New York Academic Model and Market Model

# Purpose:

The purpose of this document is to describe the models in sufficient detail for those not familiar with them to be able to use it in their work.

# New York Academic Model

## Background:

In 2016, the NYISO and RPI formed a research agreement to develop a reduced network model of the New York transmission system. The goal was to create a network that mimicked some behaviors of the NY network without using, and thereby not revealing, and confidential information.

This model was to be used in academic research. It would be most useful if it contained a set of generators that matched the NY fleet, included interfaces that somewhat resembled interfaces of interest to the NYISO including Central-East, and allowed for study of a unique New York areas like Long Island. It was desirable that the model be usable for both long-term planning/market style analysis and short-term stability style analysis.

## Introduction:

The NYAM was formed using the old NPCC 16-machine, 68-bus system as a basis. The out-of-date model, which covers the NPCC footprint as shown in Figure 1, was developed in the 1980’s. Unit 9 in Maine is the long-retired Maine Yankee unit.

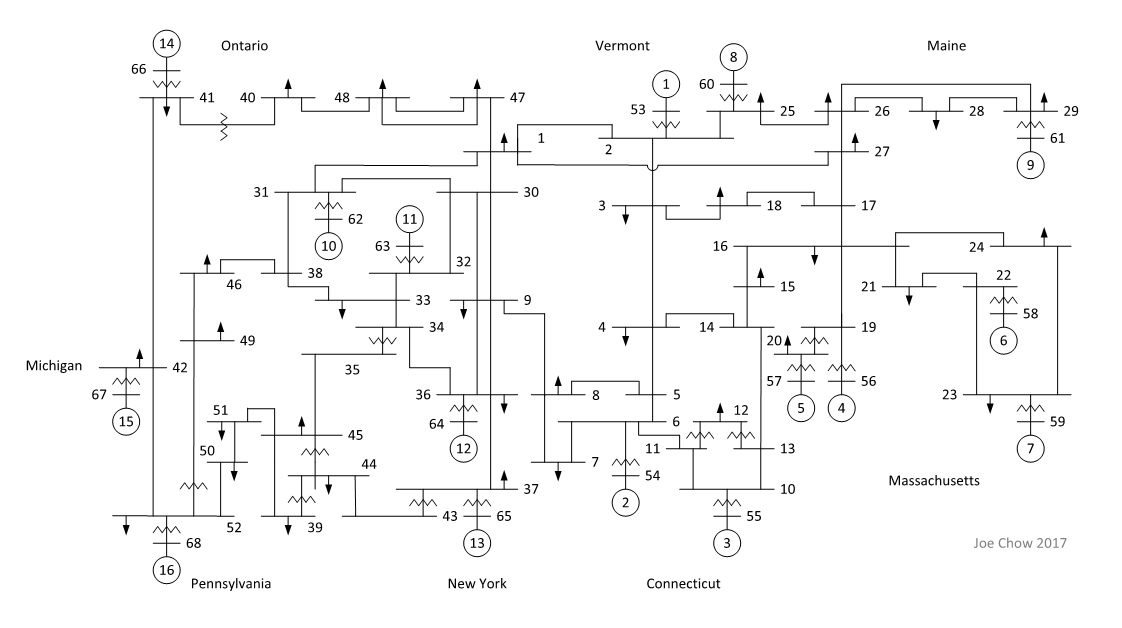


Figure 1: NPCC 16-Machine, 68-Bus Network Model

The NYAM reuses this network configuration and identifies regions with interfaces that somewhat resemble those of interest to the NYISO. These are shown in Figure 2. While the NYISO has 11 load zones, the NYAM uses only 5. The Upstate Region includes Zones A through F. Lower Hudson Valley (LHV) includes G through I. Zones J and K are still NYC and Long Island (LI), respectively. A fifth region represents ISONE with a gen bus and load buses representing the ability to import and export, respectively. Generator 14 in the Upstate region represents import from Hydro Quebec.

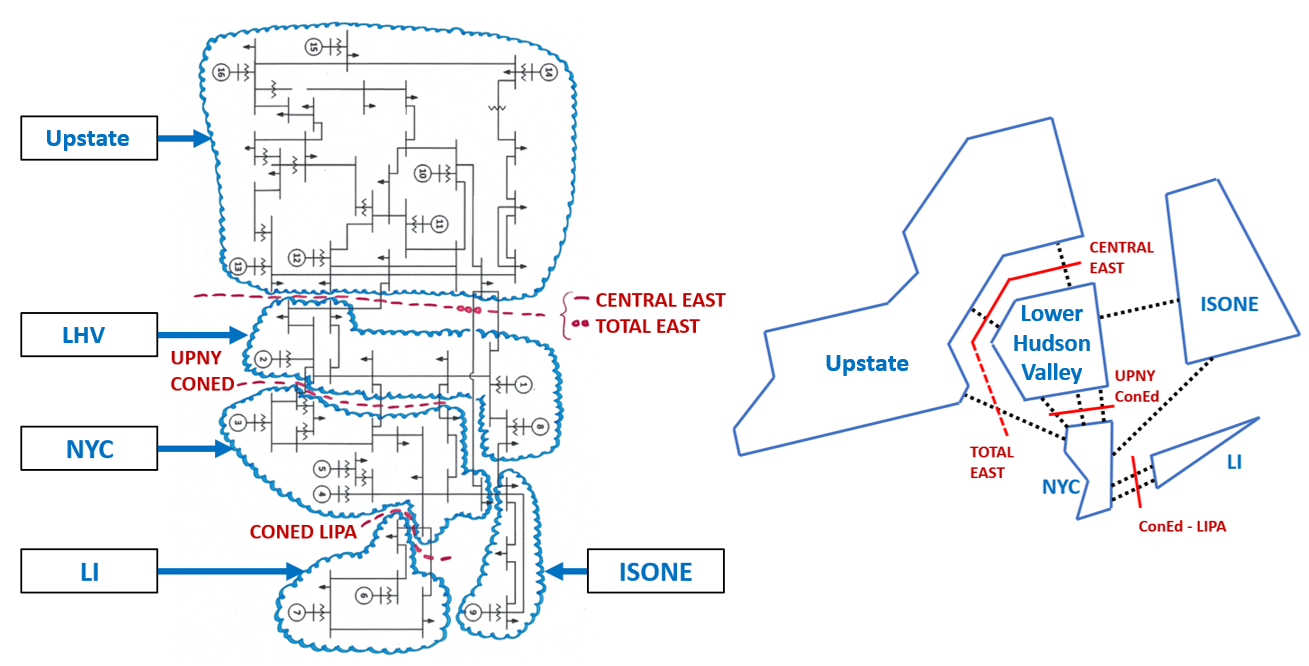


Figure 2: NYAM Regions and Interfaces.

It is important to note that while these interfaces bear strong resemblance to those of the NYISO, it is incorrect to assume that the flows across them will exactly match real operational flows. By design, this is an academic model appropriate for analysis of general trends, not an exact replica of the NY system.

### Generation

#### Renewable Generation

The NYISO provided two sets of data for use in developing the NYAM’s renewable generation. The first is a set of installed MW by renewable generator type, by zone. The second is 4 days of 5-minute output data by renewable generator type, masked by zone.

#### Installed Renewable Capacity

Governor Cuomo’s so called “50-by-30” target establishes a goal that 50% of the energy consumed by New Yorkers come from renewable resources by the year 2030. The NYISO indicated that this goal would be achievable in an “incremental” amount of renewable capacity was added to the NY system.

The NYISO provided (i) the amount of renewable capacity installed as of 2016 and (ii) the incremental capacity which must be added to the 2016-existing capacity in order to reach the 2030 target. As seen in Figure 3, by adding the Incremental capacity to the 2016 capacity, capacity for the 2030 case is obtained.

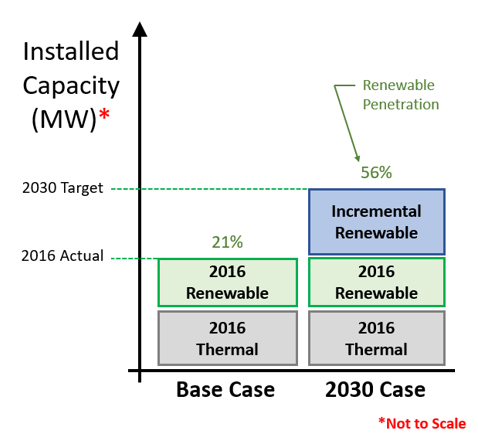


Figure 3: Renewable Penetration Cases

The importance of distinguishing the incremental capacity will become apparent in the next section. Note that the amount of installed capacity needed to ensure that 50% renewable consumption is higher than 50% penetration. Figure 4 shows the actual installed capacity by region and type.

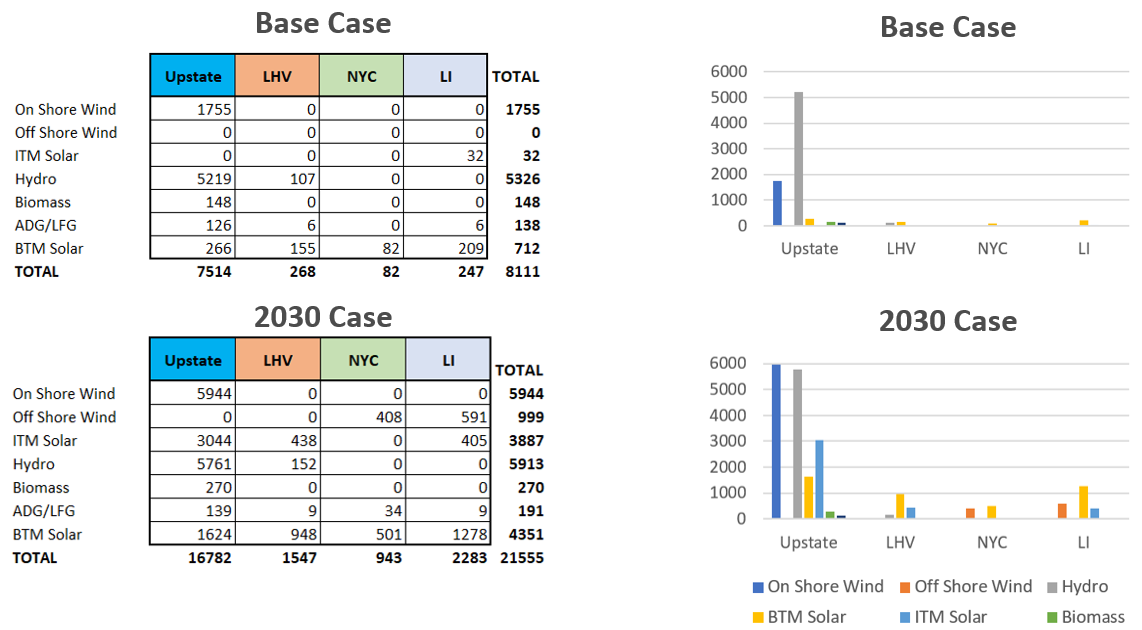


Figure 4: Installed Renewable Capacity by Region and Type

#### Renewable Output Profiles

The NYISO provided 4 days of output profiles, each with 5-minute granularity. A unique profile was provided for each renewable generator type and for each of the 11 load zones. Each profile represents the instantaneous MW output for the incremental installed capacity of a given generator type in a given zone. This output series was divided by the number of installed incremental MW and then scaled to the desired case by multiplying by the MW capacity installed in that case.

The renewable output in the desired case was calculated for each zone, summed by NYAM region, and distributed evenly over the generator buses in each NYAM region. No new generator buses were created.

In day-ahead analysis, hourly averages of renewable output are used. In real-time, actual 5-minute values are used.

### Thermal Generation

A thermal generating fleet was constructed in an attempt to match the NYISO fleet as of 2016. Due to the limitation on the number of available generator buses, multiple NYISO units of the same engine type were combined into single units in the NYAM. Using the 2016 goldbook as a guide, a set of 12 thermal generators was created consisting of 3 nuclear units, 5 steam units, 2 combined cycles, and 2 gas turbine blocks. The operational parameters are shown in Figure 5 .

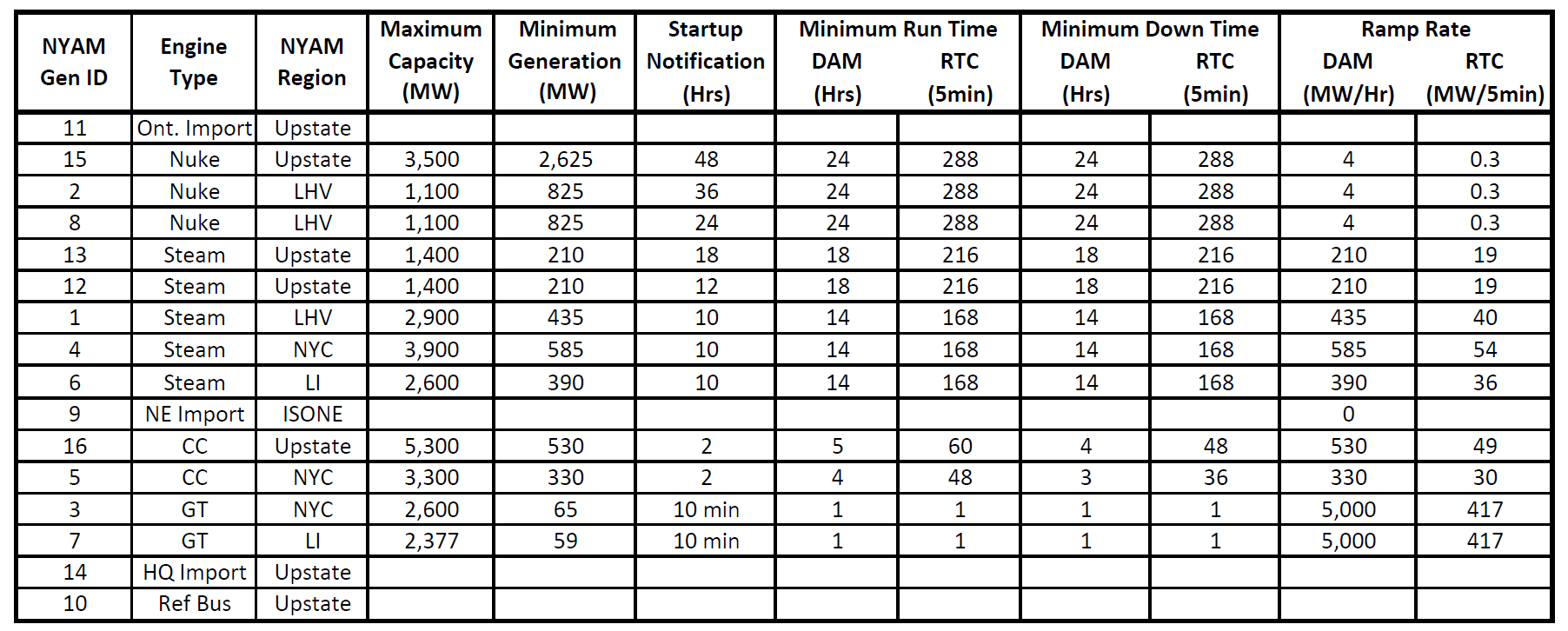


Figure 5: NYAM Thermal Generator Operational Parameters

### Load

NYAM load is obtained from the NYISO’s OASIS. 5-minute actual load by zone is regionally aggregated and distributed evenly over the that region’s load buses. In day-ahead analysis, hourly averages of regional load are used. In real-time, actual 5-minute values are used.

Net load is an input to the market model. The process for calculating net load for each simulation is as follows: 2016 net load is obtained from OASIS; the effect of 2016 BTM generation is removed to produce the 2016 true load; BTM generation for the present simulation is calculated; the effect of this BTM generation is added to the 2016 true load. The upshot is net load for the desired penetration case.

# Market Model

## Introduction

The Market Model is a two-settlement system with day-ahead and real-time capability. The real-time engine has separate dispatch and commitment processes that occur every 5 and 15 minutes, respectively. The MATLAB model uses MATPOWER and MOST to solve DC power flow with Gurobi selected as the solving engine. Users can select the day(s) and renewable penetration case(s) to study, using a number of variables to specify desired parameters for these simulations.

## Up and Running

Users must install MATLAB, MATPOWER, MOST, and Gurobi on their computers. Additionally, load and renewable generation data files must be locally stored, along with the actual market model matlab files. Users must have a good working knowledge of matlab to use the market model. The “Main File” is where most computation occurs. Here, users must update paths and file locations as needed to enable the algorithm to properly access data as needed. New users should expect to spend a significant amount of time getting the program up and running.

Once the program is operational, it can be expected to run and automatically print graphs to a word document.

## General Layout

The code is organized as follows:

* Setup
* Net load
* Renewable generation
* DAM run
* First RTC run
* First 3 RTD runs
* Second RTC run
* Start of loop
  + Next 3 RTD runs
  + Next RTC run
* End of loop
* Post run analysis

What follows is a discussion of each section. The author would like to note that this code is the product of several years’ worth of effort. There are several opportunities for improving the code from an organizational perspective. Users are more than welcome to improve the code themselves.

### Setup

* Changes

Users are strongly encouraged to modify the code to suite their own analysis needs. A sturdy system of revision control is strongly recommended. Users may be able to quickly modify code and perform various types of analysis. However, without proper documentation, these results may not be duplicatable by the user later on, leading to great frustration. Disk space is cheap and github is free. Save copies every time a significant change is made and store a copy locally and on github. Not only of the main file, but the functions and program files all together. It will take longer. But think of it as investment. If you borrow time by skipping revision control, you will pay back later with steep interest.

* Setup

These commands wipe clean the calculation slate used by the program. It also starts a timer that measures how long the program will run.

* Add Paths

This is where the user will specify the locations of key files on their computer. Make sure these are updated between each git revision.

* Font Size and Publishing

This is where users can manipulate the fonts in figures created later on. They currently set to create a size suitable for IEEE publication.

* Define initial arrays

Here is where the load buses by region are identified.

* Add transmission interface limits

The MAP array identifies the interfaces and the lines that compose them. Here we identify 4 interfaces and the branches that make up each interface. Directional flow is identified using positive and negative values on the branch number. For additional information, check out the Matpower manual’s section on interface flows. The interface flows in MOST are, as of this writing, not included in the MOST application provided by Ray Zimmerman on github or his site. I have added them myself. The “most\_if” function is used instead of the “most” function when calling on multiperiod power flow problems. The most\_if function has the interface flow limits in place. The published version of matpower has interface flow limits incorporated and well documented in its user manual.

The LIMS\_ARRAY array is where users can specify the interface flow limits.

* Pick Date Range

The dates range from 1 to 4.

|  |  |  |
| --- | --- | --- |
| **Day** | **Date** | **Fun Fact** |
| 1 | January 19 | 2016 Winter Peak |
| 2 | March 22 | Thunderstorms & IP refueling |
| 3 | July 25 | 2016 Summer Peak |
| 4 | November 10 | High Wind Day |

* Pick Case

The cases include “0”, the 2016 Base Case; “1” the 2030 Case; and two additional cases with even higher renewable penetration levels which are now defunct.

* Other variables

There are several other control variables which follow. If the comments describing their actions are not sufficiently clear, then the user should step through code operation to gain understanding.

### Net Load

* Get Net Load from OASIS

Users should download the 5-minute load data from OASIS and store it in a folder specified in this section. The code will automatically convert it from csv to .m files.

This section of code converts the inputs into various forms needed by the program later on. Users should walk through the code to gain an understanding of the specifics. Once this is complete, users may not need to consider it again.

The input is OASIS data and simulation parameters already mentioned. The output is a series of load and renewable profiles which can be used by the DAM and RTM.

### Day-Ahead Market

Most uses something appropriately called “profiles” to enable the use of variables that change value from period to period in multiperiod analysis. This is useful for specifying load and renewable outputs, which obviously should be expected to change from period to period. Further, these profiles can support changing generator parameters and anything else which the user wants to change over time.

Users should read the entire Matpower and Most manuals, perhaps with the exception of the sections that explain the math and formulas behind operations. A through understanding of these two programs is essential to understanding how the Main File works. More often than not, the answers to users’ questions may be found in these manuals. Serious users are recommended to print a copy of these manuals, place in a binder, tabbing sections and highlighting areas which have already been referenced.

Once the DAM simulation is complete, a series of DAM results may be printed. Additional graphs may be created by the user as desired.

### Real-Time Market

See Figure 6 for a visual representation of the RTM’s operation.

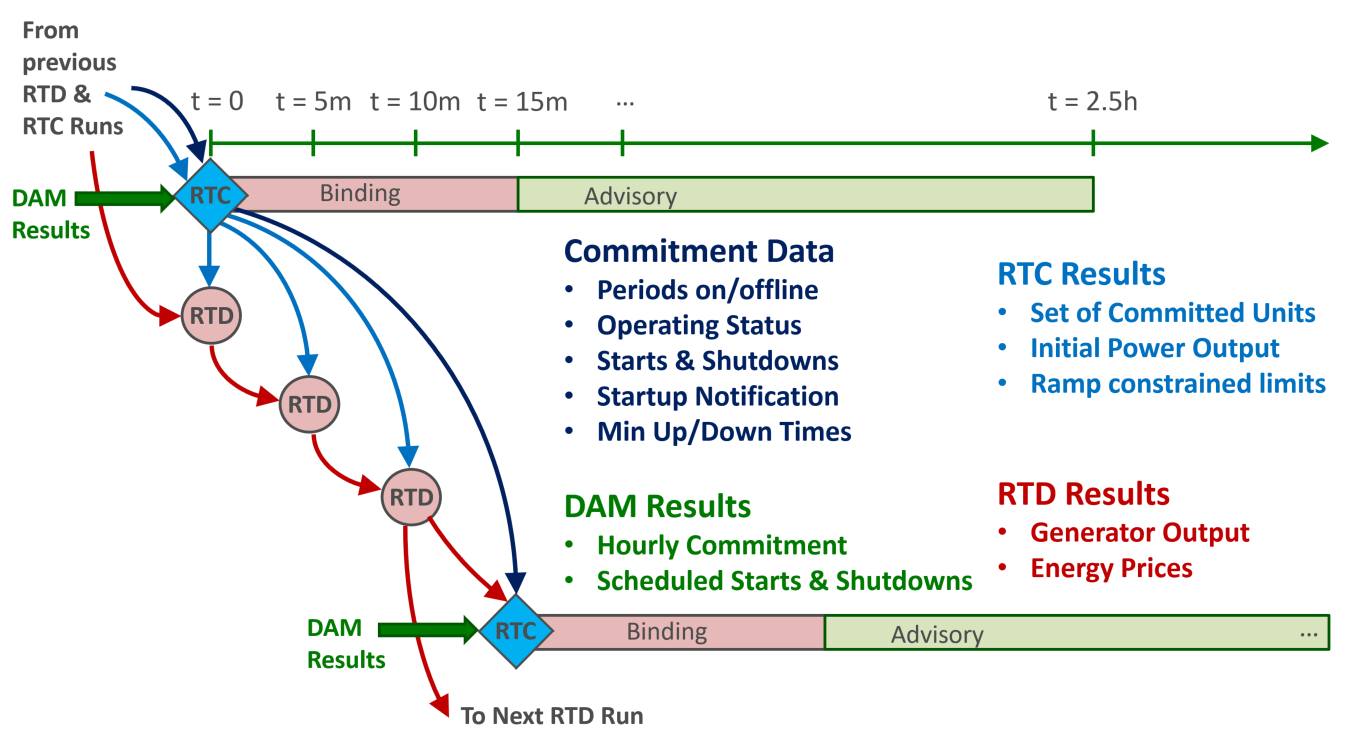


Figure 6: Real Time Market Engine

Due to the complexities of the first few RTC runs, the first two are performed individually (out of the loop). Once RTC 1; RTD 1, 2, and 3; and RTC 2 have been run, then a loop is entered where the next three RTD runs and an RTC run are performed.

#### Real-Time Commitment

The RTC engine determines the units which will be online and available for dispatch by RTD. RTC commits units while RTD dispatches them. The DAM will commit and dispatch *all* units over a 24-hour period using hourly intervals. In real-time, it may make sense to split the total set of units into two groups: those who can start in 30 minutes or less and those who cannot. DAM committed units in the later group are considered must-run in RTC. However, DAM commitments for the other set of units are ignored in real-time. RTC has complete dominion over real-time commitments of units that can start in 30 minutes or less.

Note that a unit’s startup notification time is defined by the user in gen\_startuptime\_hrs. Note that this value must be in the form of a fraction. The use of decimals may result in errors.

Bright users will gain insight into the various MOST parameters by reading the MOST manual. Outside of this, the code and its comments will be the best way for users to unravel the complexities of the real time engines.

#### Real-Time Dispatch

Unlike the multiperiod DAM and RTC problems, RTD is a single period problem. RTD is solely responsible for determining each unit’s output in an economically optimal manner. Note that what’s economically optimal in a given instant may not be optimal across a longer period of time, a concept discussed extensively in the author’s PhD dissertation.

### Post Simulation Processing

A series of graphs is produced and pasted to a word document.

Additionally, analysis of the financial results is performed. An excel file is provided which users can paste AllRunsSummary and DAMresults into to compare various sets of simulations. A thorough user will work through the formulas and calculations to gain a complete understanding of the analysis.