GRID\_MEND user manual

Table of Contents

[Overview 1](#_Toc459196469)

[Navigating the Interface 1](#_Toc459196470)

[Getting Started 1](#_Toc459196471)

[Simple Dispatch Test 1](#_Toc459196472)

[Create New Generator Set 1](#_Toc459196473)

[Create Demand Profile 1](#_Toc459196474)

[Develop Performance Map for Generator Set 1](#_Toc459196475)

[Test Generator Dispatch 2](#_Toc459196476)

[Launch GRID\_MEND Interface 2](#_Toc459196477)

[Start New Project 2](#_Toc459196478)

[Running GRID\_MEND 4](#_Toc459196479)

[Component List 5](#_Toc459196480)

[Estimated Costs 5](#_Toc459196481)

[Optimization Options 6](#_Toc459196482)

[Virtually Dispatch 6](#_Toc459196483)

[Launch Real-Time Dispatch 7](#_Toc459196484)

[Data Graph 7](#_Toc459196485)

[Dispatch Graph 7](#_Toc459196486)

[Component Models 7](#_Toc459196487)

[Utility 7](#_Toc459196488)

[Generators 8](#_Toc459196489)

[Renewables 8](#_Toc459196490)

[Electric Storage 8](#_Toc459196491)

[Thermal Storage 8](#_Toc459196492)

[HVAC 9](#_Toc459196493)

[Code Development 9](#_Toc459196494)

[Forecasting 9](#_Toc459196495)

[Predictive Control 11](#_Toc459196496)

[Storage/Self-Discharge 11](#_Toc459196497)

[Dispatch 11](#_Toc459196498)

[Glossary of Variables 11](#_Toc459196499)

[Appendix A: Interface Flow Diagrams A1](#_Toc459196500)

[1. GRID\_MEND Interface A1](#_Toc459196501)

[a) Opening GRID\_MEND A1](#_Toc459196502)

[b) Running Grid\_Mend A2](#_Toc459196503)

[c) Component List A2](#_Toc459196504)

[d) Optimization Options A3](#_Toc459196505)

[e) Virtually Dispatch A3](#_Toc459196506)

[f) Launch Real-Time Control A3](#_Toc459196507)

[g) Data Graph A4](#_Toc459196508)

[h) Dispatch Graph A4](#_Toc459196509)

[2. Component Edit/Add Interface A5](#_Toc459196510)

[a) Component List A5](#_Toc459196511)

[b) Add/Modify Utility A5](#_Toc459196512)

[c) Edit Utility A6](#_Toc459196513)

[d) Add/Edit Generator A7](#_Toc459196514)

[e) Add/Edit Renewable A8](#_Toc459196515)

[f) Add/Edit Electric Storage A9](#_Toc459196516)

[g) Add/Edit Thermal Storage A10](#_Toc459196517)

[h) Add/Edit HVAC A11](#_Toc459196518)

# Overview

GRID\_MEND is a system that was developed for optimal control of distributed generation resources.

# Navigating the Interface

## Getting Started

Open \_\_\_\_(MATLAB/Python/browser?) and select GRID\_MEND to run. A window will pop up with two options: Simple Dispatch Test or Launch GRID\_MEND Interface. Real-time dispatch can only be done through the GRID\_MEND Interface.

The Simple Dispatch Test can be used to create new generator sets or demand profiles. It can also be used to map the optimal performance of a generator set over a demand range, or to test the dispatch of a generator set for a set demand profile not in real time.

Launching the GRID\_MEND Interface is used to dispatch the components of a given grid according to localized demand and weather patterns. The grid, demand profiles, and weather data can either be loaded from a preset list or they can be created based on weather and building demand data for a given location and selected building characteristics. These new demand profiles are then matched to a custom set of generation components.

## Simple Dispatch Test

If the Simple Dispatch Test option is selected, four sub-ptions are available: Create New Generator Set, Create Demand profile, Develop Performance Map for Generator Set, and Test Generator Dispatch.

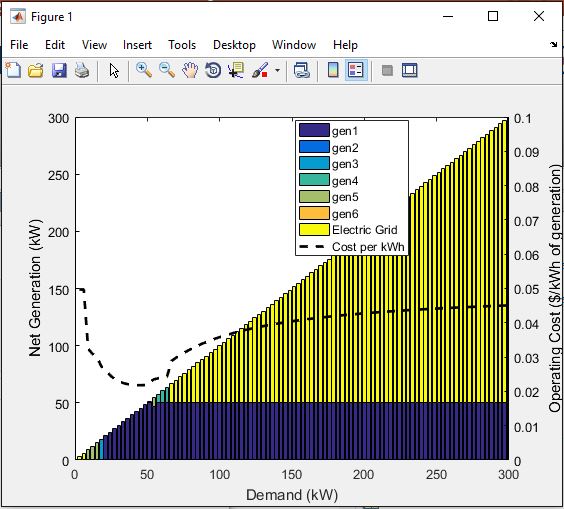
### Create New Generator Set

Creating a new generator set allows for the selection of different components from a pre-loaded list to create a new set that is stored under the file Plant/GenSets.

### Create Demand Profile

Creating a new demand profile allows the user to set \_\_\_\_\_\_\_\_\_\_\_\_\_\_ and is stored under \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Develop Performance Map for Generator Set

Developing a performance map will lead the user through a series of questions specifying which generator set to use and the characteristics of the utility/demand profiles. The program then plots the optimal generation level from each generator at each demand step. There will be a dashed-line function representing the cost/kWh with the optimal generator schedule displayed behind it. By default, the legend will appear in the top right, but can be moved around the graph. Similarly, other modifications can be made to the graph, if the user prefers, with the buttons across the top.

## Test Generator Dispatch

This option leads the user through a series of questions specifying which generator set to use, which utility to use, and the structure of the dispatch time. Once all of the specifications have been input, the program produces a plot of the dispatch of the generators optimized for cost, assuming that the demand profile is exactly as predicted over the horizon. This dispatch does not operate in real time; it is only a test run to show what the dispatch would look like over one horizon if the demand profile is exactly as predicted.

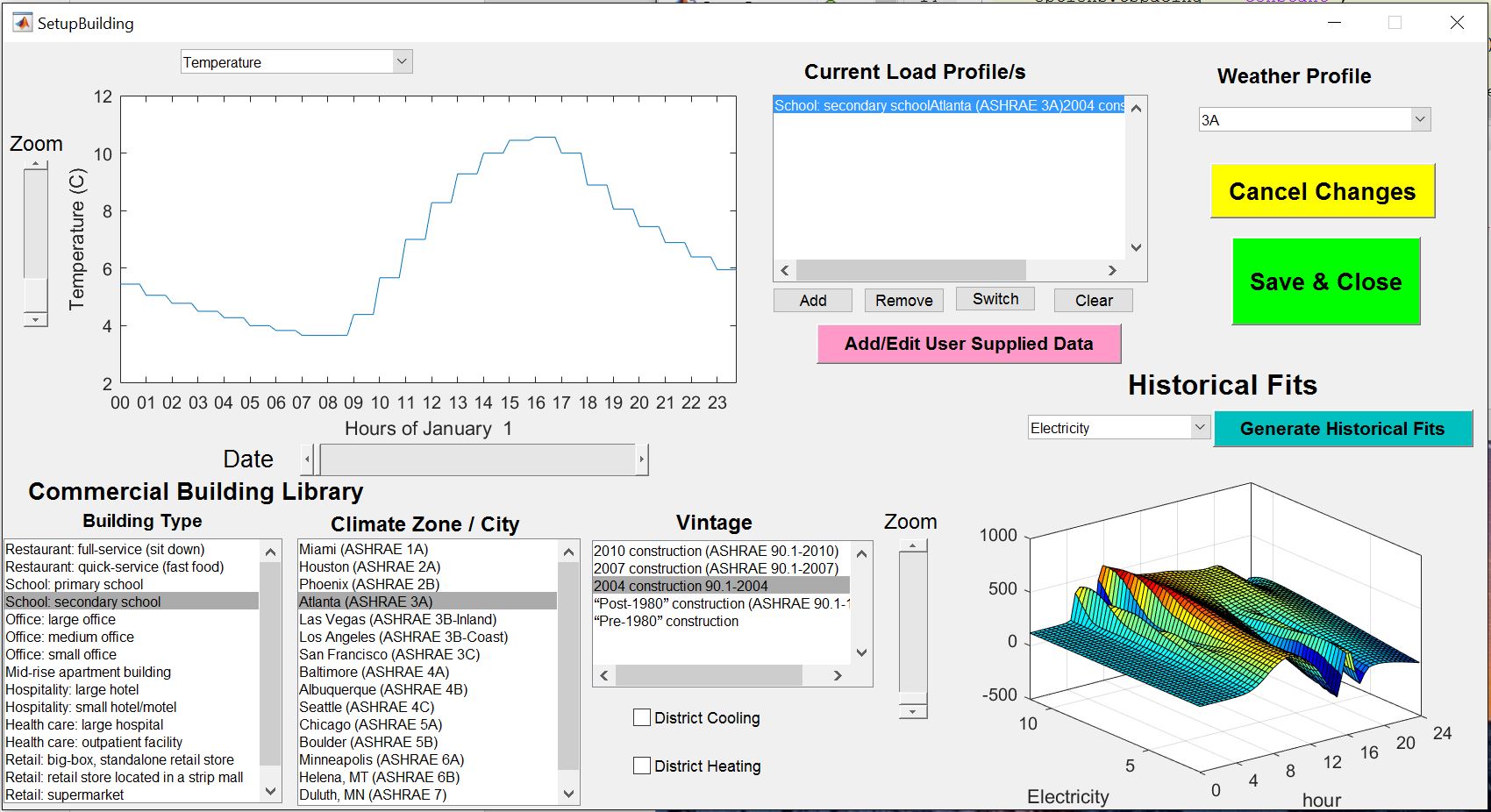
## Launch GRID\_MEND Interface

The GRID\_MEND Interface is the tool that is used for real-time or virtually simulated hierarchal model predicted control.

The user can either select the microgrid from a preloaded list or can generate their own microgrid setup by selecting Start New Project.

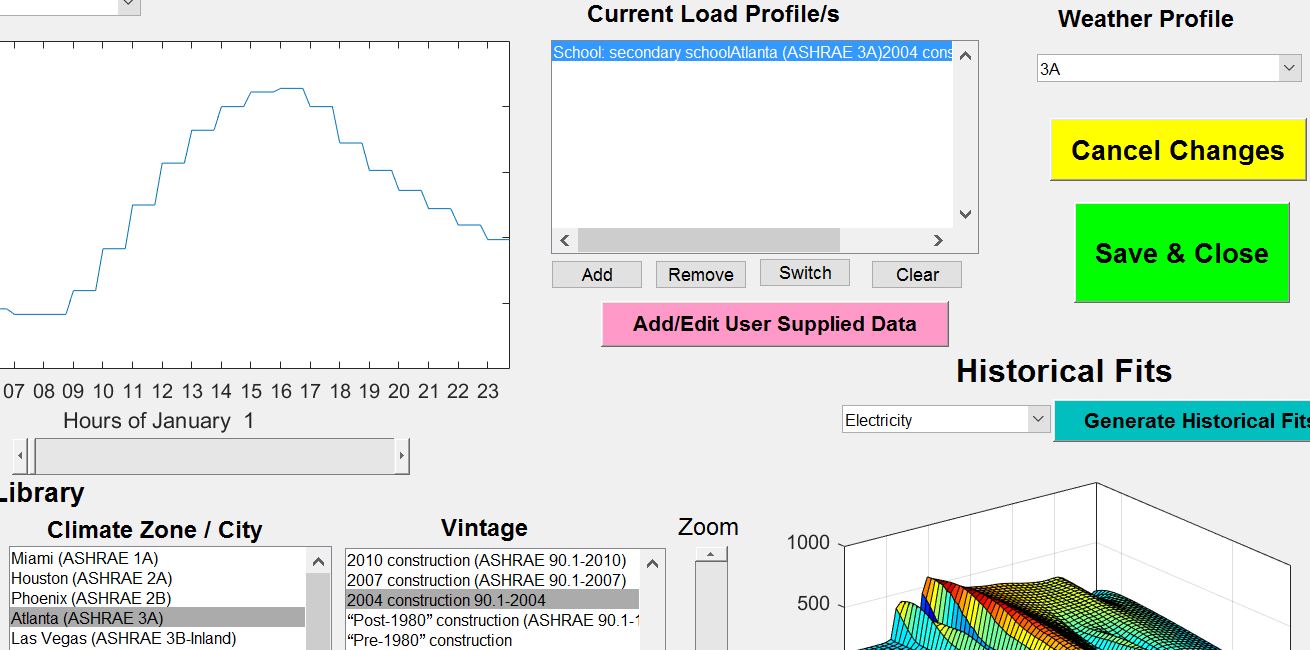
### Start New Project

If Start New Project is selected the user can either load data files characterizing the grid, or generate the data from built-in building models. A new interface will open up if the user chooses to generate the data. Buildings can be added to the microgrid by highlighting the building type, climate zone/city, and vintage and selecting Add under ‘Current Load Profile/s.’ To create data for heating and cooling demands, the District Cooling and District Heating boxes must be checked.

Figure : New micro-grid data can be added by selecting all the buildings included in the microgrid. In this example, a secondary school in Atlanta has been added to the current load profile. 

For each selected city, it is necessary that the corresponding ‘Weather Profile’ in the top right of the interface is selected as well. For example, if Atlanta is the desired city, the climate zone is ‘Atlanta (ASHRAE 3A)’. This means that 3A is the climate zone that must match the weather profile. After selecting the weather profile from the dropdown list, ‘Generate Historical Fits’ can be selected to create demand profiles.

Historical fits are available for temperature, electricity, heating, or all three. If temperature is selected, the user will be prompted to choose the desired month of to be displayed (all months can be displayed simultaneously as well). The temperature fit is line of temperature (in Celsius) vs time (in hours), where each line represents a month. If electricity, heating or all are selected, a 3D surface fit each month is created; the displayed surface will depend on the month and whether it is a weekend/holiday or weekday. All fits must be made before the dispatch can be run for the plant. For more information on how fits are made, see the Forecasting section of Code Development.

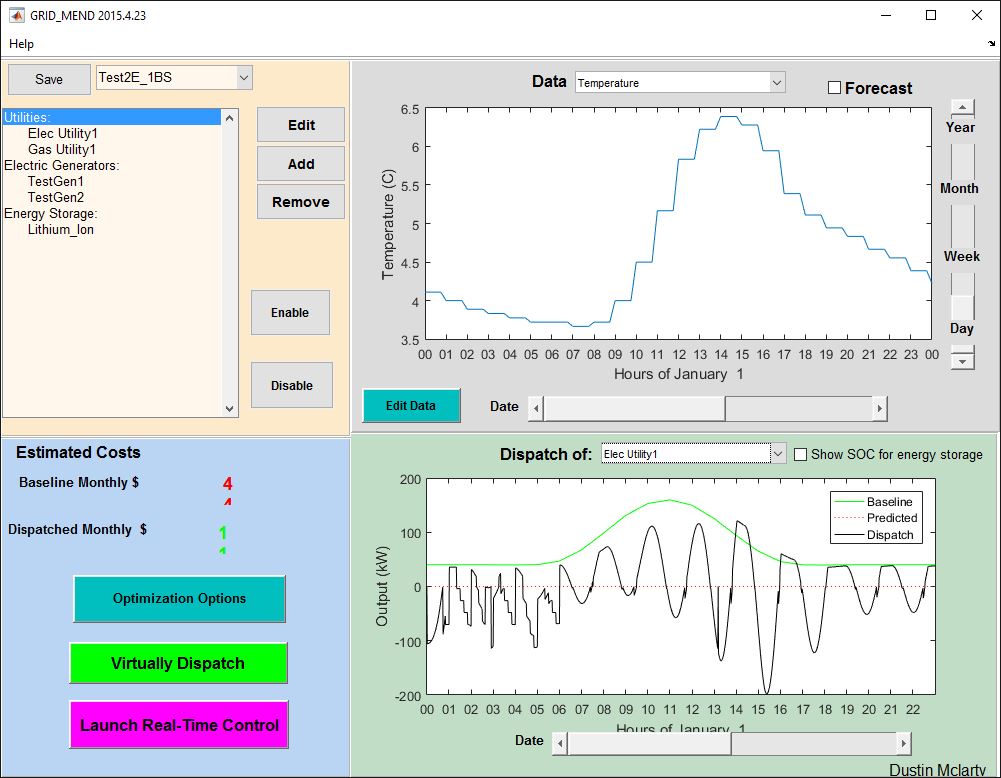
Figure : The correct weather profile has been selected for this city and the historical temperature and demand fits have been generated. 

Selecting Save and Close will save this profile for use in the dispatching tool. After the profile is loaded, the user must define how many of each component (i.e. Utilities, Generators, Heaters, Renewables, etc.) exists as well as other specifications that further describe each component within the plant. Once the microgrid is fully characterized, the main GRID\_MEND interface can be opened with the specified microgrid.

## Running GRID\_MEND

The interface must be launched for real time distributed generation dispatching. It is recommended that the real time dispatch be used as an advisory resource for generator dispatch, rather than being used as the dispatch controller. When the interface first appears, the left hand side will contain a list of the components (such as generators and batteries) included in the given microgrid. The microgrid can be changed at any time with the dropdown menu above the list. Below the list is the estimated costs (baseline and dispatched) followed by the buttons leading to ‘Optimization Options’, ‘Virtually Dispatch’ and ‘Launch Real-Time Control’. To the right of these functions are two graphics titled ‘Data’ and ‘Dispatch of:’ that will display additional information about the microgrid. The ‘Data’ graph will show temperature or demand profiles. The ‘Dispatch of:’ graph is initially blank but will show the output from each component after the dispatch is run.

Figure :The main GRID\_MEND interface from left to right: Component List (A), Estimated Costs (B), Optimization Options (C), Virtually Dispatch (D), Launch Real-Time Controls (E), Data graph (F), and Dispatch graph (G).



A

B

C

D

E

F

G

### Component List

The component list found on the left of the main interface can be changed by adding new components, removing components, or editing the characteristic models of the current components. Editing a singular member of the plant can be done either by highlighting and selecting ‘Edit’ or by right clicking the desired element. For more details on editing the components see section ‘Component Models’. The current plant can be changed by selecting another option from the dropdown list directly above the component list. Selecting Save in the top left corner will save any changes that are made to the microgrid components. Components in a microgrid can be disabled by highlighting them and selecting Disable. This removes the component from the optimization without removing it from the microgrid’s component list. To re-enable a component, highlight the disabled component and select Enable.

### Estimated Costs

Initially blank, this part of the interface will provide the user with a monthly baseline cost as well as a monthly dispatched cost. The baseline monthly cost refers to the cost that derives from updating at a single moment in time. This cost can be compared to the dispatched cost, that is based on predicting along a horizon and adjusting appropriately, instead of just to the next timestep.

### Optimization Options

Selecting Optimization Options allows the user to characterize the optimization. A new window opens with the options of a fast simulation or a simulation that is run at real time. If a Fast Simulation is selected, then the ratio of simulation time to real time must be specified; this ratio can be ignored if running in real time. By changing the ratio at which the simulations run in relation to real time, this variable scales the capacity at which a storage device performs. The power output for the storage devices is calculated to be the change in state of charge multiplied by the amount of time it took to make that change; since the time is now running at a faster/slower speed, the capacity must be scaled to account for this change. The ‘Meet heat demand within \_\_ hr’ option lets the user designate any delay that might occur while trying to meet demand. This heat demand tolerance allows for the heat demand to be met without the inclusion of district heating. It also helps prevent CHP (combined heat and power) generators from being controlled by heat demand instead of electric demand.

#### Optimized Timestep Resolution

The Optimized Timestep Resolution is the frequency at which the scheduled generator dispatch is optimized within one horizon; it can be constant, have a logarithmic spacing, or have manually input spacing. If the user opts to manually input the spacing, a new window will appear once ‘OK’ is selected. This new window allows the user to choose a vector of times within one horizon for each timestep.

#### Chiller Optimization

The Chiller Optimization options will dictate whether the optimization will be run before electric dispatch or concurrently with the electric dispatch. Running the optimization before the dispatch will add the power used to run the chillers to the electric demand. An assumed fixed chiller efficiency will be required if the user chooses to run them concurrently.

#### Dispatch Parameters

The Dispatch Parameters govern the various modifications of the timesteps in the dispatch. The simulation period dictates the length of the simulation. The initial step size is the duration of time at which the dispatch is optimized. The dispatch horizon determines how far into the future the dispatch predicts demand and generator dispatch. The set-point frequency regulates how often (within an optimization step) the online generators are given a new optimized set-point to meet demand. The MPC frequency is how often the grid balance is checked and adjusted to make sure demands are met. Selecting the option in the bottom corner ‘CCHP Generators can dump excess heat’ allows for the generators to produce more heat than is needed to meet the demand and dump the excess. This option helps prevent generators from being limited by the heat demand.

### Virtually Dispatch

Once Virtually Dispatch is selected, a new window will launch to select either manually or automatically determined initial conditions. If ‘Manually Specify Initial Conditions’ is chosen, the user must input the conditions to then run the dispatch. The initial condition can either be 0, or any value between the lower and upper bounds; if an initial condition between 0 and the lower bound is designated, the dispatch will experience an error. If ‘Automatically Determine Initial Conditions’ is selected, the dispatch will immediately run. After initial conditions are defined, the dispatch will begin and display the generator usage over the allotted horizon.

Within the window for the dispatch, the legend will be placed in the top right by default. This legend can be moved by clicking and dragging, or completely removed/inserted by selecting the ‘Insert Legend’ icon above the graph. The resulting dispatch window can be edited in a number of other ways as well, through the tools provided above the graph. These include changing colors, zooming, panning, and editing axes.

### Launch Real-Time Dispatch

### Data Graph

On the top right of the GRID\_MEND interface is a Data graph with a dropdown menu. From this menu, either temperature or energy demand can be selected. Once selected, the user has the ability to ‘Forecast’ for both of these options, by checking the box next to the dropdown. The forecast will lay on top of the already existing graph. To the right of the graph are zoom sliders that allow the user to adjust the time axis from days to years. Below the graph, the user can use the slider to scroll across a length of time. The building and climate data for the system can be adjusted through the ‘Edit Data’ button located on the bottom left of the graph. This will bring the user to the same screen that is viewed when new data is generated from built-in building models (See Section ‘Start New Project’).

### Dispatch Graph

Below the data graph exists another graph that is initially blank. Once the dispatch is run, the user can select a component from the grid, via dropdown menu, which will display a graph of the energy output as a function of time. The Dispatch graph will show the baseline use of each ‘generator’, as well as the predicted values and the actually performance of the dispatch. The user can also elect to view the state of charge of the energy storage with any graph by checking the box to the right of the dropdown.

## Component Models

Each component model has a corresponding flowchart to best navigate through the interface. All flowcharts are in Appendix A. As mentioned previously, components can be added/removed or edited from the GRID\_MEND main screen. Depending on which component is selected, new window(s) prompting the user for specifications will appear. Each component has default values that it fills each prompt, which must be updated by the user to better reflect their actual plant.

### Utility

When editing an electric utility, the user will be faced with 4 options to adjust the way the utility is being perceived by the program. The first option is a ‘Peak/Off-Peak Rate’ which, when selected, presents the user with 4 inputs that further describe the rates. The user can also choose to load 15 minutes of data or load an existing utility, as well as modify the structure.

When modifying or setting up an electric utility, you will be prompted to further describe the unit, such as naming the unit and identifying key features. The Minimum Import Threshold refers to the minimum the utility can buy at the purchase rate. This number can be negative if selling back at the same purchase rate is possible. The Grid Sell-Back refers to the ability of the utility to sell back to the grid, and for what rate. The user can either select none, a percent of the tariffs, or a reversed meter for the grid sell-back. On the left of the interface are two identical tables; one for winter and one for summer. These tables identify what hour and day the utility experiences a peak (3), partial peak (2) and when its off-peak (1). Each of these three intervals is related to both the energy charge and the demand charge. The user can define the peaks however they see fit.

Gas utilities are not as adjustable; the user can either input a constant rate or load 15-minute data for the utility.

### Generators

The generator setup menu is applied to both electric and CHP generators, as well as a variety of other components that behave similarly to generators in the dispatch. These other components include boilers, chillers, and heaters. When modifying a generator, a window will appear with several different aspects to adjust to better define what role the “generator” plays in the microgrid. On the edit screen, the user has the opportunity to ‘Specify Communication Ports’. This feature will allow the user to further describe the communication from the generator to a desired controller. The output type in combination with the energy source are used to best identify what exactly the component is within the microgrid. Several output types can be selected, but only one energy source can be chosen. For example, a CHP generator would output electricity and heat, while using natural gas as the energy source. To better navigate through these options, the flowchart for generators could be utilized.

### Renewables

The user has the choice of either wind or solar to be placed under the renewable category. Renewables offset the demand of the generators, as they provide energy to the plant.

When editing solar, several specifications such as location, size, angle, type and tracking must be input to best identify the contribution the resource is making to the microgrid. To the right of the solar setup interface is a DC-AC Conversion chart for all important values associated with the solar panel.

By selecting wind, the user can adjust \_\_\_

### Electric Storage

Electric storage refers to any battery within the microgrid, such as Lithium Ion or Sodium Nickle. While the battery type can vary, the setup menu will stay the same. The editing of battery storage allows the user to best characterize the specifications of the battery. These specifications include size, charge/discharge characteristics, as well as cell characteristics. The relationship between state of charge and voltage is also graphed on the right side. Similar to the generator, the electric storage also has the option to ‘Specify Communication Ports’. The ‘Self Discharge Rate’ considers the innate loss of energy storage associated with batteries.

### Thermal Storage

Thermal storage refers to either hot or cold water storage. For thermal storage, several different components of size must be considered, such as volume and energy. Other things to be identified are efficiencies and limits. At the top right of the interface, under the title ‘Chiller/Heat Capture Limits’ the maximums for capture and discharge are calculated by the program, whereas user-defined limits are directly below. Thermal storage units also allow the user to ‘Specify Communication Ports’. Just like a battery, thermal storage also has ‘Self Discharge Rate’ to be considered as well.

### HVAC

The Heating, Ventilation, and Air Conditioning (HVAC) system for the microgrid can be classified as one of three different types: ‘Smart Air Conditioning Only’, ‘Smart Heating Only’ or ‘Smart AC and Heat’. A warning is displayed at the bottom right of the setup menu to remind the user that a chiller and/or heater must be used to simulate the smart HVAC system in the grid. The size of the building the HVAC is working within needs to specified, as well as the amount of energy displaced by the system to better evaluate the way the HVAC operates as a component of the microgrid.

# Code Development

## Forecasting

### Temperature

Forecasting temperature is done by averaging the values from the prior day with the historical data for the current day and region. This average is then smoothed to produce the projected temperatures for the next 24 hours (Figure 5). To avoid any discontinuities in the weather pattern, the first forecasted temperature will always match the last actual temperature from the previous day; this provides the baseline for the smoothed temperature for the next horizon (Figure 6).

Figure : The temperature from yesterday has been averaged with the historical, then smoothed to form a prediction. This prediction is then compared to the actual temperature.

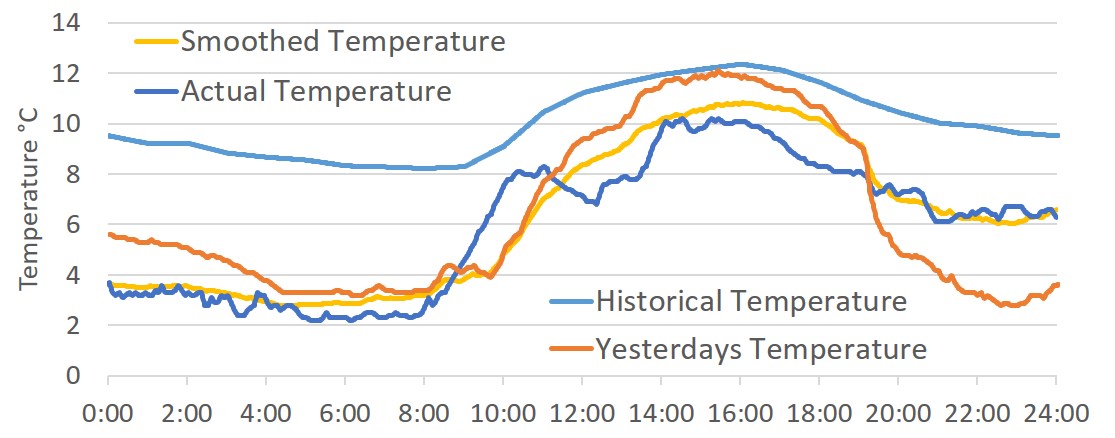
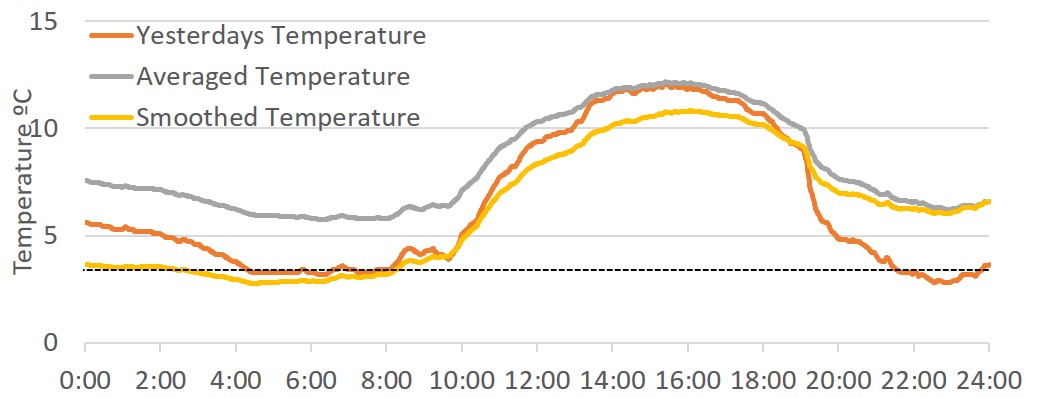


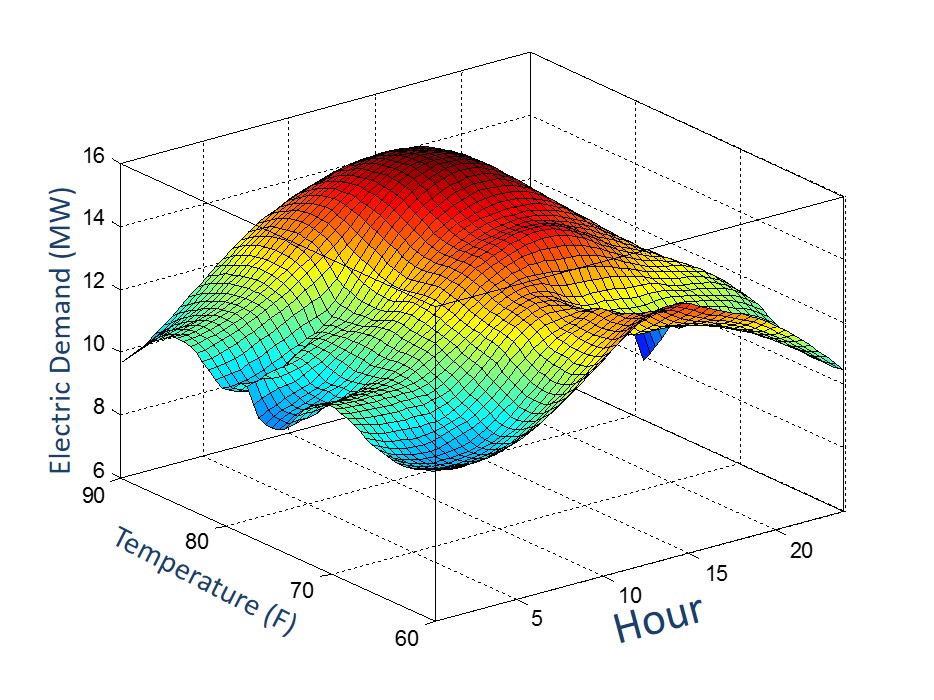
Figure : The last recorded temperature from yesterday is used to the determine the base for the smoothed temperature, which is then fit with the average of yesterday and the historical.



### Demand Profiles

The demand of each component in a generator set is dependent on the actual temperature, therefore forecasting is crucial to accurately predicting usage. In the same way temperature is forecasted, the loads for electricity, heating and cooling are calculated. An average from the previous day’s load and the historical load for the forecasted temperature are used to predict a surface fit for the load. Different surfaces are used for weekdays and weekends/holidays to ensure that the most accurate prediction is being made.

Figure :The surface fit made by forecasting electric load from averaging yesterday's and historical loads



### Power

For any dispatch with long timesteps (larger than 1 hour), power will be forecasted. This is done by looking at KWh over the last hour and predicting the power for the next hour, using left handed trapezoid integration.

## Predictive Control

To ideally predict demand along a horizon, GRID\_MEND uses a control method called Model Predictive Control (MPC). In essence, this technique effectively looks across a given horizon and updates the predicted demand after each optimization. MPC is used for controlling GRID\_MEND as it best emulates the human thought process; decisions are reevaluated as time progresses. A way of envisioning predictive control would be to think of hiking. Say you are hiking a 5-mile trail and all you can see is flat terrain; knowing that you can run 5 miles on a treadmill in a 50 minutes, you set your pace to a 10-minute mile. This speed might have been appropriate if this entire trail was flat, however you are approaching a mountain. Standard controllers would not see this mountain approaching, so you would have run all the way to the base of the mountain, tiring yourself out before the climb and all together getting a slower time by the ends of 5 miles. Using MPC methods, you would slow down once you first see the mountain to conserve energy for the impending climb; by slowing down for a few minutes, the total time for the hike will actually be shorter because you will be ready to climb the mountain.

## Storage/Self-Discharge

For every storage device, both electric and thermal, there is a value of self-discharge that must be accounted for. Self-discharge refers to the characteristic loss of stored energy over time. This is important to consider as all storage devices experience this loss in some degree. Thermal storage experiences self-discharge at a much greater rate than electric storage, so it is especially important that it is considered. In GRID\_MEND, self-discharge is understood to be a constant that is added to the overall demand of the storage. This constant is represented by the loss (in percent) multiplied by the upper bound of the device, then divided by the charge and discharge efficiencies.

## Dispatch

The dispatch is done through a number of steps, starting with a discontinuous cost curve for each generator. Each cost curve is then fit with 2 piece-wise concave curves; one with a zero intercept and one with a non-zero intercept. These piece-wise functions are used to map the prime distribution of generator outputs against demand, and thus determine the ideal schedule for the generators within the system.

# Glossary of Variables

Modularizing loading of generators:

Global variable Plant contains all information on the generators, demand, (we will add details of grid i.e. network constraints & losses here) and dispatch results.

Plant.Generator has all the information of each generator necessary to complete the GUI’s and construct the optimization matrices.

Generator.Type: defines what category the generator is in, electric, CHP, battery, chiller, storage, grid….

Generator.Name: gives the generator a name

Generator.Source: specifies what the input is, i.e. fuel, bio-gas, waste heat. \*\*It is important to note what can have a quadratic vs. linear relationship. If we relate the input of a generator to its fuel, and only give the fuel a cost, the cost-curve of the generator can only be linear or varying in time. This makes splitting up costs between different fuels i.e. natural gas, bio-fuel, difficult, but may work well for hydro-power or other cases. For electric generators with non-linear cost-curves it is better to convert to $ when building the optimization matrix.

Generator.Output: specifies what the generator generates, ie. Heat, electricity, cooling, hydrogen…, and gives the output in terms of energy efficiency as a function of capacity. \*note that storage devices i.e. batteries hot water tank, do not have an output. Their Type determines what is stored and hence what is output. This may need to be revisited if we consider storage devices that output two things (heat+electricity)

Generator.Size: generally the capacity is in kW for generators and kWh for storage devices

Generator.Enabled: whether the device is operational and should be included in the dispatch or not. \*\*\*\*\*This may need to be revisited as some different structure if we want to include planned or unplanned shutdowns in the optimization.

Generator.VariableStruct: This contains characteristics specific to this type of generator. This information is editable in the GUI and used to build the Generator.OpMatA and OpMatB structures.

Generator.OpMatA: This structure helps in the construction of the optimization matrices associated with the multi-time-step quadratic programming where it is unknown which generators are on or off, FitA, and is detailed below.

Generator.OpMatB: This structure helps in the construction of the optimization matrices associated with the multi-time-step quadratic programming where it is known which generators are on or off, FitB, and is similar to what is detailed for OpMatA below. The difference is that the number of states is likely reduced.

OpMatA.states: a string listing the states used to represent the generator i.e. x, y, & z. The states are each fields of OpMatA and contain the matrix values that should be associated with this state during the optimization.

OpMatA.cost: identifies the input cost used to convert input to $’s, so that the cost variables can be readily scaled with changing fuel costs. If the input is linked to a separate state for fuel consumption, its costs are zero and the value here does not do anything.

OpMatA.link.eq: a vector showing the values of each state in a row linking the states at a specific time-step. Paired with OpMat.A.link.beq.

OpMatA.link.ineq: same as above, but in the inequality matrix. A vector showing the values of each state in a row linking the states at a specific time-step. Paired with OpMat.A.link.b.

OpMatA.X.output.\_\_\_\_\_: Categories are same as Generator.Output (electricity, steam, heat, hydrogen…) \*\* note these are shortened to E, S, H, H2 …, if the field exists it has a value that should be associated with this state for the row relating generator outputs in this category to demand in this category.

OpMatA.X.Ramp: If this state has a ramping constraint its values are placed in Ramp.A and Ramp.b. Ramp.A can be a vector with two values if it is a single direction constraint, or a 2x2 matrix if it is constrained in ramping up & down.

OpMatA.X.H: The quadratic component of the cost, [] is interpreted as 0.

OpMatA.X.f: The quadratic component of the cost, [] is interpreted as 0.

OpMatA.X.ub: upper bound associated with this state.

OpMatA.X.lb: lower bound associated with this state.

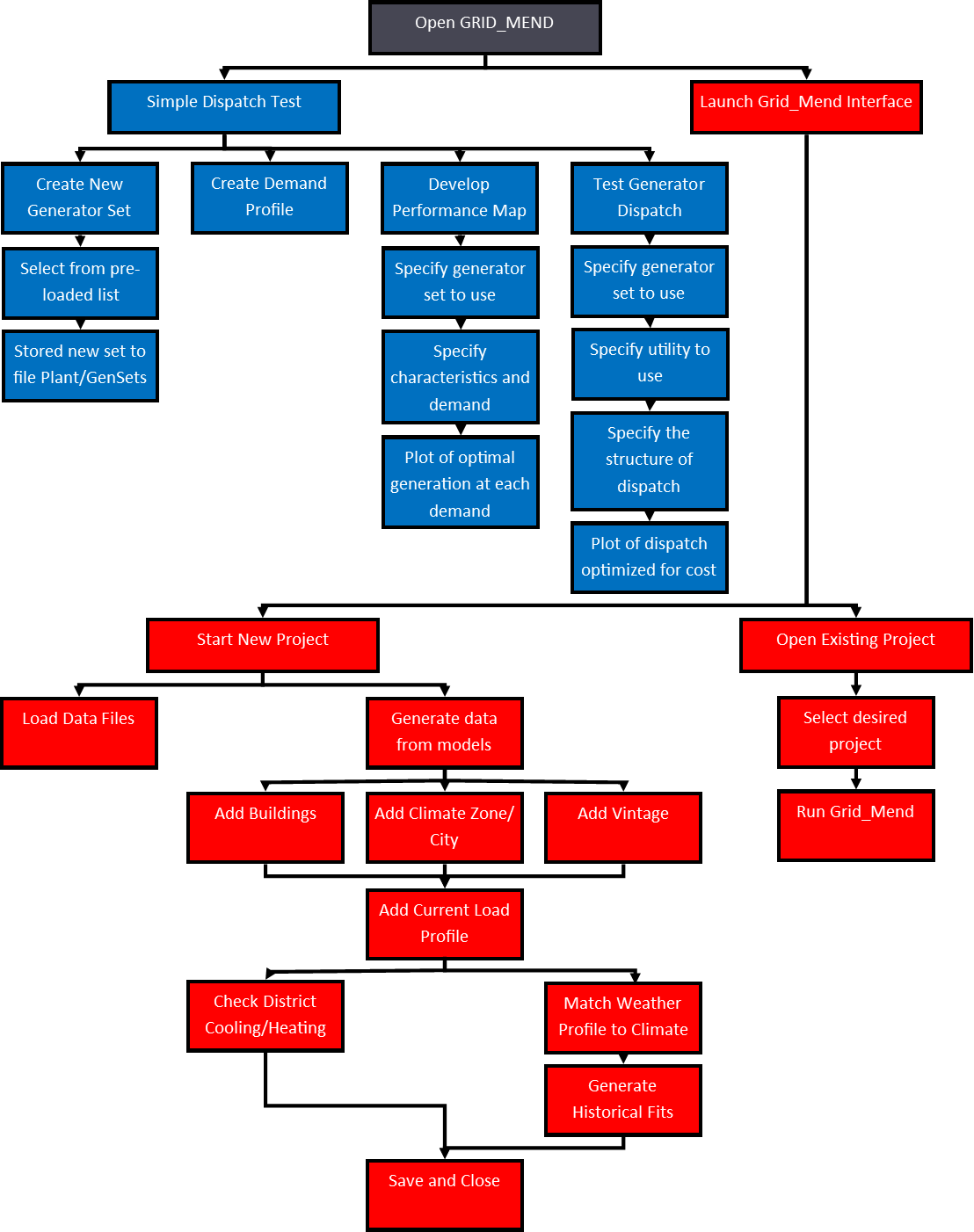
List of constraints:

|  |  |  |  |
| --- | --- | --- | --- |
| Type of output | Electrical output | OpMatA.X.output.electricity | [] or 1 |
|  | Heat output | OpMatA.X.output.heat | [] or 1 or Hratio |
|  | Cooling output | OpMatA.X.output.cooling | [] or 1 or Cratio |
| Ramp Rates | Ramp up & down | OpMatA.X.Ramp.A | [-1,1;1,-1;] |
|  |  | OpMatA.X.Ramp.b | [rampupvalue, (-rampdownvalue)] |
| State splitting | How do the different states relate | OpMatA.link.eq | For a gen [1,-1,-1]  For grid [1,-1]  For storage [1,-1/roundtrip efficiency] |
|  |  | OpMatA.link.beq | 0 |
| Boundaries | Upper bound | OpMatA.X.ub | Generators(i).Size, inf, usablesize |
|  | Lower Bound | OpMatA.X.lb | Usually 0 for OpMatA and LB for OpMatB |
| Charging Rates | Handled in Ramp |  |  |
| Depth of Discharge | Handled in lower bound |  |  |
| Self Discharge | Can either be a constant or ratio of state of charge |  |  |
| Buffer | ? |  |  |

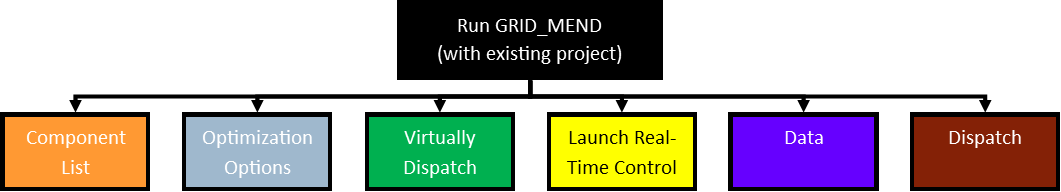
# Appendix A: Interface Flow Diagrams

## 1. GRID\_MEND Interface

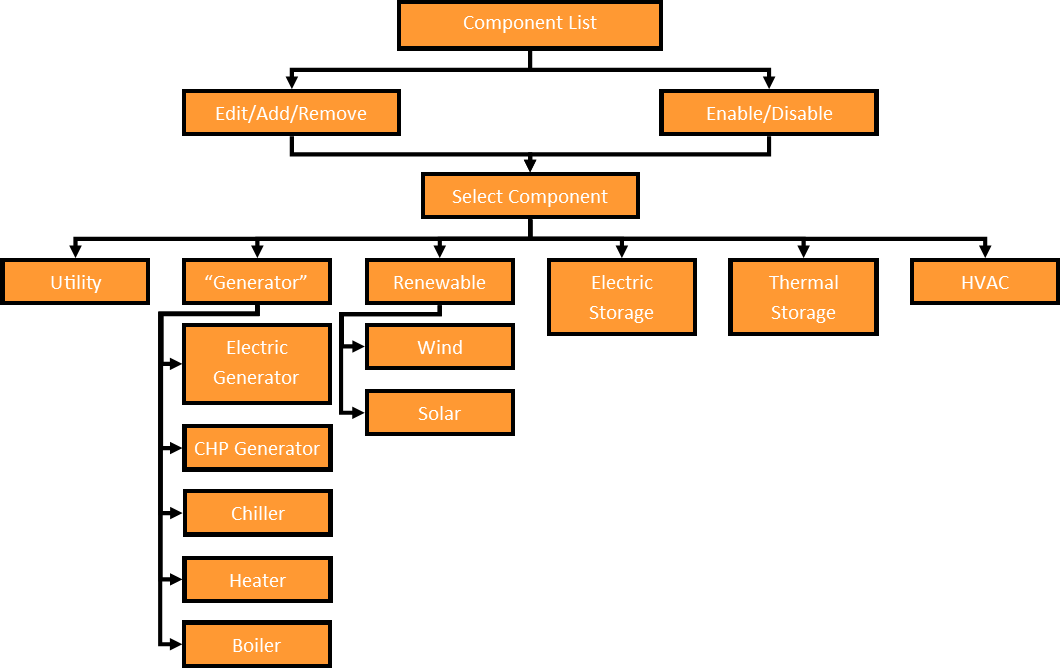
### a) Opening GRID\_MEND



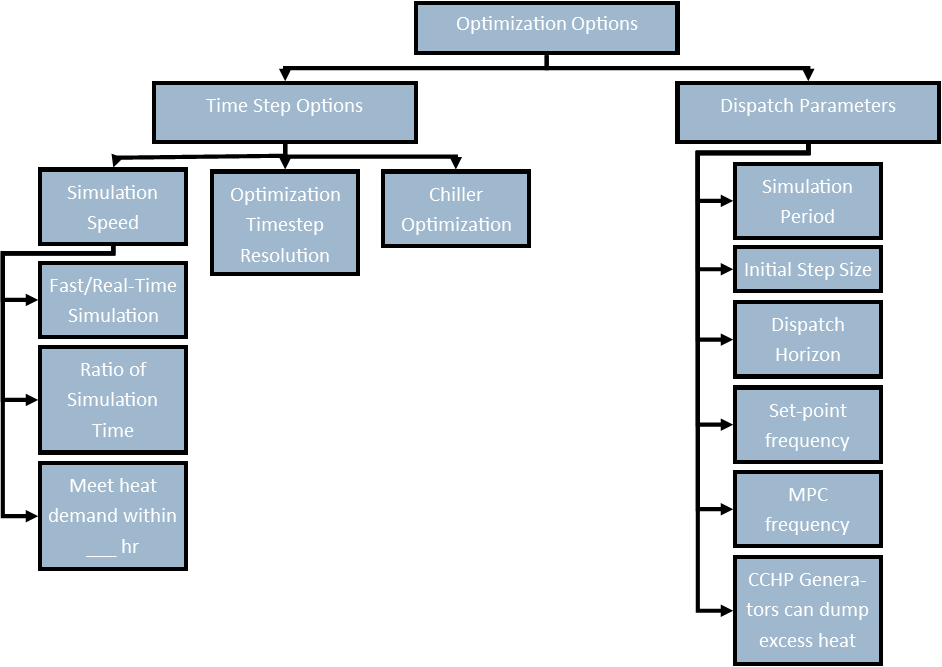
### b) Running Grid\_Mend



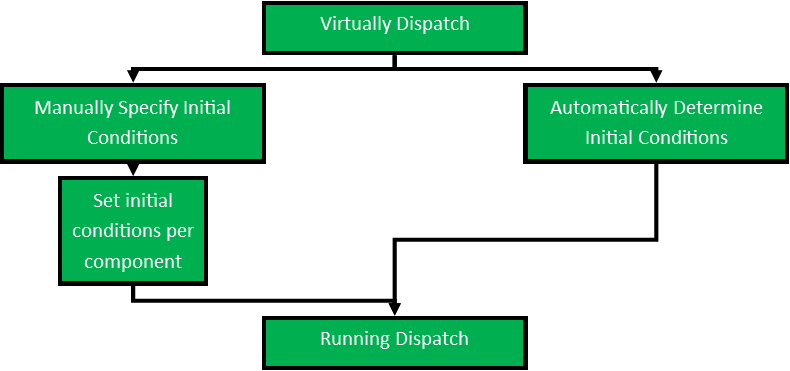
### c) Component List



### d) Optimization Options

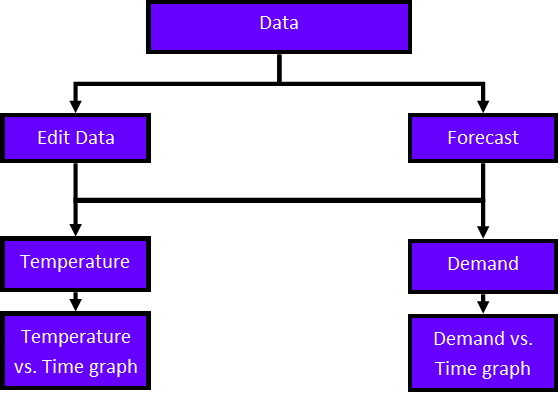


### e) Virtually Dispatch

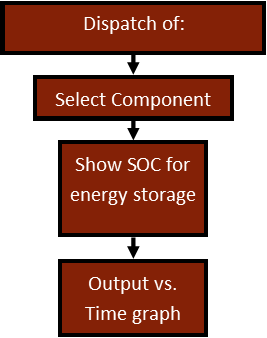


### f) Launch Real-Time Control

### g) Data Graph

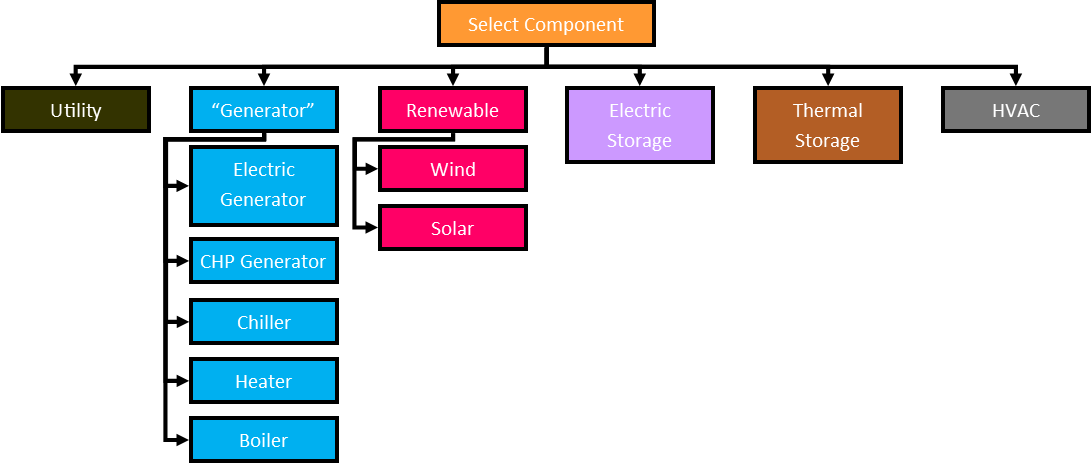


### h) Dispatch Graph

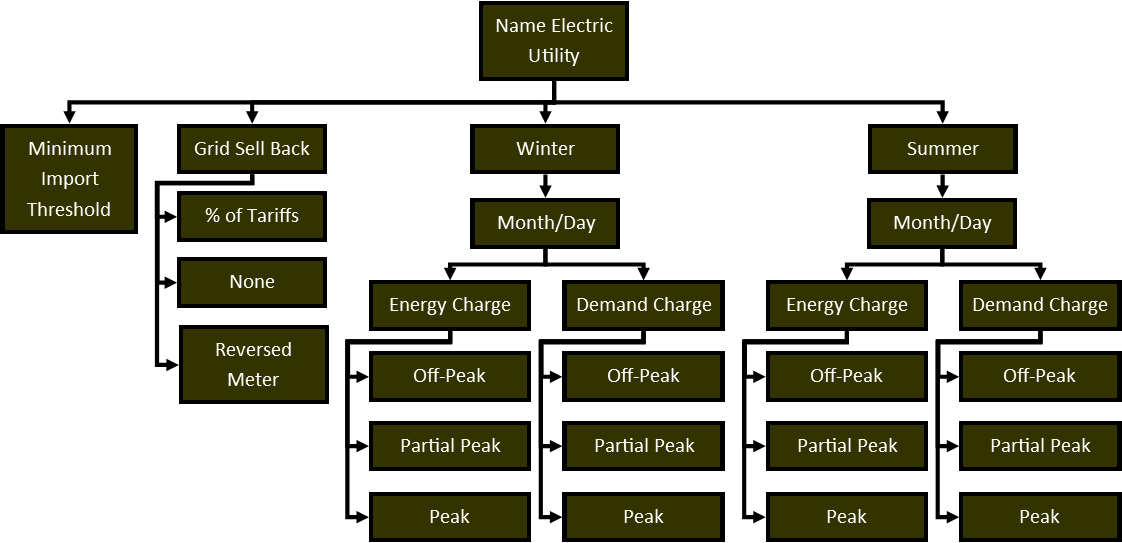


## 2. Component Edit/Add Interface

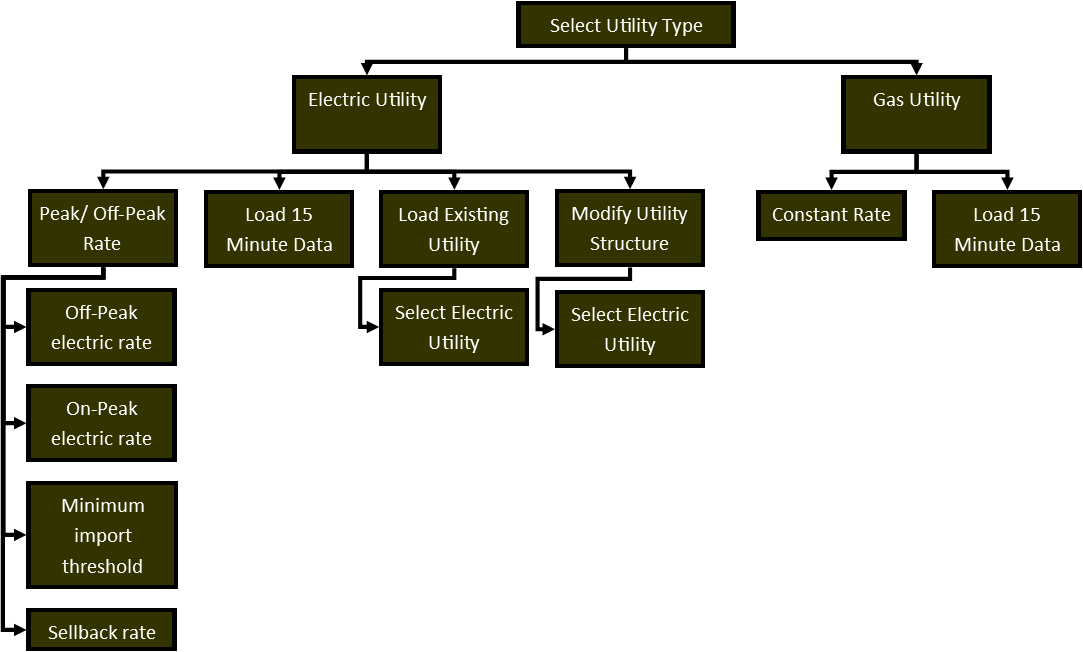
### a) Component List



### b) Add/Modify Utility



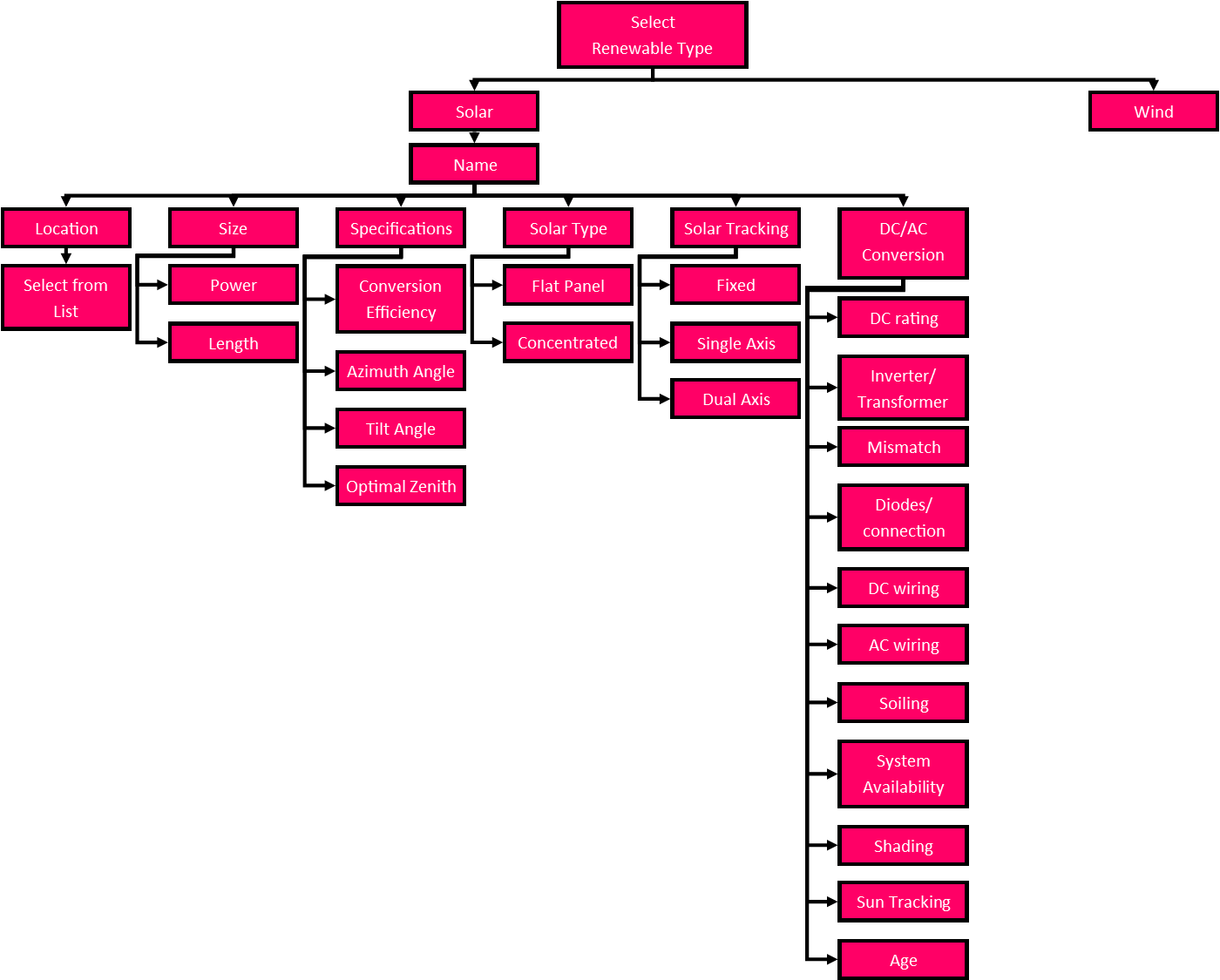
### c) Edit Utility



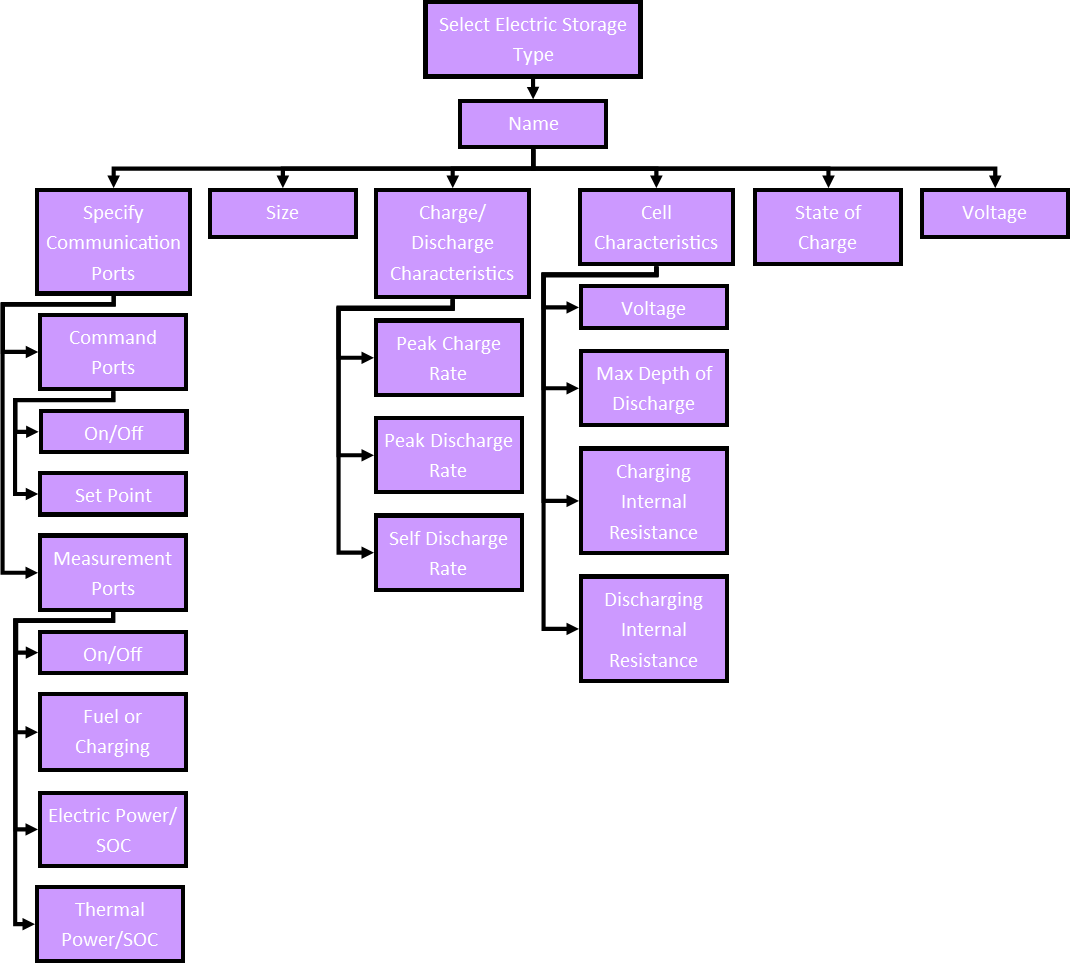
### d) Add/Edit Generator



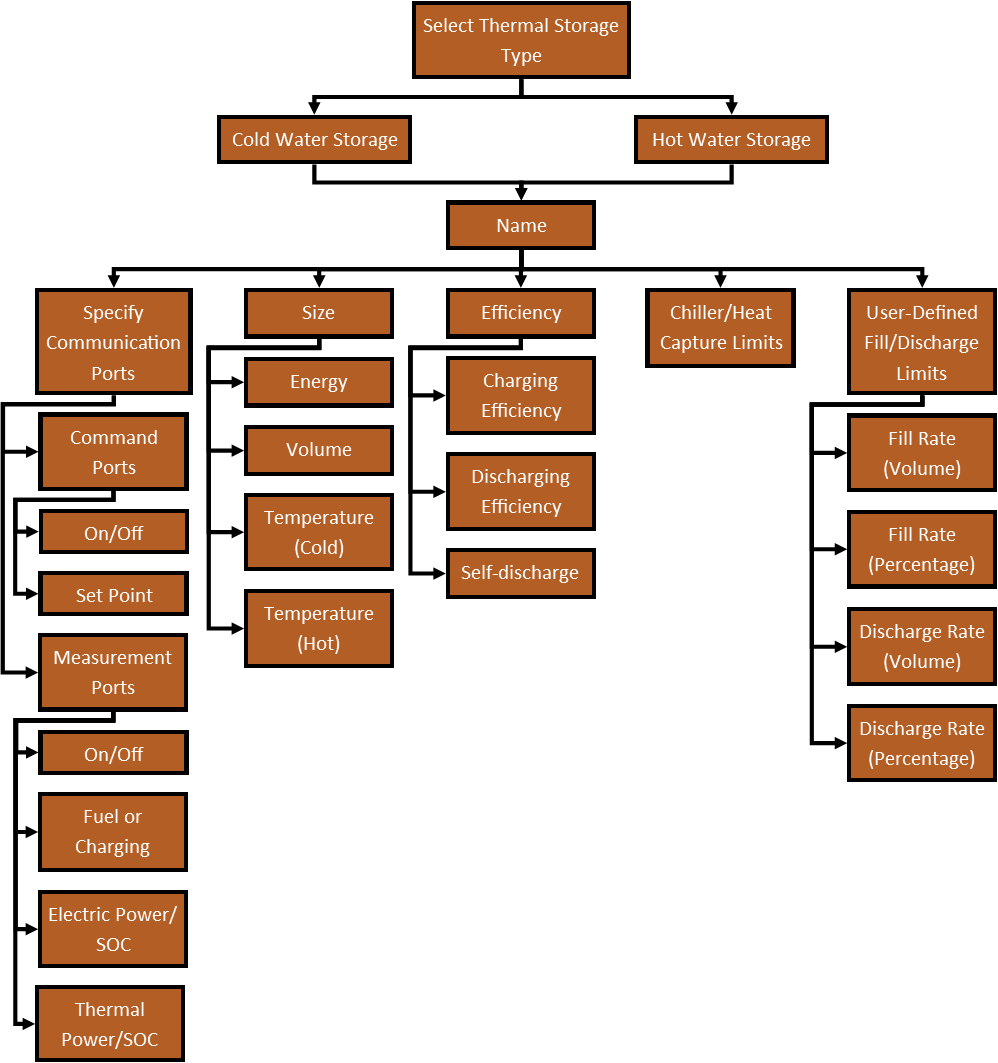
### e) Add/Edit Renewable



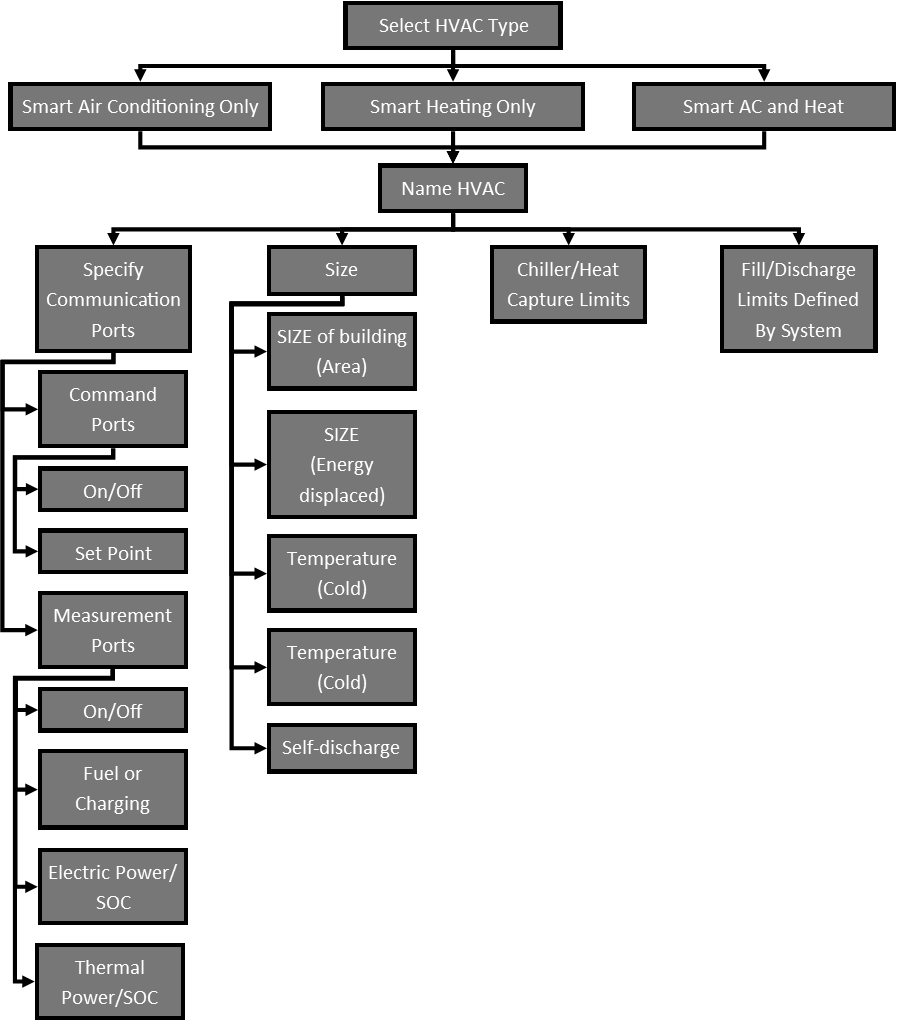
### f) Add/Edit Electric Storage



### g) Add/Edit Thermal Storage



### h) Add/Edit HVAC



Questions:

1. I have the two states for the storage charging or discharging, but I’m not sure what the other 2 states for storage are. The buffer
2. Do we still need the SSi model? See line 62 to 87 of loadGenerators

This may have been a placeholder for eventually calculating the ramp rate constraints as a function of the state-space model. Ramp rate could change depending on the current output. I shelved this idea for now.

SSi(n).A = A;

SSi(n).B = B;

SSi(n).C = C;

SSi(n).D = D;

%Use sim(SS) to calculate dX,

1. How is the cost of purchasing and selling power to and from the grid found if we don’t have a time associated with the model yet? See line 147 of loadGenerator

F= [f c\_purchas c\_sell];

It is eventually overwritten when it builds the matrices and looks of the demands and costs as a function of time. What we should do now is put a 1 here. Then before building the matrices we forecast the cost of all utilities. Anything with that utility as an input gets its costs updated by scaling what is in OptMatA.X.f & OptMatA.X.H by the updated cost / what is in OptMatA.cost

Comments:

1. I haven’t really been able to modify multitimestepQP. The newer version is multitimestepQPM, but it hasn’t been changed to accommodate the new system of building matrixes.
2. The function loadGenerators (with an s) is the old function I kept for reference. The function loadGenerator (no s) is the new one I was working on.
3. This branch wont run because loadGenerators is the only function that has been modified to accommodate the new style.
4. I need to change how the cost terms for the storage are handled and change where the buffer is found.
5. I haven’t incorporated HVAC into this.