Efficient Allocation of Grid Energy Resources including Storage (EAGERS)

user manual

Economic dispatch for micro-grids presents a highly constrained non-linear optimization problem. A common issue faced by search or gradient based optimization algorithms is the discontinuity between a generator’s minimum operating condition and its off-line state, resulting in a non-continuous solution space with many local cost minima. This challenge is compounded by the dispatch flexibility afforded by energy storage technologies, which require simultaneous optimization over the energy storage horizon. This document outlines an open-source platform, EAGERS developed to address these challenges using complementary convex quadratic optimizations. The approach applies to grid-connected or islanded micro-grids comprised of any variety of electric or combined heat and power generators, electric chillers, heaters, and all varieties of energy storage systems. It incorporates constraints for generator operating bounds, ramping limitations, and energy storage inefficiencies. Please refer to the appendix for a detailed explanation of the optimization strategy.

Table of Contents

[II. Overview 1](#_Toc466985280)

[III. Navigating the Interface 1](#_Toc466985281)

[i. Getting Started 1](#_Toc466985282)

[ii. Starting a New Project 1](#_Toc466985283)

[iii. Running EAGERS 3](#_Toc466985284)

[a. Component List 4](#_Toc466985285)

[b. Estimated Costs 4](#_Toc466985286)

[c. Optimization Options 5](#_Toc466985287)

[d. Virtually Dispatch 6](#_Toc466985288)

[e. Launch Real-Time Dispatch 6](#_Toc466985289)

[f. Data Graph 6](#_Toc466985290)

[g. Dispatch Graph 6](#_Toc466985291)

[iv. Component Models 6](#_Toc466985292)

[h. Utility 6](#_Toc466985293)

[i. Generators 7](#_Toc466985294)

[j. Renewables 7](#_Toc466985295)

[k. Electric Storage 7](#_Toc466985296)

[l. Thermal Storage 7](#_Toc466985297)

[m. HVAC 8](#_Toc466985298)

[IV. Code Development 8](#_Toc466985299)

[v. Forecasting 8](#_Toc466985300)

[n. Temperature 8](#_Toc466985301)

[o. Demand Profiles 9](#_Toc466985302)

[p. Power 10](#_Toc466985303)

[vi. Predictive Control 10](#_Toc466985304)

[vii. Storage/Self-Discharge 10](#_Toc466985305)

[viii. Dispatch 10](#_Toc466985306)

[V. Glossary of Variables 10](#_Toc466985307)

[VI. Interface Flow Diagrams 1](#_Toc466985308)

[ix. 1. EAGERS Interface 1](#_Toc466985309)

[a. Opening EAGERS 1](#_Toc466985310)

[b. Running EAGERS 2](#_Toc466985311)

[c. Component List 2](#_Toc466985312)

[d. Optimization Options 3](#_Toc466985313)

[e. Virtually Dispatch 3](#_Toc466985314)

[f. Launch Real-Time Control 3](#_Toc466985315)

[g. Data Graph 4](#_Toc466985316)

[h. Dispatch Graph 4](#_Toc466985317)

[2. Component Edit/Add Interface 5](#_Toc466985318)

[a. Component List 5](#_Toc466985319)

[b. Add/Modify Utility 5](#_Toc466985320)

[c. Edit Utility 6](#_Toc466985321)

[d. Add/Edit Generator 7](#_Toc466985322)

[e. Add/Edit Renewable 8](#_Toc466985323)

[f. Add/Edit Electric Storage 9](#_Toc466985324)

[Add/Edit Thermal Storage 10](#_Toc466985325)

[g. Add/Edit HVAC 11](#_Toc466985326)

[VII. Appendix 12](#_Toc466985327)

[x. Modifications to Dynamic Economic Dispatch 13](#_Toc466985328)

[xi. Piecewise Convex Quadratic Cost Fitting 14](#_Toc466985329)

[xii. Valuing Final State-Of-Charge 15](#_Toc466985330)

[h. Heuristic Aspects 15](#_Toc466985331)

[a. CCHP Systems 17](#_Toc466985332)

# Overview

EAGERS is a system that was developed for optimal control of distributed generation resources.

# Navigating the Interface

## Getting Started

Open Matlab and navigate to the EAGERS/main directory. Type the command ‘RUN’. A window will pop up with two options: Simple Dispatch Test or Launch EAGERS Interface. Real-time dispatch can only be done through the EAGERS Interface.

The Simple Dispatch Test can be used to evaluate the optimization strategy or building/generator configuration by running the optimization once, rather than in a receding horizon control. This option also allows the user to compute a map the optimal generator configuration for every possible power output.

Launching the EAGERS Interface opens a GUI with many options for selecting your building and generator system, then dispatching and controlling those generators to meet the building demands. Data on generators, utilities, building demand profiles, and weather data can either be loaded from a preset list or created through a series of GUI’s.

## Starting a New Project

If Start New Project is selected the user can either load data files, or construct a building demand from the prototypical building models available. Buildings can be added to the microgrid by highlighting the building type, climate zone/city, and vintage desired, then 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.

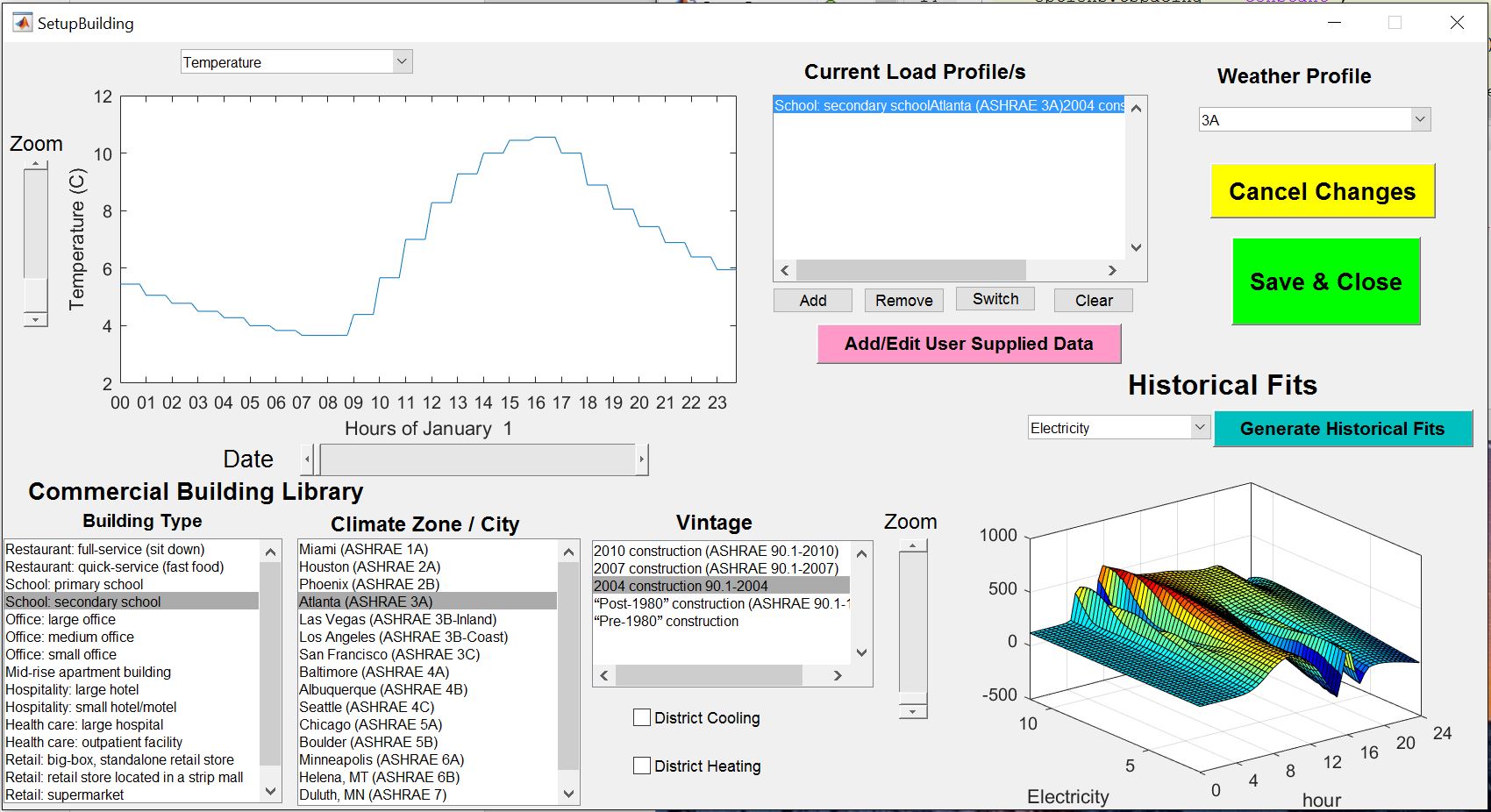


Figure 1 New micro-grid data can be added by selecting all the buildings included in the microgrid. In this example, a secondary school in Atlanta has been added to the current load profile

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 is an average daily temperature profile relating temperature (in Celsius) to the time of day. If electricity, heating or all are selected, a 2D surface fit of demand vs. time of day and ambient temperature is created for each month. Separate fits are made for weekend/holidays and weekdays. 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.

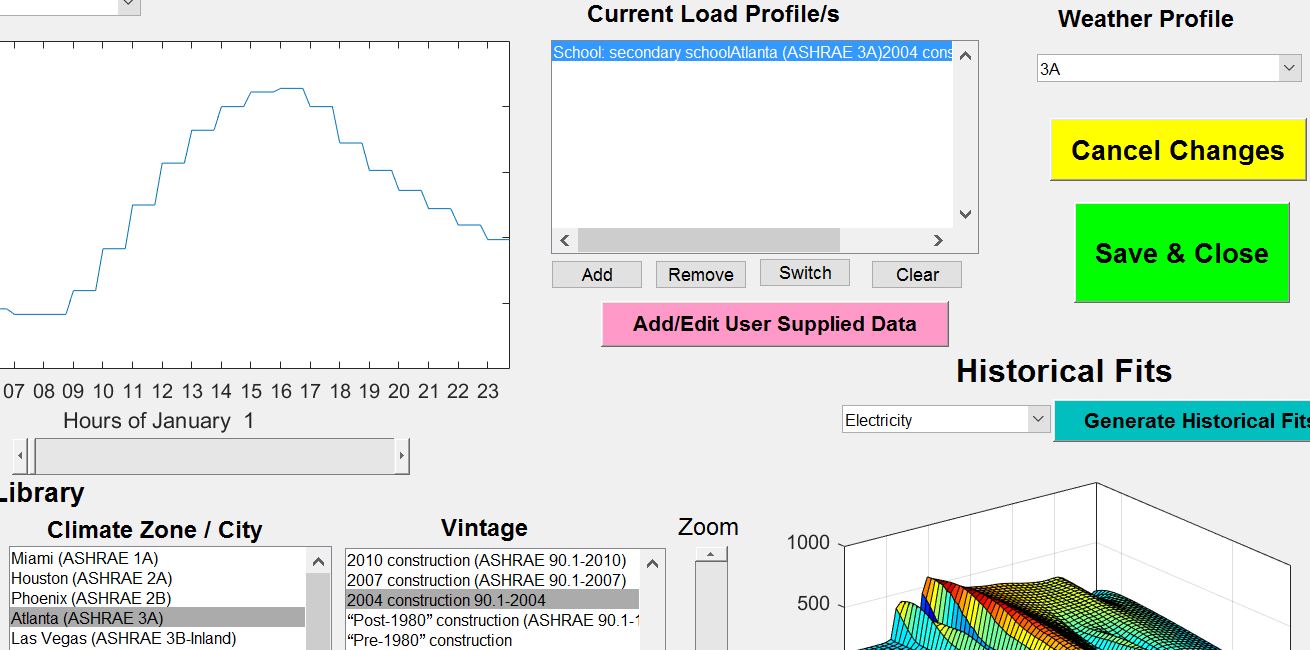
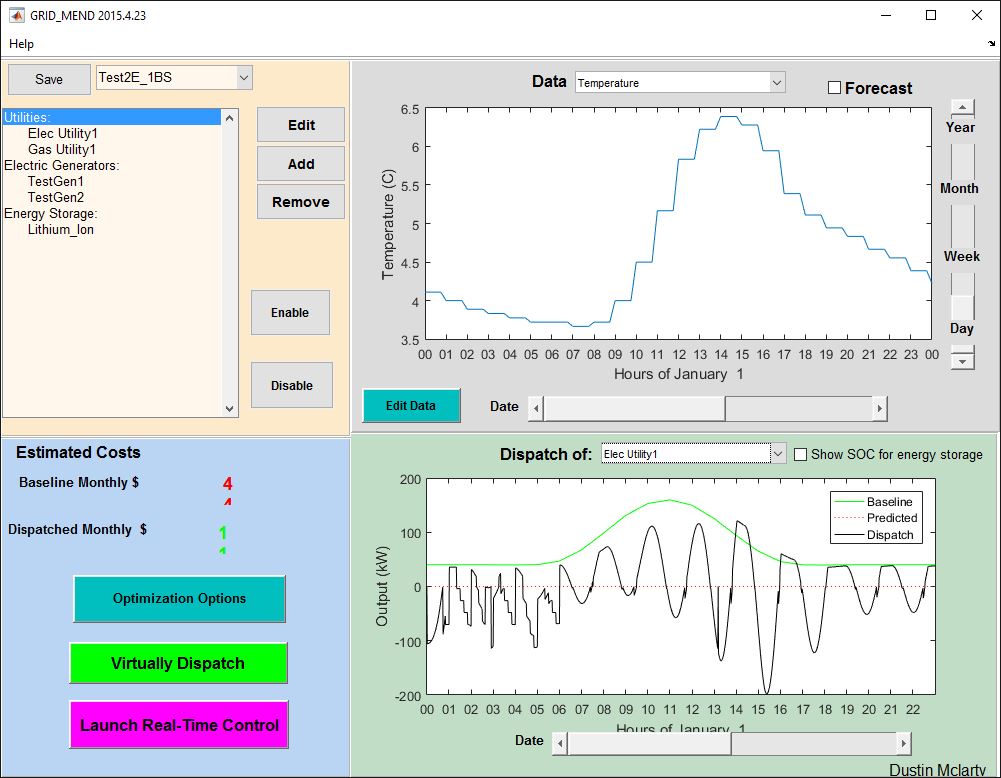


Figure 2 The correct weather profile has been selected for this city and the historical temperature and demand fits have been generated

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 EAGERS interface can be opened with the specified microgrid.

## Running EAGERS

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.



A

B

C

D

E

F

G

Figure 3 The main EAGERS 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).

### 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 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 allows the user to pick between one of 3 options. Constant time steps uses time intervals equal to the initial step size selected in the Dispatch Parameters. Logarithmically spacing uses a set of 8 time intervals of growing size between the initial time and the end of the dispatch horizon. Manually specify time steps, allows the user to specify the time intervals of the dispatch optimization.

#### 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 optimization frequencies 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.

### Launch Real-Time Dispatch

### Data Graph

On the top right of the EAGERS 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 EAGERS 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 specified as either a constant cost ($/mmbtu) or a variable cost which the user must provide as a vector of costs of the same length as the building data.

### 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.

### Electric Storage

Electric storage refers to any battery within the microgrid. All batteries are specified by size, peak charge/discharge rates, charge/discharge resistances and a voltage vs. state of charge curve. 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 thermal storage, typically of water. The size of the storage system can be quantified in volume and temperature difference (water) or in kWh of storage capacity. Charging rate limits and efficiencies are also specified.

### 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 4.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 5.

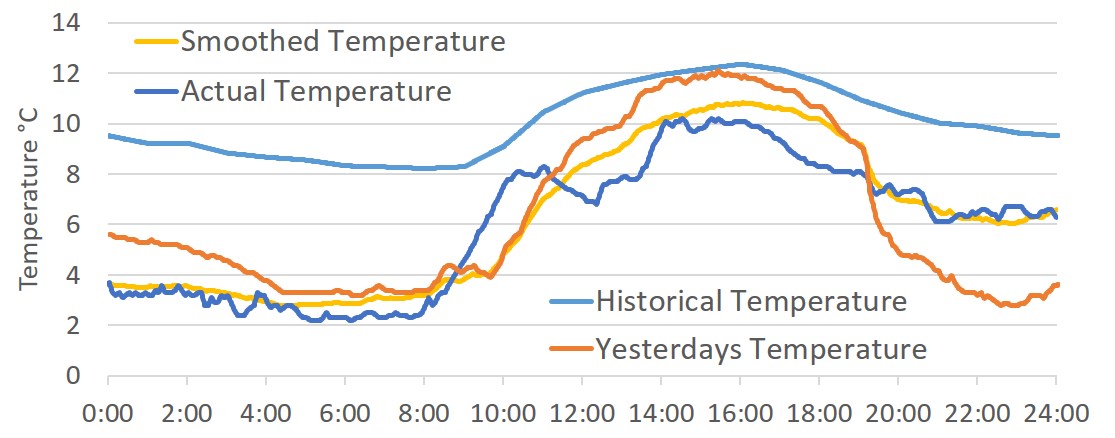


Figure 4 The temperature from yesterday has been averaged with the historical, then smoothed to form a prediction. This prediction is then compared to the actual temperature.

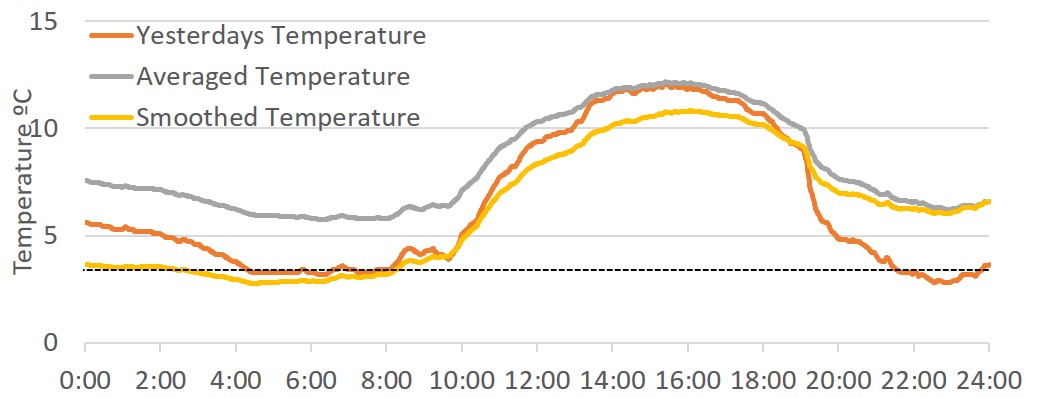


Figure 5 The last recorded temperature from yesterday is used to the determine the base for the smoothed temperature, which is then fit with the average of yesterday and the historical.

### 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.

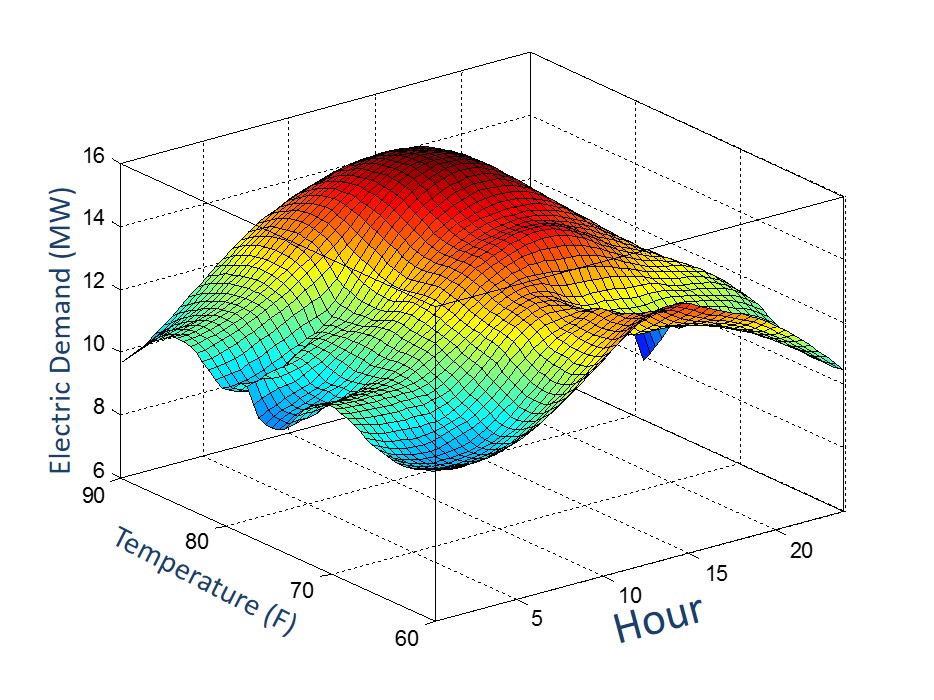


Figure 6 The surface fit made by forecasting electric load from averaging yesterday's and historical loads

### 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, EAGERS 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 EAGERS 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 EAGERS, 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

Refer to appendix for details of the complementary quadratic programming optimization.

# Glossary of Variables

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: 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.

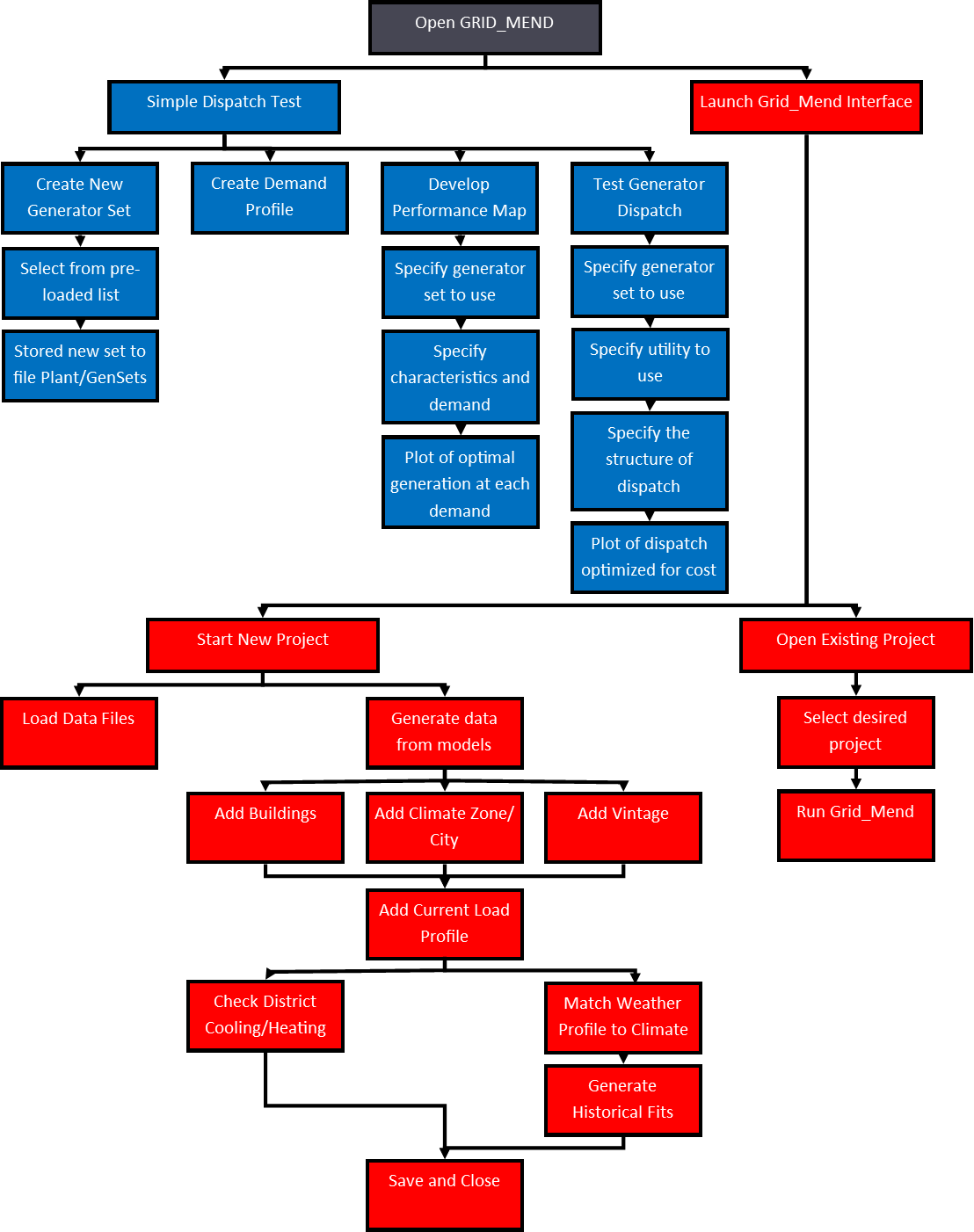
List of constraints:

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | ? |  |  |

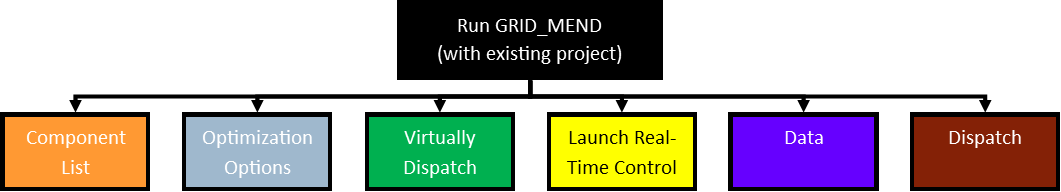
# Interface Flow Diagrams

## 1. EAGERS Interface

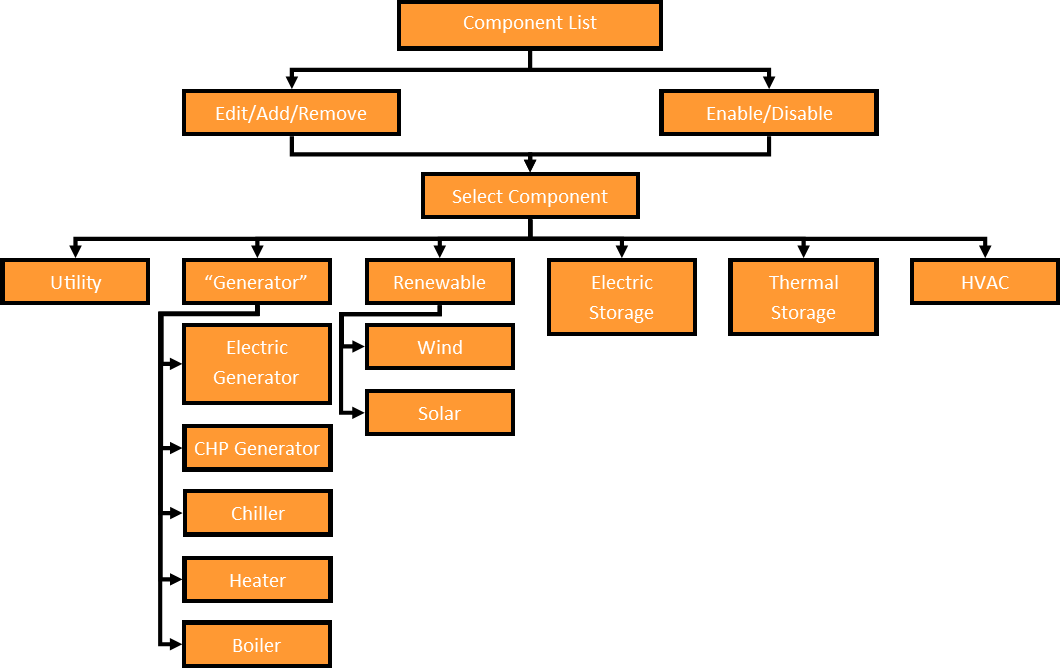
### Opening EAGERS



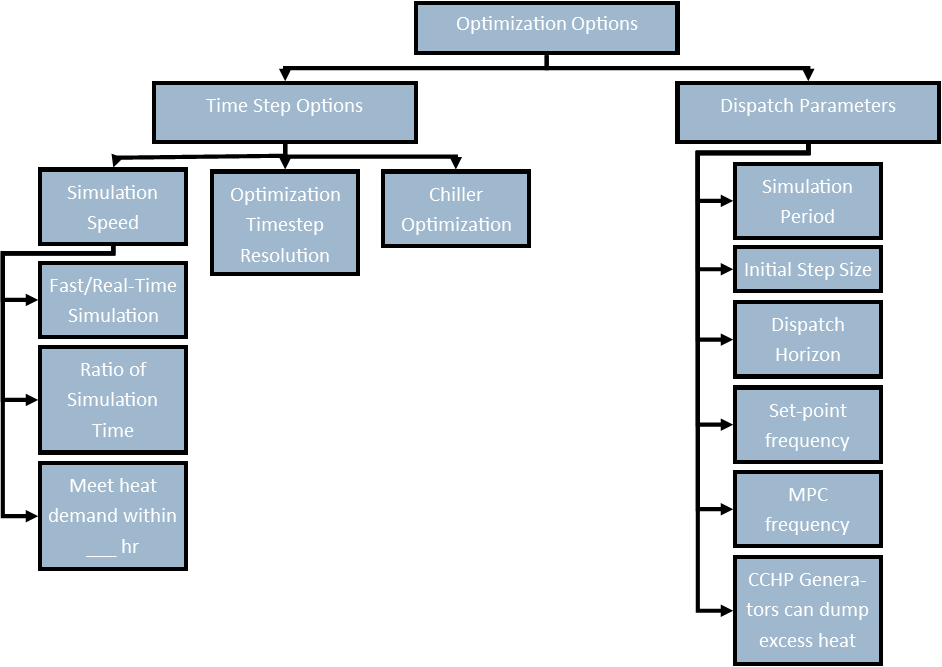
### Running EAGERS



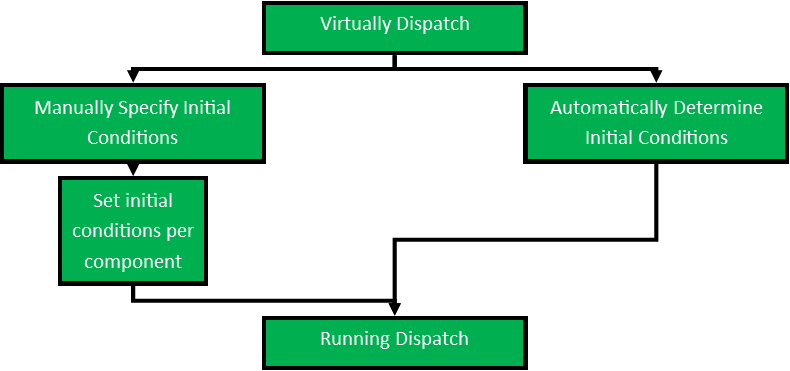
### Component List



### Optimization Options

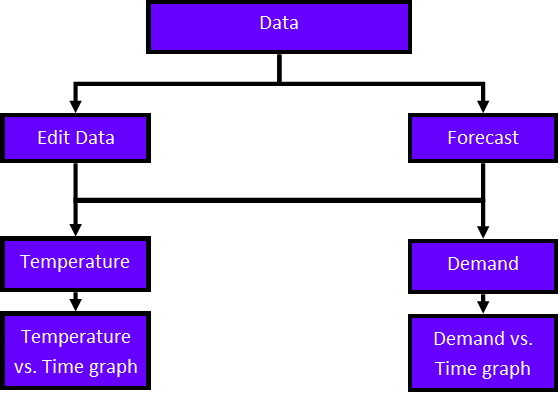


### Virtually Dispatch

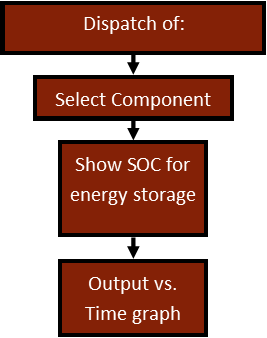


### Launch Real-Time Control

### Data Graph

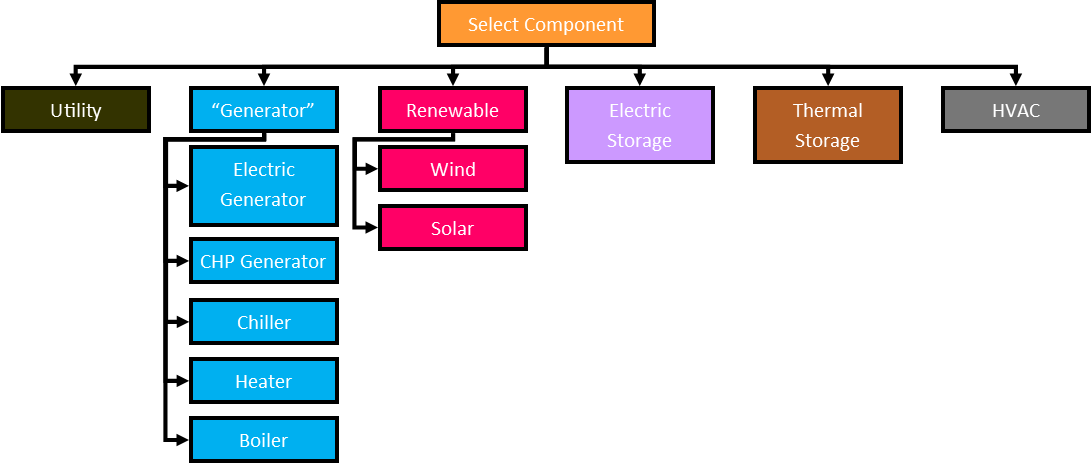


### Dispatch Graph

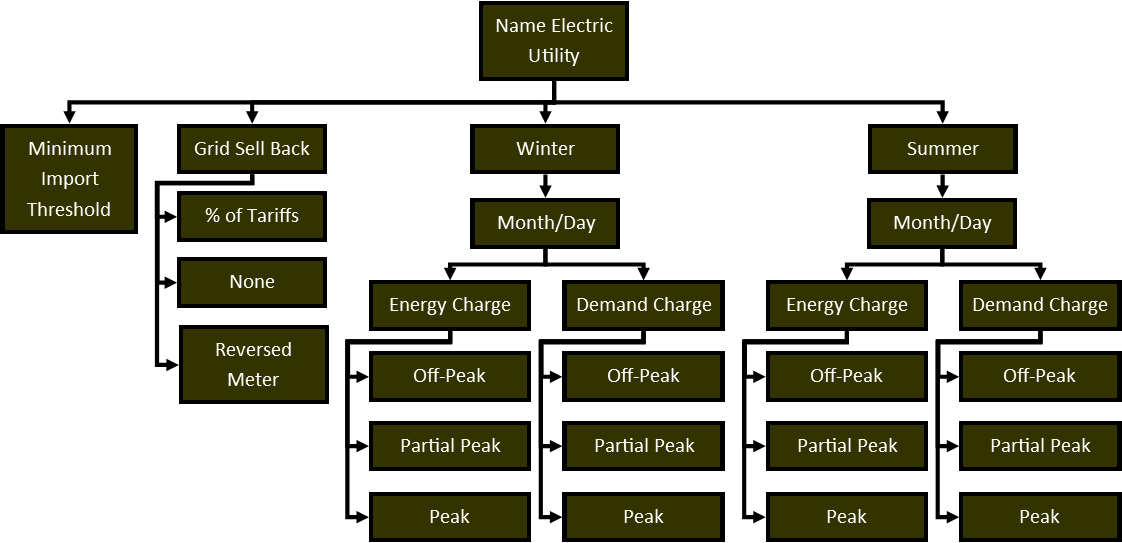


## 2. Component Edit/Add Interface

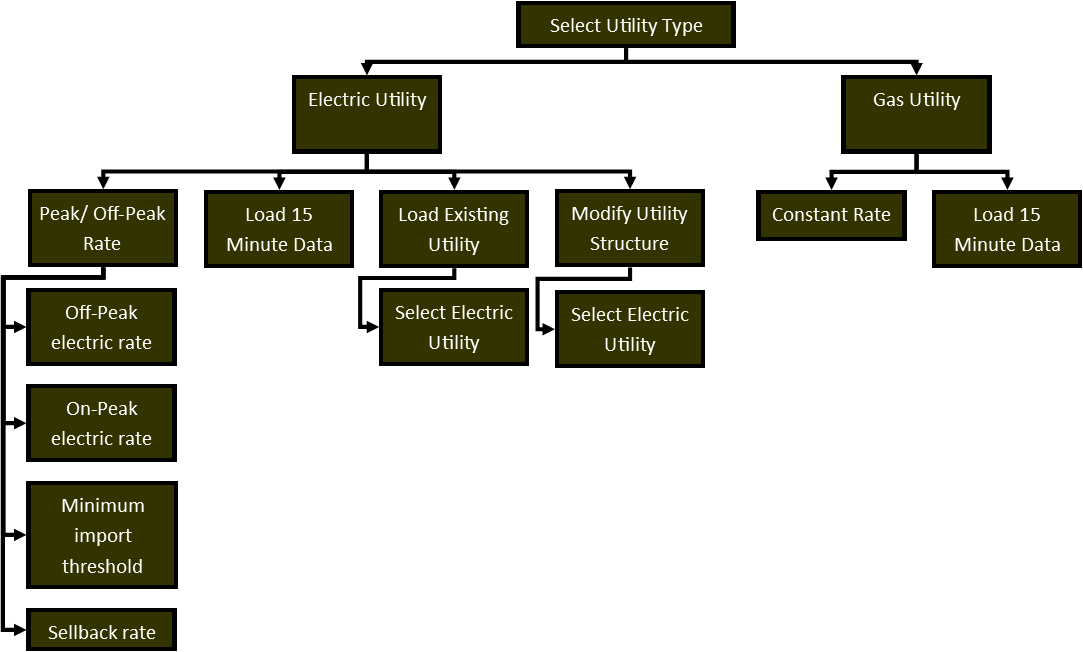
### Component List



### Add/Modify Utility



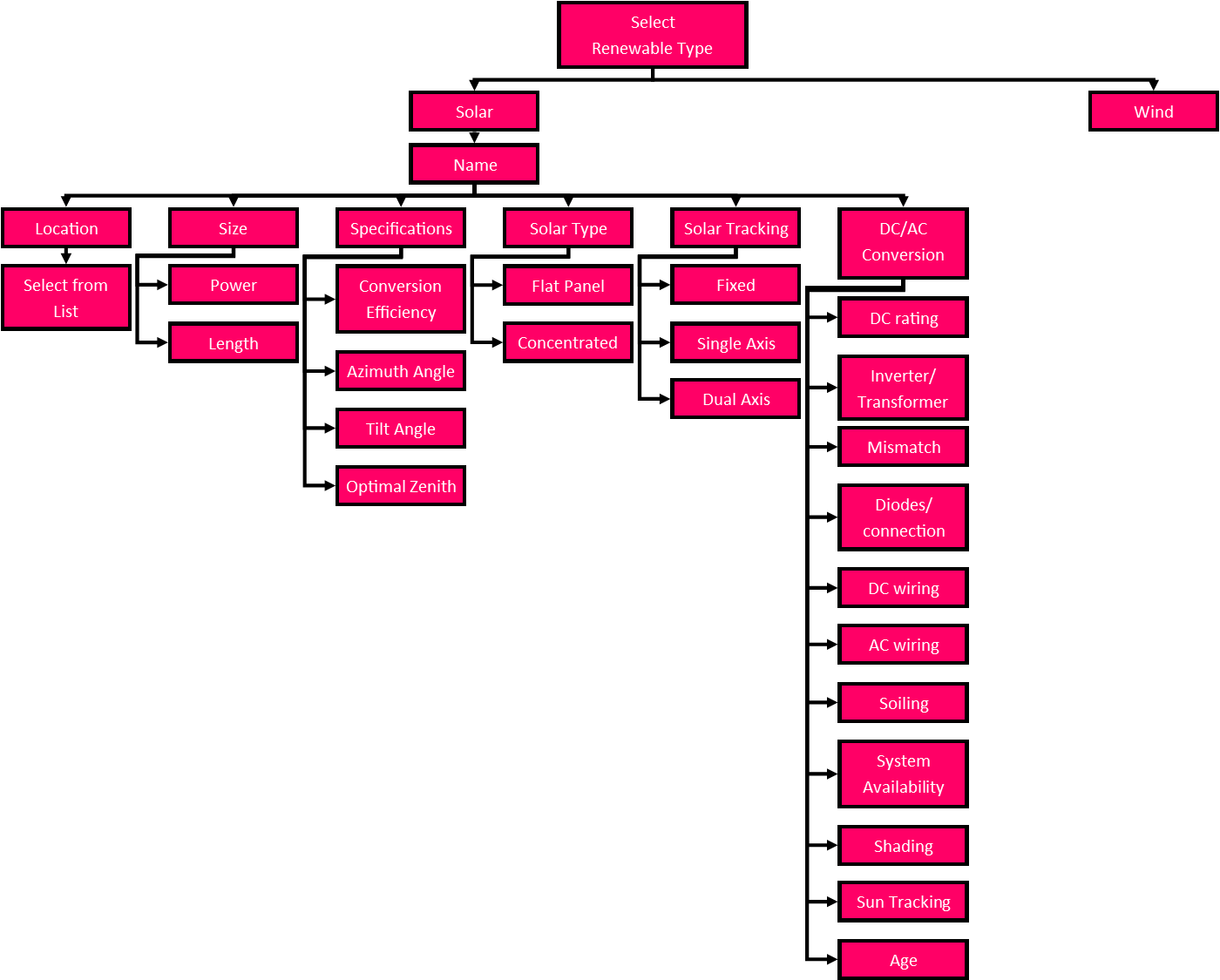
### Edit Utility



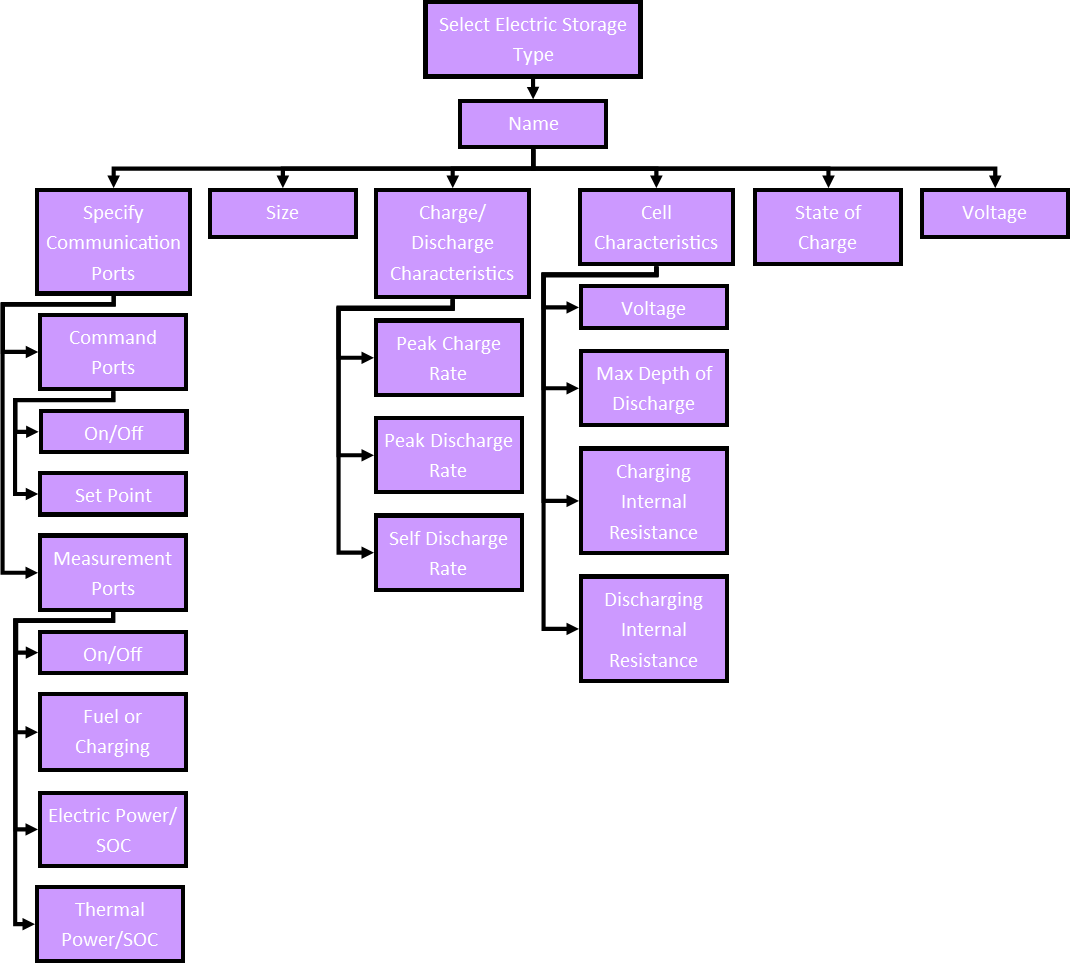
### Add/Edit Generator



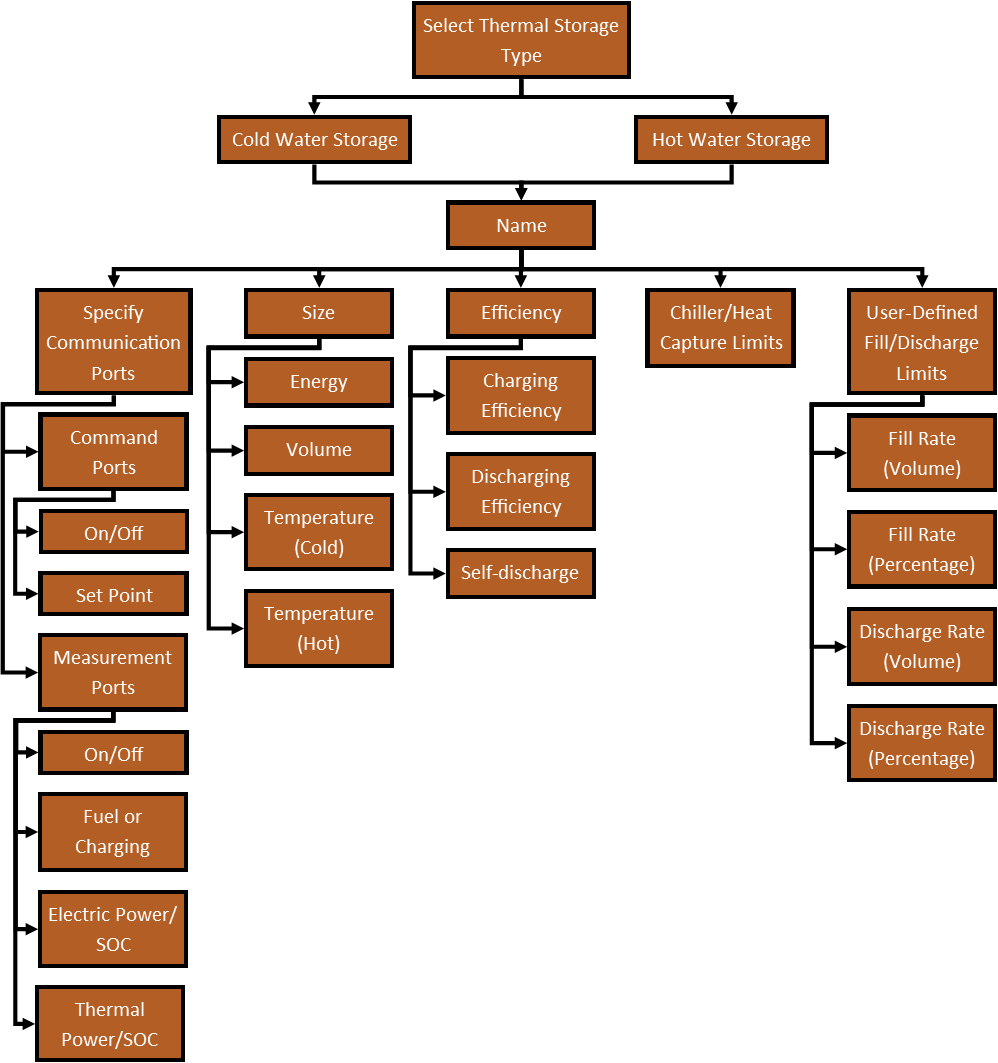
### Add/Edit Renewable



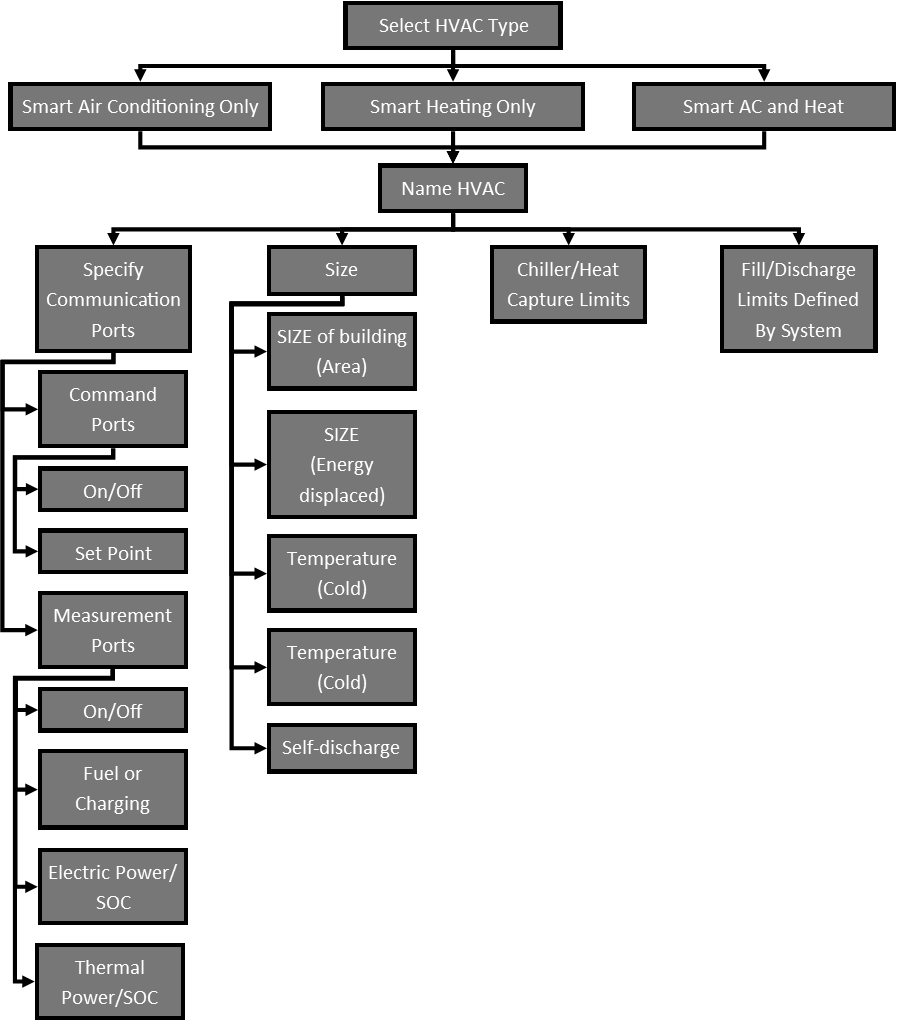
### Add/Edit Electric Storage



### Add/Edit Thermal Storage



### Add/Edit HVAC



# Appendix

The following is a description of complementary quadratic programming. It is based on a modified dynamic economic dispatch formulation that utilizes the state-of-charge (SOC) as the dispatchable state of the energy storage, and only assigns value to the final condition. This approach incorporates charging and discharging losses directly into the optimization.

The objective function is thus defined as:

 (1)

where the optimization horizon includes *N* time periods, the network includes *m* controllable generators and *r* dispatchable storage systems, and there are external connections to the grid at *g* points in the network. In (1) *Ph,k* represents the power output of generator *h* at time interval *k*, *Pf,k* represents the power transfer from external grid connection *f* at time interval *k*, and *SOCr,N* represents the final state-of-charge for energy storage system *r*. *Fh,k*, *Ff,k*, and *Fr*, represent the cost functions of the generators, grid connections and final value of stored energy respectively. The cost functions are scaled by the duration of the time interval, *Δtk*, to relate power to energy. The energy storage charging and discharging efficiencies are represented by *ηc* and *ηd.*

This general formulation described in (2-10) can meet demands for multiple products (e.g. electricity, heat, cooling), incorporate energy storage losses, and connect to a utility. Intermediary states and additional constraints discussed in section VII extend the formulation below to incorporate spinning reserve capacity and demand response as well as multiple grid connection points and transmission constraints including line losses.

 (2)

 (3)

 (4)

 (5)

 (6)

 (7)

 (8)

 (9)

 (10)

The power balance constraint is represented in (2), while (3) relates the power discharged from the storage system, *Pr,k*, to the change in the energy storage SOC. The charging penalty, *ϕ*, represents the additional power needed to recharge storage due to charging and discharging inefficiencies (4). This term appears in the demand equality (2) as an additional load when the storage is charging, and is zero otherwise (5). If implemented as an equality constraint, the energy dissipated during charging would re-appear as power when discharging. No cost is directly assigned to *ϕ*, instead the cost appears implicitly through the additional generation required to balance the power in (2). The additional inequalities constrain the output of the generators (6) between a lower limit, *Ph,kmin*, and upper limit, *Ph,kmin*, the generator ramp rates (7) below *rh,kmax*. The supply of power from the grid is similarly constrained (8), as is the rate of storage charging and discharging (9), and the minimum/maximum storage state-of-charge (10).

## Modifications to Dynamic Economic Dispatch

The solution approach of complementary quadratic programming differs from solutions based on dynamic system theory. The complementary approach solves three different optimizations and uses the instantaneous marginal costs of generation to determine a near-optimal dispatch solution that respects the lower operating bounds of all generators. In classic approaches, those that do not utilize mixed-integer programming, enforcing a non-zero lower limit, *Ph,kmin>0*, ensures the generator remains active over the entire dispatch horizon. Relaxing this constraint may schedule individual generators to at infeasible operating states. Each decision of state, on/off, must be made for multiple generators at each interval. The number of on/off decision variables quickly increases beyond what is practical to solve. The complementary quadratic programming approach breaks this into three steps:

*Step 1) Solving* (1) *with all generators, for the entire time-horizon. The lower boundary constraints are relaxed (i.e. Ph,kmin=0 and Fh,k(0)=0), but all other constraints are enforced. An initial marginal cost for any storage system is determined from the lowest and highest marginal operating costs of any generator. The resulting dispatch of generators, utilities, and storage systems becomes Schedule I. This dispatch may prove infeasible at certain intervals due to the relaxed lower boundary constraints, but steps 2 and 3 should resolve this issue.*

*Step 2) Solving the combinatorial problem at each individual time interval. The load at each interval is determined by the net generation calculated at that interval in Schedule I, thus pulling the initial estimate of the storage profile into step 2. The problem is formulated similarly to* (1)*, but with a horizon of a single step, and limited to some feasible subset of the 2m-1 generator combinations. Small deviations from the storage charging profile determined in step 1 are allowed in order to feasible subset of generator combinations. These deviations are given upper and lower limits according to capacity and charging rates permitted, as described by* (11)*.*

 (11)

*The deviations are assigned a linear cost equal to the marginal cost of generation at that moment to appropriately value any change in storage, and a tunable quadratic penalty to limit the deviation since this step cannot foresee when the storage may be more valuable. The marginal cost is the local slope of the cost function for the individual generators at the operating output determined by Schedule I. the Ramping constraints are ignored between successive steps to enable switching of generators, but the range of feasible operation at the current interval is determined from the ramping constraint and the initial condition at the start of the forecast horizon. The resulting lowest cost feasible combination at each step becomes Schedule II.*

*Step 3) A series of heuristic rules detailed in section IV are applied to Schedules I and II in order to determine the appropriate on/off sequence for the generators. With this sequence of generators in place* (1) *is solved enforcing all constraints. The resulting dispatch is termed Schedule III.*

At each interval there exist ***2m-1*** generator combinations, assuming unique generators, thus resulting in ***2(m-1)·N*** possibilities over the dispatch horizon. Steps 1 and 3 are each solving a single quadratic programing problem in the form of (12). Step 2 solves multiple quadratic programming problems in the form of (12), but with far fewer states, reducing the number of optimizations to less than ***N·2m-1***.

 (12)

Limiting the cost functions, *Fh,k*, *Ff,k*, and *Fr*, to convex quadratics enables a gradient based interior-point search method to quickly converge on a global minimum cost for the entire time horizon. The benefit of the proposed approach is that no arbitrary cost is assigned to the power flowing to or from the energy storage system. The drawback of this complementary approach is the limitation to convex quadratic functions, thus a best-fit convex function is desired.

## Piecewise Convex Quadratic Cost Fitting

Electric generators exhibit non-linear performance curves, depicted in Figure 1A. The fuel-to-electric efficiency (η) may be a function of power output, ambient conditions, fuel composition or maintenance frequency. The unit commitment problem inverts efficiency to find the specific cost of generation ($/kWh). Networks of generators can determine a sequence of operation defined by a single convex curve. When multiple generators with non-uniform capacity are to be operated over a range of efficiencies, the specific cost must be weighted by power output of each generator to determine an operating cost ($/hr), shown in Fig. 1A. This is generally non-linear and non-convex.

**Fig. 1A. Conceptual depiction of generator performance and cost functions. Typical electric generator efficiency (η), specific cost of generation ($/kWh), and non-linear operating cost curve ($/hr).**

**B)**

LB

UB

D

I

Fit A

Fit B

Discontinuity

η

$

kWh

LB

UB

Generator Output (kW)

**A)**

**$/hr**

With peak efficiencies typically at or near the upper boundary of power output, a piecewise convex quadratic with a zero y-intercept, Fit A in Fig. 1B, provides only a small improvement over estimating generator efficiency with a constant value. Fit B provides a better non-linear fit, but results in a non-zero cost when the generator is off-line.

**Fig. 1B. Two piecewise convex quadratic fits to the non-linear operating cost. Labels represent lower bound (LB), upper bound (UB), peak design efficiency (D), and the inflection point (I).**

These piecewise quadratic cost functions are implemented in the economic dispatch formulation by representing the generator output at each interval, *Ph,k*, with two intermediary states, *βh,k* and *γh,k*, and the following constraints.

 (13)

*Fit A:*

 (14)

 (15)

 (16)

*Fit B:*

 (17)

 (18)

 (19)

Schedule I uses Fit A when no prior knowledge of which generators are on-line at each interval is available. Schedule II and III use Fit B to more accurately reflect the marginal cost of generation at part-load. A linear-least-squares problem determines the coefficients, *a* and *b*, constrained to ensure a continuous convex fit. The constant term, *b4,h*, not present in (12), is used to compare costs between different generator sub-sets in step 2.

## Valuing Final State-Of-Charge

Rather than constraining the final SOC with an equality constraint, the final SOC is given value in the cost function. A greater residual SOC must lower the net cost, otherwise the SOC would always be driven to zero at the end of the dispatch horizon. A quadratic cost (20) is determined from the marginal cost of the least expensive and most expensive generation using (21) and (22). The coefficient c1,r will always be negative implying a savings with additional stored energy, and the coefficient c2,r will always be positive to maintain a convex problem. The steep negative slope of this cost function at zero SOC implies a preference to use the most expensive generator before fully depleting the storage. Similarly, the less negative, or possibly positive, slope at full SOC suggests a preference to discharge storage before using the least expensive generator. This does not assign value to the intermediate steps, thus allowing the storage to be fully utilized between its limits and maximize its value within the dispatch horizon. A small refinement in step 3 is to avoid considering inactive generators when calculating the maximum marginal cost, so as to not over-value stored energy.

 (20)

 (21)

 (22)

### Heuristic Aspects

Step 1 does not enforce the lower bound constraints on each generator’s operation, and thus eliminates generators with the highest marginal cost according to Fit A. Schedule I may permit infeasible operating conditions for some generators and may incorrectly estimate marginal costs due to the error in Fit A at part load conditions. Generally, Schedule I approximates an ideal dispatch for the energy storage.

Step 2 iterates through a sub-set of the feasible generator combinations at each interval to identify the optimal configuration for generating *Pnet,k*, the net generation at interval *k* according to Schedule I as formulated in (23).

 (23)

The ***2m-1*** generator combinations are reduced by first ruling out any infeasible combinations given the upper and lower operating constraints. If the sum of the minimum output for a set of generators exceeds the demand, less any uncontrollable generation and what the grid and storage can provide, then it is infeasible (24). Likewise, if the maximum output is greater than the sum of demand, what can be offloaded to the grid, and what can be diverted to storage, the generator set is infeasible (25). If either (24) or (25) is true, that particular generator set is infeasible. Energy storage and utilities, if present in the system, are included with all generator sub-sets. Thus if the grid connection permits unlimited purchasing and selling of power, all generator combinations are feasible. The upper limit of storage charging or discharging may also be limited by the current SOC when applying (24) and (25).

 (24)

 (25)

The combinations to be tested can be further reduced by starting with feasible sub-sets that include the fewest number of active generators and applying the following logic:

* If the current set is the least expensive tested thus far, calculate the average cost per kWh for this interval with this set. Ignore any untested sets that include all generators from the current set and that do not include additional generators with a cost less than this average.
* For each generator, *h*, in the current set, identify untested combinations that include all other generators from the current set. Do not consider combinations that replace *h* with at generators with a minimum actual cost per kW higher than the cost per kW of *h* at the power setting determined for the current set.

Step 2 does not constrain the generator ramp rates between successive intervals, allowing for rapid switching between generators. Step 3 balances this desire for optimality at every moment in time with the global perspective of Step 1. It does this by applying the following heuristic rules to determine the on/off sequence for each generator, then solving (1) with all boundary constraints enforced.

1. At each interval, do not use any generator which is off-line in both Schedule I and Schedule II.
2. Any generator initially off-line can remain off-line until it first appears in Schedule II or until it must begin ramping up to reach the target set in Schedule II. Similarly, any generator expected to be off-line at the end of the horizon can remain off-line from their last appearance in Schedule II onward, subject to the ramping constraint.
3. If a generator appear off in Schedule II for a period of time longer than that needed to ramp down to zero from its last setting and then ramp up to the next setting in Schedule II, compare the duration of time it would remain off, multiplied by *b4,h*, to the specified start-up cost for the generator. If turning off then on saves money, after accounting for the start-up cost, proceed with the temporary shut-down.
4. If a generator appears in Schedule II for a consecutive window shorter than ¼ of the horizon or less than the time required to ramp to full power and back off, consider leaving it off. Compare the net cost of operation during this window plus the start-up cost of the generator, to the average marginal cost of the other on-line generators. If not using this generator during this short window does not render the dispatch infeasible and results in net savings, then leave this generator off in this window.

## CCHP Systems

Incorporating additional generation products, e.g. heating, cooling, or steam, requires an additional equality constraint similar to (2) for each time step, *k*. In many instances excess production of heat, cooling, or steam can be vented, and the equality constraint can be replaced with an inequality that ensures enough production to satisfy demand. This may help reduce the constraints on CCHP generators, allowing them to produce more electricity and vent any excess heat that is recovered. A limitation of this approach is that any secondary output, e.g. *Ph,k|heating,* of a generator must be assumed to be proportional to the primary output, e.g. *Ph,k|electricity*, as in (26).

 (26)

Optimizing for both electricity and cooling production using electric driven chillers adds a degree of complexity to the dispatch since the chiller input cost is the output of the electric generators. If there is no cold thermal storage, there is little flexibility in meeting the thermal demand, so it is preferable to first optimize the chiller dispatch, then add the resulting electric demand from the chillers to the building’s electric demand. This approach captures the off-design performance of the chillers with a similar piecewise convex quadratic fit where the cost is the kWh of electricity needed.

With cold energy storage it becomes feasible to use chiller loads to balance the electric demand, and thus the dispatch should be done concurrently. In this scenario the chillers are assigned zero cost, but in the electric demand equality the inverse of the peak chiller COP, *αc*, appears as an additional electric load on the chiller output state, *Pc,k*, as shown in (27). Lacking a quadratic term in this equality implies that a single value COP must be used for the chiller. However, given the flexibility in dispatch afforded by the thermal storage, it is generally preferable to operate all chillers at their design condition, thus making this simplification easily justifiable.

 (27)

When solving for multiple outputs, it is advantageous to separately determine feasible combinations in step 2, searching a space of ***2m-1*** + ***2d-1***, rather than ***2(m-1)(d-1)***; otherwise, the methodology remains unchanged.