by a solar pre-existing resource.

The excel generated table gives the option to Tri-State Generation and Transmission Association staff to use it, in order to create any graph of their liking, directly at the excel spreadsheet. Also, the code produces two graphs automatically, with all 34 possible outputs. One bar graph generated by python into an excel spreadsheet is “Utilization vs resources” and “Cost vs resources”. The graphs on the excel spreadsheet are more clearly labeled than matplotlib, also all data stays together and displayed in one graph, so they are easier to read and interpret. The main reason for this is the superior graphing capabilities of excel when it comes to producing a clean bar graph with many bars in it. An example for each graph is given in Figure 8: Excel Graphing Examples.

The customer interacts with the code through the use of a tkinter generated GUI (Graphical User Interface). The GUI is depicted in Figure 9: Graphical User Interface. The main parts of the GUI are described in detail below. At the top drop button the user can choose the site of the location that they desire to test. There are seven different sites that Tri-State

Generation and Transmission Association asked our team to consider. At the entry below, the customer can choose the minimum level desired to fill in the transmission interconnection. This feature is provided, since early on in the project, team Lions wanted to use it, as one of the criteria, that the code will implement in order to determine the ideal size of the the additional resource, however, the team decided to allow the second derivative of utilization, as the best way to selected resource/generator sizes, as described above. The '% transmission line to simulate' entry, however, has been embedded into the functionality of the code and it is needed for the code to operate, therefore a percentage required to be typed in the box. The entry below 'number of years to simulate', has no functionality, since it is not supported by the code. The team ran out of time and this feature is not available to the customer. When a 'Yes' is selected at the drop buttons to the left of the Gui, the program runs the resource with the 'Yes' selected and determines the ideal size of resource/generator, for least cost and maximum utilization of the transmission interconnection. When two compatible resources are selected, such as solar or wind with any of the gas generators or battery, then in addition to the single resource analyzed, a combination of both resources together, added to the existing resource are analyzed as well. Using the example depicted at the GUI in the figure, Solar and CC are selected at the left column of the GUI, therefore the results not only depict the recommended size and values of an ideal solar and CC resource, but they also show the ideal recommendation of the combination of the two. (look at Sol, CC and SolCC) Figure 3: Python Data Output Table. The column in the middle of the GUI 'Select resource size to simulate' allows the customer to choose the resource/generator, as well as the size of it. Therefore for the resources selected in the middle column the customer specifies the size, not the code, however the code analyzes this size and produces all the available outcome values. Finally, at the right 'Select plot resource size', the GUI allows the user to select the type of plot displayed by the code. These line plots are 'Size vs Size', which is really a default plot, with no real purpose, and is there simply as a space holder. The rest of the plots are 'Energy vs Size','Curtailment vs Size', 'Utilization vs Size', '1st derivative vs Size', '2nd derivative vs Size', and they provide the customer with important information. To prevent overpopulation of the screen and making it hard to find specific plots, only one type of plots are allowed to run at once. Each plot type geretates ten plots, one for each resource and combination of two resources available in this program. In this example the plot of 'Utilization vs Size' was selected and an example of this is depicted in Figure 4: Utilization of Added Resource vs. Resource Size Plot.

Testing of the code was implemented in many levels. First the code needed to run. When errors were displayed the code had to be fixed. Then prints were used throughout the development of the code and the results of the prints were examined and interpreted, for being the right type of data, for instance a list or a matrix as opposed to a float or integer. The results had to be reasonable, for instance 25 [$/MWh] is a reasonable answer but 29069899 [$/MWh] is way too large and obviously a code error. As the code got closer to the end, the results were primarily monitored by the python table Figure 3: Python Data Output Table, and the python graphs explained earlier. The data of the table/graphs were evaluated in two ways primarily. Are the data displayed where they should, and are the results within the realm of expectation?

The best way to demonstrate how the results were tested is by walking through an example. For this example, the figures already provided in Appendix B will be used. Based on the customer preference, depicted by Figure 9: Graphical User Interface, the user wants to examine the site Carousel. Based on the selection at the left side of the GUI, they want to see the code's size recommendation and important value outputs for Solar, CC, CT, RIC, Aro, Batteries and any combination of solar with CC, CT, RIC, Aro, Batteries. By the choices in the middle column of the GUI the customer is also interested to see the data output for specific size of Solar (.2* nameplate) in this instance, CC2 (.4* nameplate), and a Battery attached to the solar resource in size (.5* nameplate), The nameplate of the Carousel site is 150 MW. Based on the plot selection displayed here, the customer is interested in a line plot of 'Utilization vs Size'.

The results are displayed in the python table depicted in Figure 3: Python Data Output Table. Examining this table closely, it is clear that all data are displayed where they are supposed to. So, here the table displays the output, of all the single resources, that the code's size suggestion was requested (left side of the GUI). The table displays Solar as Sol, CC, CT, RIC, Aro. In addition, the table displays the combination of Solar plus any of the following CC, CT, RIC, Aro, as SolCc, SolCt, SolRic, Sol_a. The specific size selection, made by the customer, at the middle column of the GUI, are also properly displayed. So here, Solar (.2* nameplate), CC2 (.4* nameplate), are displayed as Sol1, CC2. Aslo, the combination of the selections in the middle of the GUI are displayed as Bat2, Sol1Cc, Sol1Cc2, Sol1Ct, Sol1Ric, Sol1_a. And as important is that there is nothing that was not requested by the customer that was displayed. This is a great sign that the code operates as should.

Continuing with the same example and examining the table closer here are some additional comments. The 'Utilization' column is always displayed, it starts at 44 [%] in this case, which is the utilization provided by the existing resource, wind in this case. As additional resources have been added to the pre-existing one, the utilization is increasing.

This is true in any single display in this example. Furthermore, when one additional resource is added, the ‘Size1’ column is populated, and when a combination of two additional resources is used, ‘Size1’ is populated by the solar resource and ‘Size2’ is occupied by the gas or battery. When no additional resource was implemented, then ‘Size1 & 2’ columns display 0. All these criteria were met in this example.

Some more important observations that validate the code and the results are as follows. Only Solar or Wind as single resources can produce curtailment, since the gas can be utilized as needed, and this is the case here. All columns but utilization, as explained above, display 0, unless additional resources have been examined. This is also true here. Utilization values cannot surpass 100%. As the size of an additional solar resource increases so does its energy output and/or its curtailment. This is usually the case for gas as well but not necessarily, since no matter the size of the gas generator, it will only run to the capacity needed to keep the line full. All numbers seem reasonable and nothing strikes as impossible. And all these requirements are met by the code, in the example displayed here.

Finally, at the right column of the GUI the customer wanted a plot display of ‘Utilization vs Size’ and they got exactly that as demonstrated in Figure 4: Utilization of Added Resource vs. Resource Size Plot. The plot is very reasonable. As the size of the resource increases, x-axis, so does the utilization of the line. However, the graph bends at some point around 95 [MWh]. From that point farther, utilization increase slows down. This is the point that lots of the energy input becomes curtailment, and this is the size of the resource that the code recommends. Any size of resource bigger than 95 [MWh], in this case, will lose too much energy in curtailment and it will be too expensive and wasteful to operate.

All these requirements and inspections have also been implemented in the matplotlib bar-graph, the excel code generated table and excel bar-graph as well. Since the data are identical to the ones displayed by the python table, there is no need for further explanation.

So, some conclusions that we can draw from this specific example at the site Carousel, and using the data from the excel generated table, with values identical to the python printed table, since we are referencing the same example Figure 7: Data Output for Existing Resource. Based on the table the most usable energy production occurs with the combination of 75 [MW] battery and 160 [MW] solar. This combination produces 374,034 [MWh] energy in a year of operation. Since this combination produces the highest energy output, it also provides the highest utilization of the line 73.43%. In addition, it should be noted that the use of solar and battery combination has a very low cost to operate 24.22 [$/MWh]. The cheapest of all resources used here is CC alone. The reason for this is because a combustion turbine is cheap to operate ,partially because it produces no curtailment at all, since you can always adjust the output of a gas turbine. However, the utilization of CC is only 56.17 [%], which is significantly lower than that of the battery and solar combination. Therefore if it was my decision to make, I would most likely favor the battery and solar combination, as the additional resource of energy, for the better utilization of the transmission interconnection.

In reality the data was investigated in much more scrutiny, by the team, and many more scenarios were runned during the testing and debugging phase. Using the approach described above, many modifications on the code have occurred, before and after the code was declared complete.

Specification Evaluation

In this section some of the specifics of the code functionality will be discussed. The code treats the energy generation from solar or wind the exact same way, as an hourly varying irregularly shaped type of function. The two only get differentiated during the economic analysis stage, since the price of generation of the two energy resources is different. The code knows how to differentiate the two based on the site of analysis. If the existing resource is wind then the additional energy will be solar and vice versa. The code can recommend the ideal size of resource, and in addition the customer can test up to two resource sizes per trial of his/her own preference.

When it comes to gas resources, gas turbines are only allowed to operate when the use of gas is cheaper than the market price. The code compares the two hourly and implements that logic. In addition, gas is not allowed to exceed the nameplate of the transmission interconnection, and that is reinforced by code as well. The code can recommend the ideal size of generator, and in addition the customer can test up to two different generator sizes per trial of his/her own preference.

A combination of two or more additional resources is analyzed any time that the customer chooses a wind or solar resource, in addition to any gas or battery. In the combination of solar and gas the code analyzes the following four scenarios. The first scenario is that the code first determines the ideal size of solar or wind and in succession, the code also determines the recommended size of the gas turbine. Another option is that the customer can choose a solar/wind resource size and the code will match it with the ideal size of gas generator. For option 3, the code determines the ideal size of solar/wind resource. This ideal size of solar/wind is matched by the customer's decision of testing a specific gas size turbine, and the code produces the data output of this combination. The fourth and final combination between solar/wind and gas is both resource and gas generator sizes are picked by the customer and the code produces the appropriate outputs of such combinations.

For the solar/wind and battery combination, the user chooses the size of battery that they want to test and the code determines the ideal size of solar resource, that the size of the battery is best suited to accommodate. The customer can test a maximum of 3 different battery sizes in each trial.

The code is also supplemented with several mechanisms to prevent it from stopping running due to errors. A not exclusive list of such mechanisms are the following. The code can run with a minimum input from the customer in the GUI. This gives the freedom to the user to test as little, or as much resources at once, as they please. If no resource is picked the code will still run, however all output data populating the tabla and graphs will be zero with an exception of the utilization levels that will be equivalent to the utilization supplied by the existing resource. This is due to several mechanisms embedded in the code to prevent errors of division by zero, accomplished through the use of if, else statements. Also, mechanisms to convert constants into arrays are in place for when a line-plot graph is created, but one of the variables is a constant rather than an array. Such features and many more, are in place to guarantee a smooth run, at every trial and make the code a user friendly experience.

Timeline

As the team resumed working on the project at the beginning of spring semester with feedback from the work done during Christmas break, it realized that a few tasks were needed on the schedule. The first task added was the definition of blocks and functions of each module to smooth the integration of the different modules into the main program. The second task added was the user interface module, which was broken into an individual module that previously was part of the forecast module, to improve the flow of work on the project. The third and fourth tasks added were to retrieve values from the user interface.

This step was a priority, and there were two options to achieve the goal. The first one was to use Python code and a function in Excel and the second one was to use only Python code. The team faced issues retrieving the values captured on the user interface and decided to work from two different ends to increase the success of the task. The fifth task added was the integration of all modules to run for a base of one year, which is broken down from the main program and made more sense to improve the speed and flow of the programing. The team's schedule on the proposal was shifted by a week or two; most of the module's preliminary draft met the deadline, but there was work in progress due to dependency among the modules. By midterm, the team decided to set two goals regarding the schedule and the pending work. The first goal was to complete all modules by April 5, 2021 and the second one was to integrate modules and make the program run for one year by April 12, 2021; both goals were accomplished on time. The comparison of final schedule, midterm schedule, and proposal schedule are shown on Figure 10: Gantt Chart Comparison. (EM b, c, f, g, h, m, p).

The team's milestones were broken down as follows: the first milestone was the user interface module to allow the user interaction with the program, requesting different inputs such as transmission line to simulate, years to simulate, renewable resources to simulate, resource size to simulate, and type of plot to display the resource size. The second milestone was the forecast analysis module, which reads weather data by location from a CSV file and the nameplate capacity by location from another CSV file, then calculating the forecast of unused transmission capacity in megawatt-hour, which is the input to the resource analysis module. The third milestone was the resource analysis module, which finds resource size options by stand-alone or combination of renewable resources in megawatt-hour at any given location and is input to the next module. The fourth milestone was the economic analysis module, which calculates the pricing of the reserved capacity in megawatt-hour that can be a stand-alone or hybrid combination of resources. The fifth milestone was the colocation module, which selects the most cost-effective option that satisfies the forecast and maximizes the surplus transmission utilization. The sixth milestone was the integration of the five modules to make the program run for a basis of one year. The seventh milestone was dropped as the team ran out of time. The eighth milestone was to integrate the final program and make it run for one-year base. The ninth milestone was to write the user manual to provide to Tri-State a guide to navigate the program. The team's last milestone was the demo day to present the project. (EM a, b, c, d, e, f, g, h, i, m, n, o, p, q.)

Deliverables

1/18 Notebook 1 -- Prior to the start of the semester Magaly and Anastasios had programmed a GUI, and large portions of Modules 1-3.

1/25 Notebook 2

2/1 Notebook 3

2/8 Notebook 4

2/15 Notebook 5, Video Report 1

2/22 Notebook 6

3/1 Notebook 7

3/8 Notebook 8, Progress Report

3/15 Notebook 9

3/20 Presentation -- By this point Anastasios had most of the Module 3 code working in its entirety, meaning the 2nd derivative portion was working correctly.

3/22 Notebook 10

3/29 Notebook 11

4/5 Notebook 12

4/12 Notebook 13 -- By this point we had a more or less finalized GUI after several iterations.

4/19 Notebook 14

4/23 **Final Demo,** Poster

4/26 Notebook 15, **Final Report,** Video Report 2

All milestones were met with minimal fuss.

Possible Future Work

More of a user manual still needs to be written; thus far we have a “How to get started with Python” section and some explanation of the GUI; much of the material to explain the rest of the program in depth, especially the economic analysis module, is there, it just needs to be translated into words *and* pictures.

More options for multi-year analyses would also be useful. That could complicate the analysis but not necessarily by much, unless we were to add things like load growth and perhaps even monetary inflation.

Most of the graphing we’ve utilized so far is using more basic Python libraries--there are other graphing libraries that are more robust, more colorful, and more interactive. Some of that could be done with the original libraries used like Matplotlib and Pandas’ graphing capabilities but other libraries’ greater hipness and overall attractiveness is hard to deny.

The GUI is fine but is limited to one window. It could be expanded to multiple menus and submenus, scattered throughout the program. More menus would help to differentiate the different modules of the program, and increase the sense of logic and flow between them.

Per interactivity and GUI windows, we could have some interactive windows with sliders for the various options represented in a given part of the program, perhaps even some kind of giant slider-and-option-filled monstrosity that allows for very granular and very specific control of the output plots and conditions given for the analysis.

One thing that can be done with the code is to integrate it into Unity, which is useful for any kind of graphical integration. One could create a very basic model of Tri-State Generation and Transmission Association’s utility footprint and create a scale model of it, on which solar panels and wind turbines and so on can take Python code and run with the Unity interface.

Distribution of Work

The bulk of our work for this project can be divided into three primary categories. These three categories include research, writing of code, and submissions. The research includes research done both at the beginning and throughout the duration of this project and was primarily accomplished by Anastasios, Magaly, and Jessie. The writing of code was mainly spearheaded by Anastasios and Magaly with lesser contributions by Jessie and Justin. The submissions made by the team in the form of reports and videos had contributions from all with Jessie and Justin leading the development of formating, proofreading, and putting the video submissions together.

Research

The research that our team conducted consisted of three significant parts. The first was research of the regulations and rules surrounding surplus interconnection service. The second was the researching of the different types of resources and their operational parameters and the third was the researching of how to write the code to make our software run.

The research of regulations and rules surrounding surplus interconnection service was primarily performed by Jessie as he is a NERC (North American Electric Reliability Council) certified system operator and works within the rules and regulations set forth by FERC and NERC on a daily basis. Jessie's familiarity with FERC Orders, NERC Reliability Standards, and Title 18 CFR allowed him to understand where to look and how to properly read these sometimes confusing texts.

Researching the different types of resources and their operational parameters was accomplished by all team members. Jessie has hands-on experience as a power plant operator and also with the economic and reliability dispatch of a number of different types or generation resource technologies as an hourly trader at a merchant generation fleet owner and a system operator with an ISO. Jessie was also able to research the economic factors due to his familiarity with the gas and electricity trading markets throughout the United States. Magaly has experience with different types of generation resources and their substation requirements as she currently works as a substation designer for an electric co-op. Anastasios and Justin do not have any experience with generation or transmission outside of their knowledge gained from through school but both proved valuable in this role and made significant contributions.

The research to write the code required for our project was done by all team members but largely was performed by Anastasios and Magaly. None of our team members have any software experience outside of school, however Anastasios has a real passion for it and desired to do as much of the code writing as possible. Magaly focused on the research required for the GUI while Jessie and Justin worked on designing task specific functions required throughout the process.

Writing Code

This project had five major coding components. These components are the GUI which allow the user to interact with the system. Module 1 which uploads information from .csv files for all location sites examined and organizes the information in dataframes. Module 2, that contains code responsible for resource/generator size decision making, determining energy outputs for all resources in use and displaying this data in a matplotlib line-plot diagram as a 'Size vs Size',, 'Energy vs Size', 'Curtailment vs Size', 'Utilization vs Size', '1st derivative vs Size', '2nd derivative vs Size'. Module 3 with the main purpose of analyzing the cost of the resources in [$/MWh] and displaying all important information in a python table and bar-graphs. Finally, module 4 with a main function of generating an excel spreadsheet, with all the important data produced by the code, in a customer friendly form of tables and graphs.

All code development began after the termination of the fall semester, which was primarily dedicated to preliminary work. Magaly spent part of her Christmas break becoming familiar with python spyder, pandas and tkinter and during that time she produced an initial prototype of a functional GUI. There were some initial problems with retrieving data from the GUI into the editor window of python. Primarily Justin worked with Magaly, also Jessie did some, in fixing that issue. Their initial attempt was through pulling the data into an excel spreadsheet, using python to call them back into the code. Jessie created some function that could accomplish such a task. Justin devoted several weeks into this task. The solution came weeks later by Magaly and the team was able to retrieve the data from the GUI directly into the code without the need for excel files. The solution in part involved the creation of a new python file 'GUI_values.py' document, that facilitated such an action, developed by Magaly.

Magaly also created Module 1. The work for Module 1 took place throughout the Spring semester. The purpose of this module was to organize information for several location sites into a dataframe. The main reason that Magaly was chosen for this task was because she had worked in the organization of such information into .csv files , since the Fall semester. She completed the code of this module on her own.

The first draft of module 3, was created by Anastasios, during Christmas break. He spent time familiarizing with python spyder and openpyxl, as well as other python-excel packages, such as pandas, xlwings, xlrd and xlwt, for that purpose. The first draft was a first attempt for determining the cost of resources coming from Module 2. However, since module 2, was not designed yet, major changes and continuous improvements of the Module 3 code occurred throughout the semester. The end product analyzed 34 instead of the initial version of 10 outputs, it included additional graphs and a table.

The production of Module 2 started during the early weeks of the Spring semester. While there was an attempt early on for collective work in the creation of this module, it became clear early that one person producing the code alone will be advantageous. The main reasons for this decision was the number of unknowns, which made it hard for the team to follow a coherent plan, in the coding of the module. The number of unknowns created the need for quick decision making and changing of plans in a short notice, making it a more suitable task for one person rather than a group. Anastasios jumped in and he took the initiative to complete this module, since he had already worked on Module 3 and he was more aware of what the output of this module should be and how these outputs should complement the inputs of module 3. Module 2 was a hard and challenging module to complete, with many functions and the most lines of code from all modules.

Module four was small, Anastasios had already started creating some of its functions, through the use of openpyxl during the Christmas break. As a result he was the one completing it, during the first week of April.

The GUI was connected with the rest of the code somewhere in the beginning of April. Magaly was responsible for connecting the Gui with the code in Module 1. This is because she was the creator of both the GUI and Module1 and therefore she was best suited for it. The connection of the GUI with the code in Module 2, which was most of the connection between the GUI and the code, was primarily the work of Anastasios, with the help of Magaly. The main reason that these two were chosen to complete this task was because Magaly created the GUI and Anastasios created Module 2 and therefore, they were most familiar with the particular code.

Some noticeable efforts in the creation of the code was Jessie's idea for the use of GitHub, as platform, for the team to coordinate efforts in the creation of the code. Even though Jessie shared his GitHub account with the team, and even though this idea got some momentum initially, it did not go much farther. The main reason for this was due to the unfamiliarity of the majority of team's members with the GitHub environment, the limitation of time to learn, particularly during an already busy semester.

Furthermore lots of code was written and rewritten, modified or rejected all together, during the semester. There were many examples of that in all modules. So for instance, Justin, with some help from Anastasios, created some graphs and tables in excel, generated by python code through openpyxl, that were not utilized at all in the final project, due to significant modification of Module 3 by the end of the semester.

Submissions

Templates and outlines for all project course submissions were created by Jessie who consulted the rubric for each assignment to ensure all required information was provided and it was done so in the correct format. Editing of all submissions was also done by Jessie to ensure that figures were properly numbered and to ensure the proper font, spacing and alignment was used throughout the document submitted. Basic grammar editing was done by Jessie as well but any changes were minimal as it is important that every team member's voice is heard through our submissions. Any time there was video editing to do Justin did this and submitted it via youtube. Every team member contributed to the weekly notebooks both in their content and the submission as well.

References

[1] U. S. A. Federal Energy Regulatory Commission, Reform of Generator Interconnection Procedures and Agreements, Docket No. RM17-8-000, Order No. 845, Final Rule, April 19, 2018.

[2] U. S. A. Federal Energy Regulatory Commission, Reform of Generator Interconnection Procedures and Agreements, Docket No. RM17-8-001, Order No. 845-A, Order on Rehearing and Clarification, February 21, 2019.

[3] U. S. A. Federal Energy Regulatory Commission, Standard Large Generator Interconnection Procedures (LGIP), February 21, 2019.

[4] Burcu Mantar Gundogdu, Dan T. Gladwin, Shahab Nejad, David Andrew Stone. (July 2019). Scheduling of grid-tied battery energy storage system participating in frequency response services and energy arbitrage. IET Generation, Transmission & Distribution.

[5] Marc Bollinger, Joachim Seel, Dana Robson, Cody Warner. (November 2020). Utility-Scale Solar Data Update: 2020 Edition. Lawrence Berkeley National Laboratory.

[6] Richard Bowers. (January 2021). U.S. Wind Energy Production Tax Credit Extended Through 2021. U.S. Energy Information Administration

[7] Jonathan Devilbiss. (February 2021). Electric Monthly Update. U.S. Energy Information Administration

Budget

The budget for this project is solely the time opportunity cost associated with every hour spent working on this project by team members. There was no need to purchase equipment or information to complete this project as access to technical journals is provided to us through the ASU (Arizona State University) library.

Acknowledgements

This project gave team Lions an enormous opportunity to understand the world of generation and transmission co-operates. Specifically, we got to learn lots about Tristate Inc. their territory, and their needs and challenges, when it comes to generation of power and the requirements to maintain the transmission interconnection at full capacity, at all times. We got a glimpse of the future of the electrical-corporate world, as we were called to solve one of the major issues, when it comes to the future generation of electricity, which is how to keep the switch on when no wind blows and no sun shines. We were called to solve this problem and to do so in the most cost efficient way. This is how the co-location project was born. Because of this tremendous learning opportunity we want to thank TriState inc. Their contribution to us was not monetary but they provided us with documents, guidance and a lot of their employees' time. And while the team Lions thanks and acknowledges all the supervisors and staff that got involved behind the scenes, for assisting us with our project, our thank you is directed mostly to Jaqueline Vogel, our mentor. She participated in all weekly meetings, listened to, contributed and encouraged the team throughout this process. She stayed active in the google handouts and in the slack communication platforms and she was quick to address every question directed towards her. Therefore, from all members of team Lions a big thank you to Jacqueline and to Tristate Inc. for being part of two memorable and productive semesters in ASU. Also, we want to acknowledge ASU staff, and especially the TA's for this class. In particular we want to thank Abdullah Akbaythat for all the interaction he had with our team. Finally, and most definitely, team Lions want to give a big thank you to our professor Dr. Kozicki, for his time, his advice, for teaching, and guiding us throughout this process.

Appendix A

List of entrepreneurial mindset indicators.

a) Critically observes surroundings to recognize opportunity.
b) Explores multiple solution paths.
c) Gathers data to support and refute ideas.
d) Suspends initial judgment on new ideas.
e) Observes trends about the changing world with a future-focused orientation / perspective.
f) Collects feedback and data from many customers and customer segments.
g) Applies technical skills / knowledge to the development of a technology / project.
h) Modifies an idea / product based on feedback.
i) Focuses on understanding the value proposition of a discovery.
j) Describes how a discovery could be scaled and / or sustained, using elements such as revenue streams, key partners, costs, and key resources.
k) Defines a market and market opportunities.
l) Engages in actions with the understanding that they have the potential to lead to both gains and losses.
m) Articulates the idea to diverse audiences.
n) Persuades why a discovery adds value from multiple perspectives (Technological, societal, financial, environmental, etc).
o) Understands how elements of an ecosystem are connected.
p) Identifies and works with individuals with complementary skill sets, expertise, etc.
q) Integrates / synthesizes different kinds of knowledge.

Appendix B

EconomicAnalysisM3							
	resources						
inputs				Natural Gas			
	Wind	Solar	Batteries	Combine Cycle	Combination Turbine	Reciprocating IC	Aeroderivative
CapEx [$/kW]	1470	1,331	1400	1079	710	1802	1170
Fixed O&M [$/kw/yr]	44	15.19	15	14.04	6.97	35.01	16.23
Var O&M [$/MWh]	N/A	N/A	N/A	2.54	4.48	5.65	4.28
Heat Rate [BTU/kWh]	N/A	N/A	N/A	6431	9905	8295	9124
frwd. curve [$/mmBTU]	N/A	N/A	N/A	givenDown	givenDown	givenDown	givenDown
margin (apprx.)	0.07	0.07	N/A	N/A	N/A	N/A	N/A
life [yr] (apprx.)	25	25	15	30	30	30	30
Capacity Nameplate [MW]	100	100	100	418	237	21	105
Capacity factor (apprx.)	0.40	0.30	1.00	1.00	1.00	1.00	1.00
efficiency	N/A	N/A	0.75	N/A	N/A	N/A	N/A
hr/yr	8760	8760	8760	8760	8760	8760	8760
PTC [15 $/MWh](1st 10 yr)	15	N/A	N/A	N/A	N/A	N/A	N/A
ITC (20% capEx)	N/A	0.20	N/A	N/A	N/A	N/A	N/A
calculations							
CapEx [$]	147000000	133100000	N/A	N/A	N/A	N/A	N/A
fixed [$]	110000000.00	37975000.00	N/A	N/A	N/A	N/A	N/A
total Energy [MWh]	8760000	6570000	N/A	N/A	N/A	N/A	N/A
production tax Credit [$]	52560000	26620000	N/A	N/A	N/A	N/A	N/A
Total cost (+margin)	218750800	154566850	N/A	N/A	N/A	N/A	N/A
output							
PPA [$/MWh]	**24.97**	**N/A**	**33.30**	givenDown	givenDown	givenDown	givenDown
	N/A	**23.53**	**31.37**				

Figure 1: Economic Research

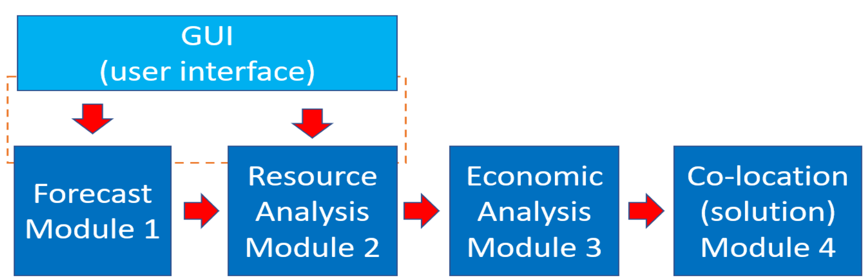


Figure 2: Modified Program Architecture

	Enrg_Res	Enrg_Curt	$/MWh	Utiliz	Size1	Size2
Sol	228363	17739	25	62	95	0
Sol1	77716	0	23	50	30	0
Sol2	0	0	0	44	0	0
CC	147255	0	22	56	105	0
CC1	0	0	0	44	0	0
CC2	98510	0	22	52	60	0
CT	44819	0	35	48	105	0
CT1	0	0	0	44	0	0
CT2	0	0	0	44	0	0
RIC	59753	0	31	49	105	0
RIC1	0	0	0	44	0	0
RIC2	0	0	0	44	0	0
ARO	55911	0	32	49	105	0
ARO1	0	0	0	44	0	0
ARO2	0	0	0	44	0	0
Bat	0	0	0	44	0	0
Bat1	0	0	0	44	0	0
Bat2	374034	11095	24	73	160	75
SolCc	242978	17739	25	63	95	10
Sol1Cc	187303	0	22	59	30	75
SolCc1	0	0	0	44	0	0
Sol1Cc2	170287	0	23	57	30	60
SolCt	232609	17739	25	62	95	10
Sol1Ct	110826	0	27	53	30	75
SolCt1	0	0	0	44	0	0
Sol1Ct2	0	0	0	44	0	0
SolRic	234134	17739	25	62	95	10
Sol1Ric	121965	0	26	54	30	75
SolRic1	0	0	0	44	0	0
Sol1Ric2	0	0	0	44	0	0
Sol_a	233744	17739	25	62	95	10
Sol1_a	119101	0	26	54	30	75
Sol_a1	0	0	0	44	0	0
Sol1_a2	0	0	0	44	0	0

Figure 3: Python Data Output Table

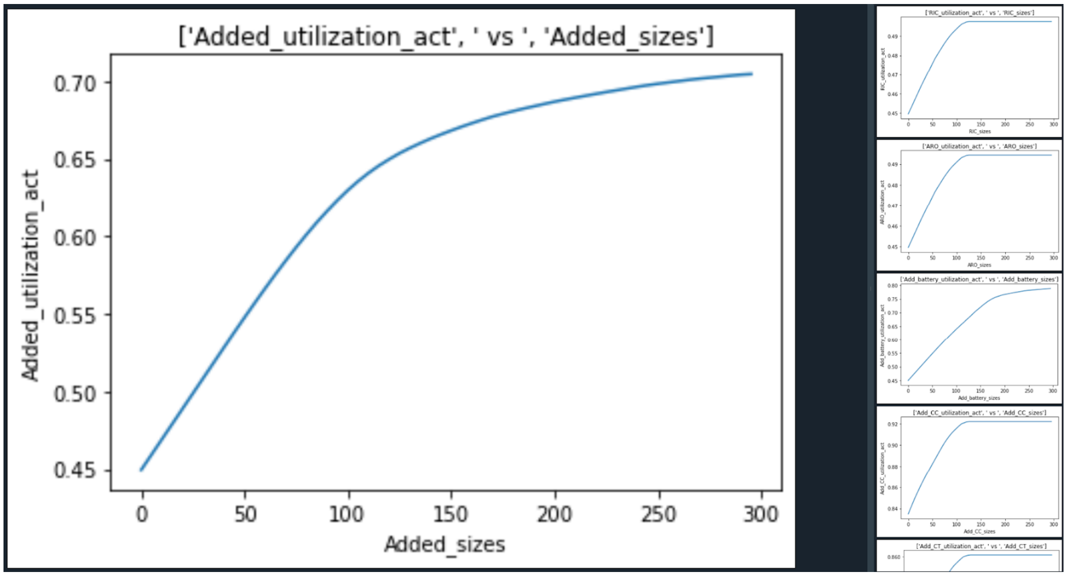


Figure 4: Utilization of Added Resource vs. Resource Size Plot

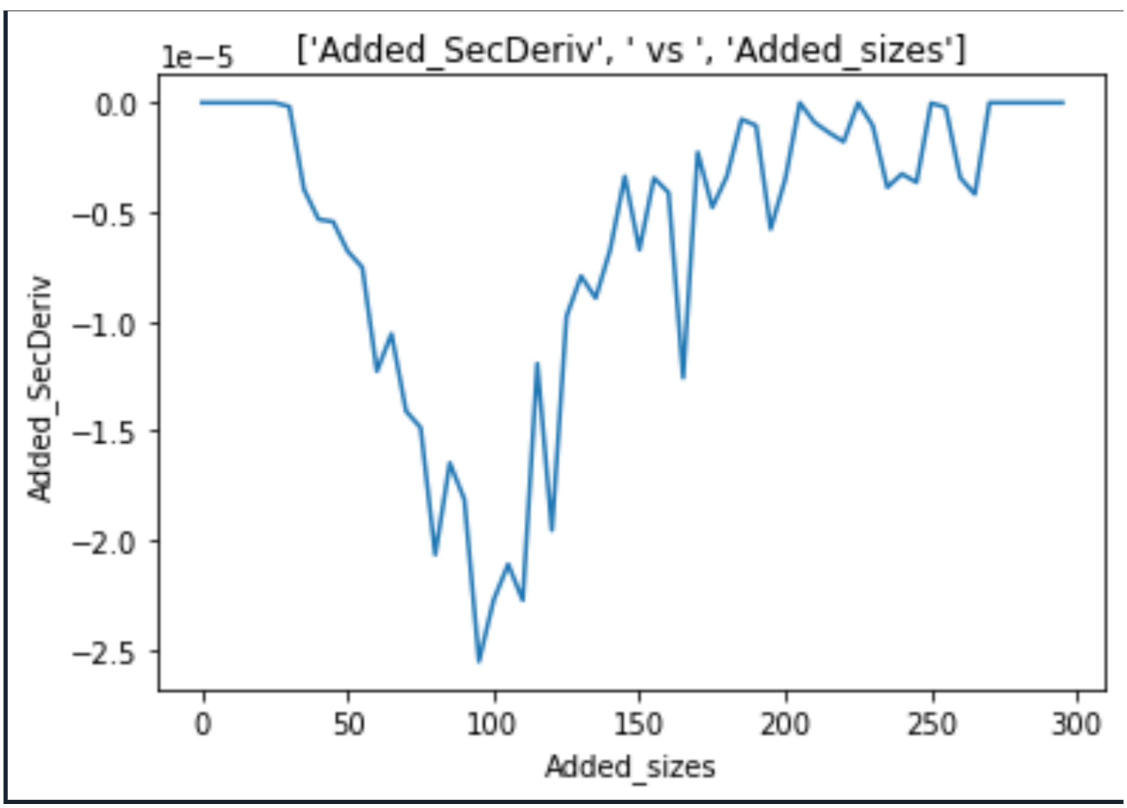


Figure 5: Second Derivative of Added Resource vs Resource Size Plot

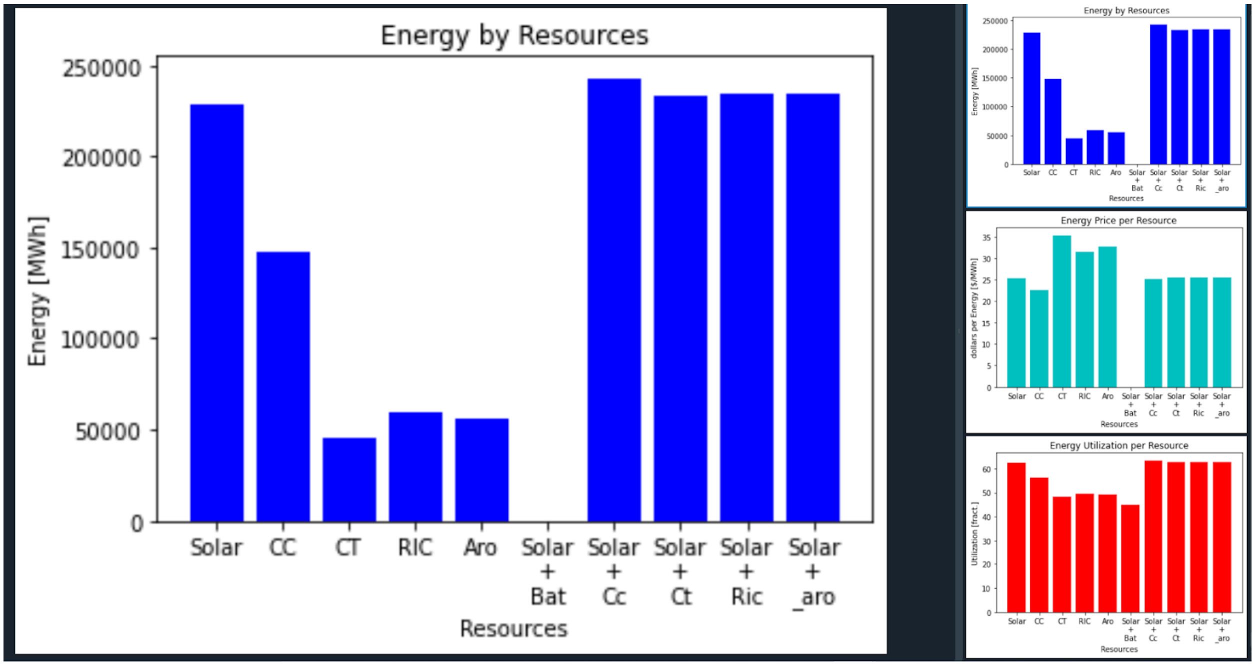
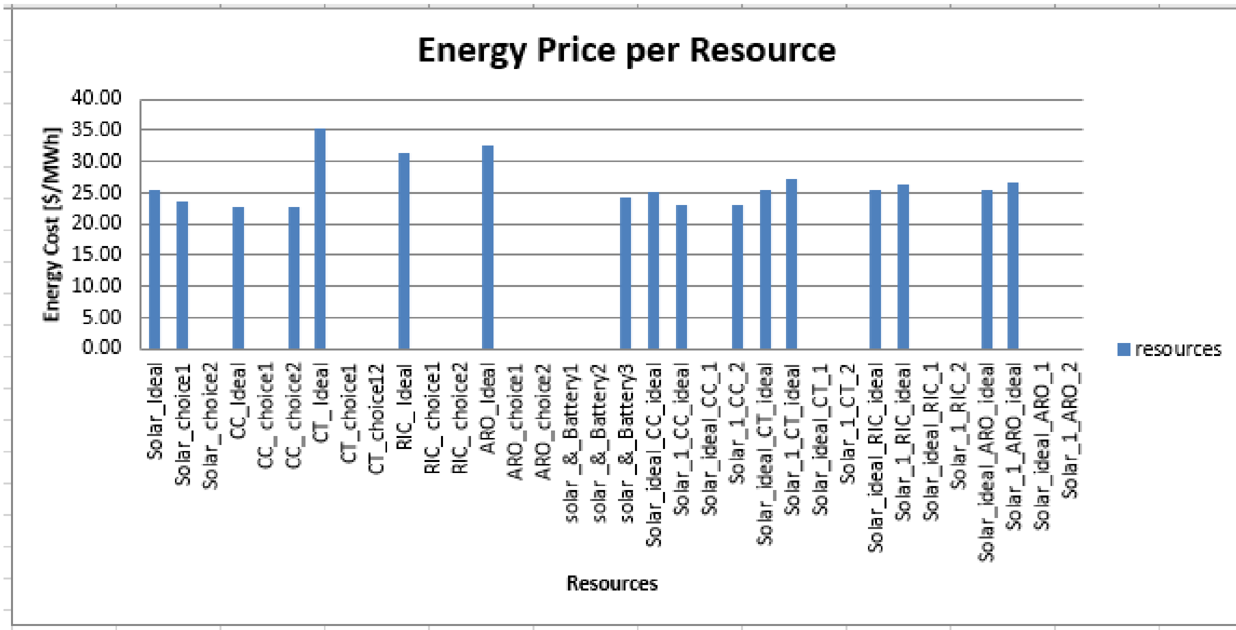


Figure 6: Bar Graph Display

	A	B	C	D	E	F	G	H	I
1	**Existing Resource**		**Additional Resources**						
2	**Wind**	case num.	resources	values					
3				Energy resource [MWh]	Energy Curt. [MWh]	resource [$/MWh]	Utilization [%]	size of 1st resource	size of end resource
4		1	Solar_Ideal	228,363.50	17,739.41	25.35	62.34	95.00	0.00
5		2	Solar_choice1	77,716.71	0.00	23.53	50.88	30.00	0.00
6		3	Solar_ choice2	0.00	0.00	0.00	44.96	0.00	0.00
7		4	CC_Ideal	147,255.78	0.00	22.58	56.17	105.00	0.00
8		5	CC_ choice1	0.00	0.00	0.00	44.96	0.00	0.00
9		6	CC_ choice2	98,510.62	0.00	22.60	52.46	60.00	0.00
10		7	CT_ Ideal	44,819.70	0.00	35.34	48.37	105.00	0.00
11		8	CT_choice1	0.00	0.00	0.00	44.96	0.00	0.00
12		9	CT_choice12	0.00	0.00	0.00	44.96	0.00	0.00
13		10	RIC_ Ideal	59,753.65	0.00	31.49	49.51	105.00	0.00
14		11	RIC_ choice1	0.00	0.00	0.00	44.96	0.00	0.00
15		12	RIC_ choice2	0.00	0.00	0.00	44.96	0.00	0.00
16		13	ARO_Ideal	55,911.61	0.00	32.70	49.22	105.00	0.00
17		14	ARO_choice1	0.00	0.00	0.00	44.96	0.00	0.00
18		15	ARO_choice2	0.00	0.00	0.00	44.96	0.00	0.00
19		16	solar _&_Battery1	0.00	0.00	0.00	44.96	0.00	0.00
20		17	solar _&_Battery2	0.00	0.00	0.00	44.96	0.00	0.00
21		18	solar _&_Battery3	374,034.58	11,095.63	24.22	73.43	160.00	75.00
22		19	Solar_ideal_CC_ideal	242,978.24	17,739.41	25.19	63.45	95.00	10.00
23		20	Solar_1_CC_ideal	187,303.30	0.00	22.98	59.22	30.00	75.00
24		21	Solar_ideal_CC_1	0.00	0.00	0.00	44.96	0.00	0.00
25		22	Solar_1_CC_2	170,287.27	0.00	23.02	57.92	30.00	60.00
26		23	Solar_ideal_CT_ideal	232,609.58	17,739.41	25.54	62.67	95.00	10.00
27		24	Solar_1_CT_ideal	110,826.53	0.00	27.06	53.40	30.00	75.00
28		25	Solar_ideal_CT_1	0.00	0.00	0.00	44.96	0.00	0.00
29		26	Solar_1_CT_2	0.00	0.00	0.00	44.96	0.00	0.00
30		27	Solar_ideal_RIC_ideal	234,134.44	17,739.41	25.51	62.78	95.00	10.00
31		28	Solar_1_RIC_ideal	121,965.65	0.00	26.42	54.24	30.00	75.00
32		29	Solar_ideal_RIC_1	0.00	0.00	0.00	44.96	0.00	0.00
33		30	Solar_1_RIC_2	0.00	0.00	0.00	44.96	0.00	0.00
34		31	Solar_ideal_ARO_ideal	233,744.44	17,739.41	25.52	62.75	95.00	10.00
35		32	Solar_1_ARO_ideal	119,101.10	0.00	26.72	54.03	30.00	75.00
36		33	Solar_ideal_ARO_1	0.00	0.00	0.00	44.96	0.00	0.00
37		34	Solar_1_ARO_2	0.00	0.00	0.00	44.96	0.00	0.00

Figure 7: Data Output for Existing Resource



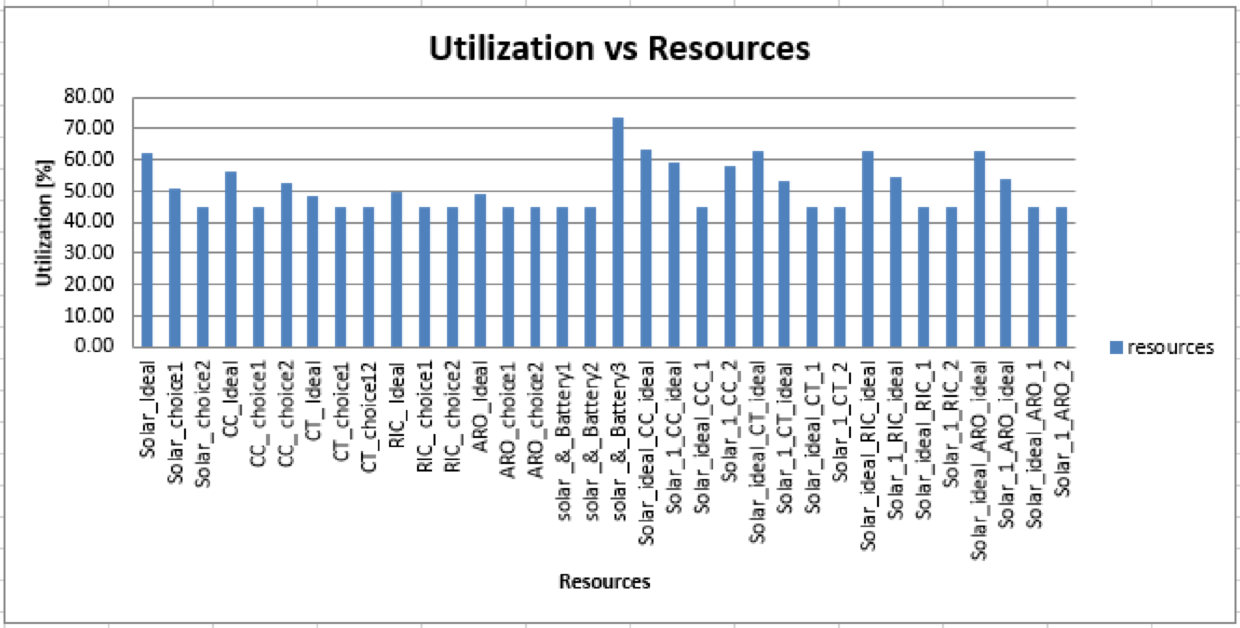


Figure 8: Excel Graphing Examples

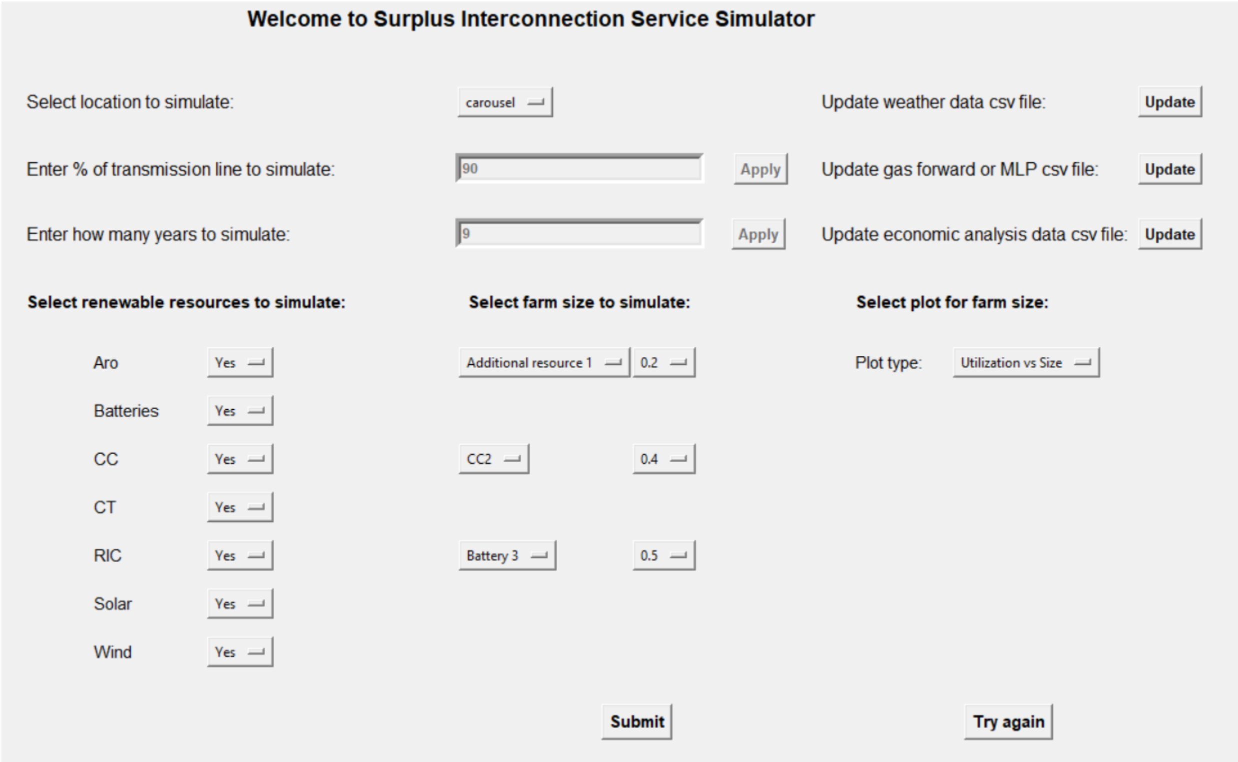


Figure 9: Graphical User Interface

Project Surplus Transmission and Hybrid resource co-location

Tri-State
Jacqueline Vogel

Project Start Date: 1/11/2021
Scrolling Increment:

Milestone Description	Category	Assigned To	Progress	Start	Actual	Midterm	Proposal	No. Days
Tesing and refinement								
Definition of blocks & fuctions of each module	Task	Anastasios, Magaly Jessie & Justin	100%	1/11/2021	2/15/2021			35
Finalized database schema	Task	Anastasios, Magaly Jessie & Justin	100%	1/11/2021	2/15/2021		1/18/2021	35
User interface prototype	Milestone	Magaly	100%	12/7/2020	1/25/2021			49
Retrieve values from GUI	Task	Justin/Jessie/Magaly	100%	1/28/2021	2/18/2021			21
Function to retrive values from GUI	Task	Jessie	100%	1/28/2021	2/18/2021			21
Program forecast module prototype	Milestone	Magaly	100%	1/25/2021	2/8/2021	4/5/2021	2/1/2021	14
Program resource analysis module prototype	Milestone	Anastasios	100%	2/8/2021	3/25/2021	4/5/2021	2/22/2021	45
Program economic analysis module prototype	Milestone	Anastasios	100%	12/7/2020	3/15/2021	4/5/2021	3/8/2021	98
Plots displays figures for output	Task	Anastasios	100%	3/15/2021	3/29/2021	4/5/2021	3/22/2021	14
Program solution module prototype	Milestone	Justin	100%	3/29/2021	4/11/2021	4/5/2021	4/5/2021	13
Integrate the program (4 modules run for a year)	Milestone	Anastasios & Magaly	100%	3/29/2021	4/12/2021	4/5/2021		14
Evaluate prototype to see if design meets specifications	Task	Anastasios, Magaly Jessie & Justin	100%	4/11/2021	4/18/2021	4/12/2021	4/12/2021	7
Refine desing if neccesary	Task	Anastasios, Magaly Jessie & Justin	100%	4/11/2021	4/20/2021	4/12/2021	4/26/2021	9
~~Write main program & test main program to run for 'n' years~~	~~Milestone~~	~~Jessie~~		~~4/4/2021~~	~~4/19/2021~~	~~4/19/2021~~		~~15~~
Final program	Milestone	Anastasios, Magaly Jessie & Justin	100%	4/11/2021	4/21/2021	4/19/2021	4/27/2021	10
user manual Guide: source of data, methodology, issues & constraints and control methodology.	Milestone	Justin	100%	1/25/2021	4/20/2021		4/19/2021	85

Figure 10: Gantt Chart Comparison

Discussion of EC2000 Criterion 4 Considerations

r) Critically observes surroundings to recognize opportunity. -- Much of the project was defined in the previous semester. There wasn't that much need to identify new pieces as time went on, except perhaps to depthen a given module, or tie them together in a new way, as when GUI modifications were made. We attempted a GitHub implementation that didn't take off.
s) Explores multiple solution paths. -- We were able to be flexible in trying different ways to route the code through the modules.
t) Gathers data to support and refute ideas. -- If there was any data gathering it was in the first semester. We were more concerned with data gathering when the project was less defined; here we were, again, just trying to make it work.
u) Suspends initial judgment on new ideas. -- We may not have been encountering enough new ideas for this to be truly applicable. A lot of what we were doing was fleshing out pre-existing concepts, again.
v) Observes trends about the changing world with a future-focused orientation / perspective. -- The project was more or less based around this to begin with. We recognized that FERC was changing and pursued a path that would help Tri-State deal with that change.
w) Collects feedback and data from many customers and customer segments. -- We really only had one customer we were getting feedback from. We felt comfortable paying attention to the one that commissioned the project.
x) Applies technical skills / knowledge to the development of a technology / project. -- This one was a constant learning process. We were learning new things all of the time, especially our main coders.
y) Modifies an idea / product based on feedback. -- We did iterate the GUI on that basis, and the Excel outputs, though it was based on internal idea iteration more than feedback per se.
z) Focuses on understanding the value proposition of a discovery. -- Anastasios was pretty wholly concentrated on what he'd created with the second derivative thing for Module 3. It was, it seemed, a very well-thought-through piece that was very specific to Tri-State's needs. Some of the other stuff we did, such as the GUI iteration, was a little more like iteration than like careful discovery, but it too was a discovery process.
aa) Describes how a discovery could be scaled and / or sustained, using elements such as revenue streams, key partners, costs, and key resources. -- This was technically part of the project itself, as far as doing economic analyses that scale and can be

programmed in. However it's not like we were thinking about how to scale our project across a whole organization or beyond to other organizations, or even beyond just our one isolated task; this was a very specific program for a very specific niche.

bb) Defines a market and market opportunities. -- FERC defined this for us and we just followed suit. More technically speaking Tristate had their opportunity and we happened to pass by and see that they needed help with it. Moreover, though, the project itself is very much about market definition--it's really about how to most precisely define it so you're not wasting energy for a new deployment of a product.

cc) Engages in actions with the understanding that they have the potential to lead to both gains and losses. -- We never really spent that much time wondering if a given piece of work was perhaps not as integral to the project as we thought, which may have duplicated some work for the GUI and Module 3, as well as the GitHub implementation. A lot of what we had was used and built on without being wasted but large portions of code definitely were. There was not as much thinking about that as there could have been.

dd) Articulates the idea to diverse audiences. -- Not a whole lot of this. We did what we could to make what we wrote legible and clear, for instance to random Tri-state employees, but things get technical very quick. It does NOT hurt to have a technical background in grid management to understand this project. We proceeded on that kind of assumption, that this work was going to have to dig into very technical details that might be hard to explain outside of that.

ee) Persuades why a discovery adds value from multiple perspectives (Technological, societal, financial, environmental, etc). -- We proceeded with the project with the general understanding that we were helping a utility make better use of solar and wind, and make it more likely they'll deploy it, finding it to be worthwhile financially. In that sense we were considering it along multiple vectors.

ff) Understands how elements of an ecosystem are connected. -- Curtailments, weather, load growth--these were all elements that were considered in tandem.

gg) Identifies and works with individuals with complementary skill sets, expertise, etc. -- We were sure to do this among our team, other than that there really wasn't much outreach. But Jessie had his utility background, Magaly had her planning skills and general reliability, Anastasios had his sheer enthusiasm for the project that carried much of it through, and Justin was able to provide some backup on all fronts.