

# ENERGY 2100

## Documentation

Volume **3**

---

# Electricity Supply Module

Last Revision date: June 2026

## Table of Contents

---

1.	Introduction .....	6
	Structure of Electric System .....	9
	Electric Supply Source Code .....	11
	Load Curve Submodule.....	12
	Contract Development Submodule.....	12
	Retail Purchases Submodule .....	13
	Pollution Costs Submodule.....	13
	Capacity Expansion Submodule.....	13
	Generation Dispatch Submodule.....	14
	Fuel Usage Submodule .....	15
	Pollution Submodule .....	15
	Retail Power Costs Submodule.....	16
	Electric Pricing Submodule .....	16
2.	Load Curve Submodule.....	17
	Submodule Objective.....	17
	Submodule Methodology .....	19
	Submodule Functions .....	20
3.	Contract Development Submodule.....	31
	Submodule Objective.....	31
	Submodule Methodology .....	32
	Submodule Functions .....	34
4.	Retail Purchases Submodule .....	40
	Submodule Objective.....	40
	Submodule Methodology .....	41
	Submodule Functions .....	42
5.	Pollution Costs Submodule.....	47
	Submodule Objective.....	47

Submodule Methodology .....	47
Submodule Functions .....	49
6. Capacity Expansion Submodule.....	56
Submodule Objective.....	56
Submodule Methodology .....	56
7. Electric Generation Dispatch Submodule.....	66
Submodule Objective.....	66
Submodule Methodology .....	68
Submodule Functions .....	76
8. Fuel Usage Submodule .....	97
Submodule Objective.....	97
Submodule Methodology .....	97
Submodule Functions .....	99
9. Pollution Submodule .....	104
Submodule Objective.....	104
Submodule Functions .....	104
10. Retail Power Costs Submodule.....	113
Submodule Objective.....	113
Submodule Methodology .....	114
Submodule Functions .....	115
11. Electric Price Submodule.....	120
Submodule Objective.....	120
Submodule Functions .....	122
12. Electricity Supply Sector Calibration.....	126

## Figures and Tables

---

### Figures

Figure 1. ENERGY 2100 Model Linkages .....	6
Figure 2. Electric Supply Submodules .....	8
Figure 3: Linkages in Electric Supply Module.....	9
Figure 4. Sample Transmission Network in ENERGY 2100.....	10

Figure 5: Exogenously-specified characteristics of generating units ..... 11

Figure 6: Load Curve in Electric Supply ..... 17

Figure 7. Load Curve Submodule Diagram of Inputs and Outputs ..... 18

Figure 8. Sample LDCECC - 2010 Total Canada Electricity Sales ..... 19

Figure 9: Function ElecLoadCurvesAndSales in ELoadCurve.jl..... 20

Figure 10: Contract Development in Electric Supply ..... 31

Figure 11: Contract Development Submodule - Inputs and Outputs ..... 33

Figure 12: Function DevelopContracts in EContractDevelopment.jl ..... 34

Figure 13: Retail Purchases in Electric Supply..... 40

Figure 14. Retail Purchases Submodule Diagram of Inputs and Outputs..... 41

Figure 15: Function RetailPurchases in ERetailPurchases.jl..... 42

Figure 16: Pollution Costs in Electric Supply ..... 47

Figure 17. Pollution Submodule Diagram of Inputs and Outputs..... 48

Figure 18: Function Part1 in EPollution.jl ..... 49

Figure 19: Capacity Expansion in Electric Supply ..... 56

Figure 20. Capacity Expansion Submodule Diagram of Inputs and Outputs ..... 59

Figure 21. Capacity Expansion Routines – Capacity Requirements ..... 61

Figure 22. Capacity Expansion Routines – Capacity Costs ..... 62

Figure 23. Electric Capacity Expansion Routines – Capacity Initiated ..... 64

Figure 24: Generation Dispatch in Electric Supply ..... 66

Figure 25. Generation Dispatch Submodule Diagram of Inputs and Outputs ..... 67

Figure 26. Sample Transmission Network in ENERGY 2100..... 69

Figure 27. Linear Program Inputs and Outputs..... 70

Figure 28 - Linear Program objective function and constraints ..... 72

Figure 29. Function DispatchElectricity in EDispatch.jl..... 77

Figure 30: Fuel Usage in Electric Supply ..... 97

Figure 31. Fuel Usage Submodule Diagram of Inputs and Outputs..... 98

Figure 32: Function RunFuelUsage in EFuelUsage.jl ..... 99

Figure 33. Pollution in Electric Supply..... 104

Figure 34. Function Part2 in EPollution.jl..... 105

Figure 35. Retail Power Costs in Electric Supply ..... 113

Figure 36. Retail Power Costs Submodule Diagram of Inputs and Outputs ..... 114

Figure 37. Function ElectricCosts in ERetailPowerCosts.jl ..... 115

Figure 38: Electric Price in Electric Supply ..... 120

Figure 39. Electric Price Submodule Diagram of Inputs and Outputs ..... 121

Figure 40: Function Finance in ElectricPrice.jl ..... 122

**Tables**

Table 1: Submodule File Names..... 12

Table 2: Description of Load Curve Functions ..... 21

Table 3: Input and Output Variables - LoadsPeakAverageMinimum..... 24

Table 4: Input and Output Variables - ElectricSales..... 25

Table 5: Input and Output Variables - ReCoPeakAverageMinimumLoads ..... 26

Table 6: Input and Output Variables - ReCoMaximumBaseloadPoint..... 26

Table 7: Input and Output Variables - PowerFractions ..... 27

Table 8: Input and Output Variables - SystemAnnualPeakAverageMinimumLoads ..... 28

Table 9: Input and Output Variables - LoadsInTimeIntervals ..... 29

Table 10: Input and Output Variables - HourlyLoads..... 30

Table 11: Description of Contract Development Functions..... 34

Table 12: Input and Output Variables - ContractFractionOfCapacityAndLoad..... 36

Table 13: Input and Output Variables - CapacityUnderContract..... 37

Table 14: Input and Output Variables - ContractEnergyLimit..... 37

Table 15: Input and Output Variables - PeakingEnergyLimit..... 38

Table 16: Input and Output Variables - CapacityAndEnergyCosts..... 38

Table 17: Input and Output Variables - AddExogenousContracts ..... 39

Table 18: Description of Retail Purchases Functions..... 42

Table 19: Input and Output Variables - LoadContractVariables ..... 44

Table 20: Input and Output Variables - DispatchContracts ..... 44

Table 21: Input and Output Variables - UnloadContractVariables ..... 45

Table 22: Input and Output Variables - PurchasesSummation..... 46

Table 23: Description of Pollution Costs Functions ..... 49

Table 24. Description of Functions within Generation Dispatch ..... 78

Table 25: UnitCosts Function - Input and Output Variables ..... 82

Table 26: TransmissionCosts Function - Inputs and Output Variables ..... 83

Table 27: Inputs and Output Variables - AllocatePeakHydroAvailability..... 83

Table 28: Input and Output Variables - IncreaseLoadsForRecharging ..... 85

Table 29: Input and Output Variables - UnitCapacityAvailability ..... 86

Table 30: Input and Output Variables - UnitBids ..... 88

Table 31: Input and Output Variables - ReserveUnits ..... 89

Table 32: Input and Output Variables - RunLPForDispatch ..... 92

Table 33: Input and Output Variables - RechargingRequirements ..... 93

Table 34: Input and Output Variables - PollutionForPolicy ..... 93

Table 35: Input and Output Variables - ConstrainAvailabilityBasedOnPollution ..... 94

Table 36: Description of Functions within Fuel Usage..... 99

Table 37. Description of Functions within Pollution..... 105

Table 38. Description of Functions within Retail Power Costs ..... 115

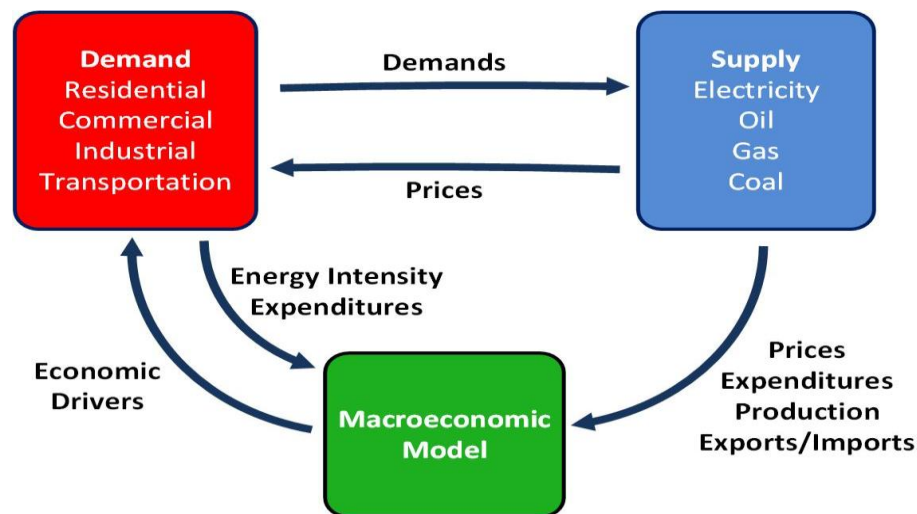
Table 39. Description of Functions within Electric Price..... 122



## 1. Introduction

ENERGY 2100 simulates the North American energy system providing long-range energy and emissions forecasts and the ability to analyze energy-related policies. Three sectors make up the core of ENERGY 2100 – a demand sector, a supply sector, and a macroeconomic sector. The relationship among these three sectors is shown in Figure 1. The energy supply sector’s major modules (electricity, oil, gas, and coal) receive energy demands from the demand sector and, in turn, send resulting energy prices back as input to the demand sector. When ENERGY 2100 is linked to a macroeconomic model, the supply sector sends energy prices, expenditures, production, imports and exports to the macroeconomic model. The macroeconomic model combines these supply sector inputs with energy intensity and expenditure inputs from the demand sector to estimate economic impacts which then are sent back to the demand sector as drivers for energy demand.

**Figure 1. ENERGY 2100 Model Linkages**



ENERGY 2100’s supply sector is made up of several major modules, including electricity, oil and natural gas, refinery, hydrogen, and coal supply in addition to other smaller supply modules, such as ethanol, and land-fill gas. This documentation specifically focuses on the electric supply module – both its methodology and model code.

The electric supply module endogenously simulates the capacity expansion, generation, fuel usage, transmission and distribution, emissions, and pricing of electricity. For each year of model execution, this module receives energy demands from the demand module, converts these demands to a system load duration curve, builds new generation capacity based on load growth at wholesale prices, dispatches individual

---

generating units to meet demand, generates a system marginal price, simulates transmission flows, calculates generation, fuel use and emissions for each unit, simulates the cost of power to each retail company, or load serving entity (LSE), through contracts and wholesale power purchases, and calculates retail electric prices from LSE power costs, LSE other costs and local regulatory factors. The model dispatches plants according to specified rules, whether they are optimal or heuristic, and simulates transmission constraints when determining dispatch. The dispatch routine selects critical hours along seasonal load duration curves as a way to provide a quick but accurate determination of system generation. The key outputs include electricity prices which are sent to the demand module, fuel consumption which is sent to the oil, gas, and coal supply modules and expenditures, production, imports and exports which are sent to the macroeconomic module along with prices.

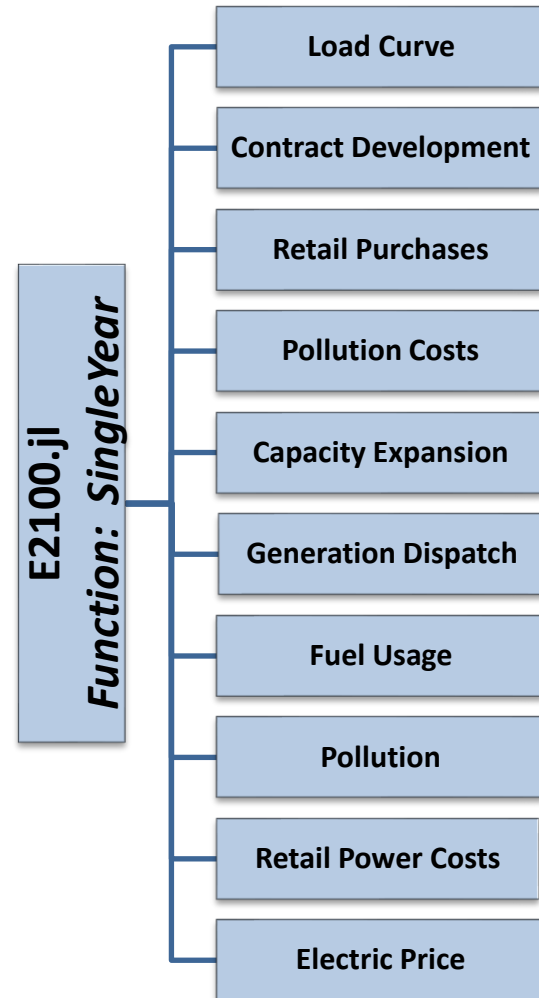
The electric supply module is comprised of ten submodules, including 1) load curves, 2) contract development, 3) retail purchases, 4) pollution costs, 5) capacity expansion, 6) generation dispatch, 7) fuel usage, 8) pollution, 9) retail power costs, and 10) electric prices.

Figure 2 provides a list of the submodules that make up the electric supply sector and are shown in the order of model execution. Each of these submodules is executed from inside ENERGY 2100's *SingleYear* Function which is housed in the file named *E2100.jl*. The objective of each of the electric supply submodules can be summarized as follows:

1. The **load curve** submodule processes electricity demand obtained from the demand sector to construct load curves which are needed for calculations within the Retail Purchases, Capacity Expansion, Generation Dispatch, and Electric Price submodules.
2. The **contract development** submodule determines the capacity and energy available for contracts made between retail and generation companies along with their associated costs.
3. The **retail purchases** submodule assigns the amount of capacity and energy that get purchased by retail companies from contracts and the wholesale market.
4. The **pollution costs** submodule calculates the cost of emissions from various fuels and plant types to be used in comparison options for capacity expansion.

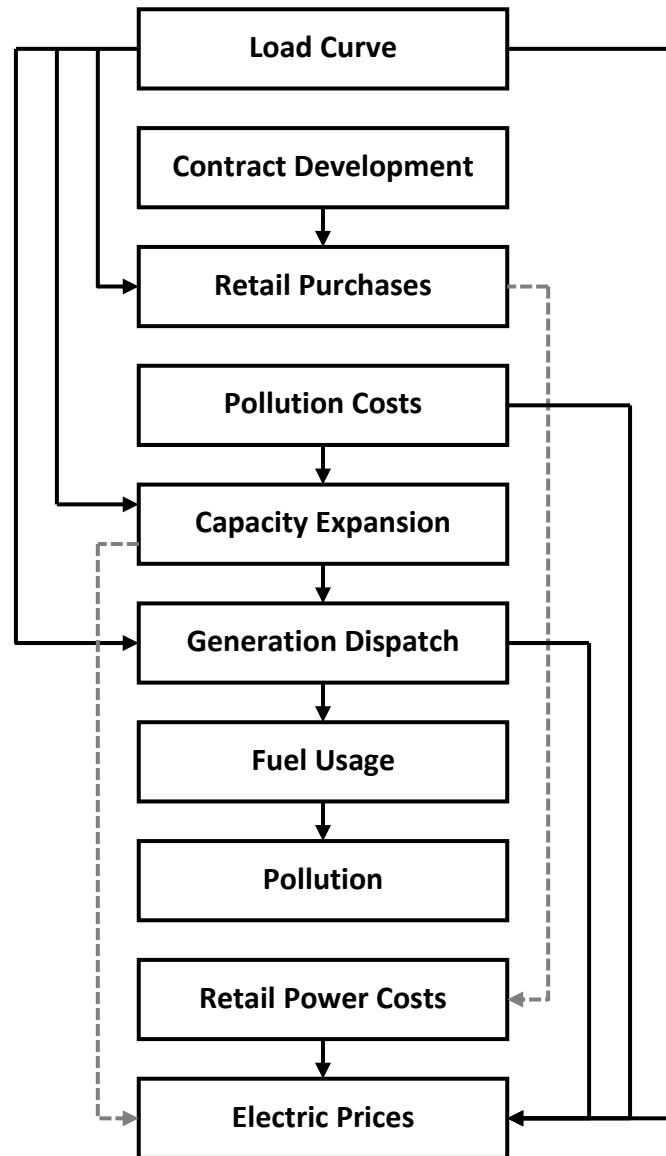
5. The **capacity expansion** submodule projects the construction of new generating plants based on meeting a reserve margin and/or based on wholesale prices. Outputs include capacity under construction and total unit generating capacity which is used as input to the generation dispatch module.
6. The **generation dispatch** submodule dispatches the available generating units using pre-specified dispatch assumptions and minimizing the overall cost to the system. Outputs include generation dispatched, transmission flows, wholesale prices, imports, and exports.
7. The **fuel usage** submodule summarizes the outputs from the electric generation dispatch into various aggregations for use by other modules within the model.
8. The **pollution** submodule calculates the emissions based on the fuel usage dispatched in the generation dispatch submodule.
9. The **retail power costs** submodule calculates the cost of the retail purchases using inputs from the costs determined in the contract development submodule combined with the amount of capacity and energy purchased.
10. The **electric pricing** submodule determines the electricity prices for the current year of execution for each retail company based on the unit cost, delivery charge and other adjustments.

Figure 2. Electric Supply Submodules



The information transfers between the electric supply submodules are shown in Figure 3. Load curve outputs are sent as input to retail purchases, capacity expansion, generation dispatch, and electric prices. Load curve outputs are sent as input to retail purchases, capacity expansion, generation dispatch, and electric prices. The capacity and energy available for contracts from contract development are sent to retail purchases. The actual amount of power purchased from contracts and the spot market from retail purchases are sent to retail power costs. Outputs from pollution costs are sent to capacity expansion and electric prices. The unit capacity outputs from capacity expansion are sent to generation dispatch. Marginal prices and generation dispatched calculated in generation dispatch are sent to fuel usage and electric prices. Outputs from fuel usage are inputs to pollution. Finally, the retail power costs outputs are used as inputs to electric prices along with the pollution costs outputs and marginal prices outputs from generation dispatch.

**Figure 3: Linkages in Electric Supply Module**



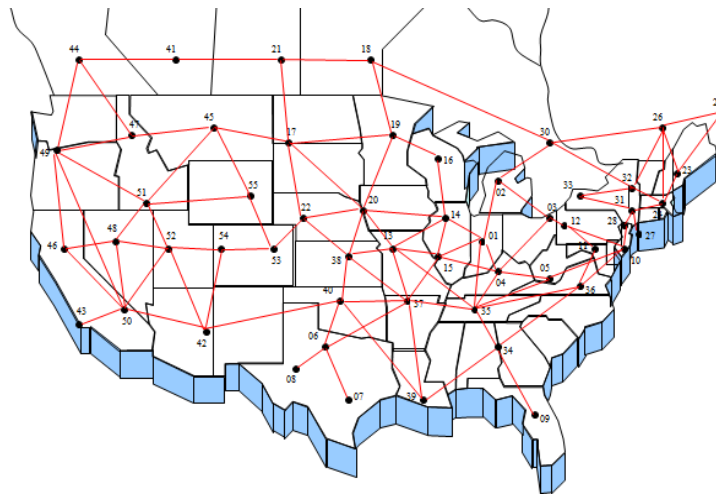
## Structure of Electric System

The electric supply module contains several building blocks including generation units, generating companies, retail companies (or load serving entities LSEs), transmission demand centers (geographic areas) and power contracts. The electric system in ENERGY 2100 is simulated as a set of nodes connected by transmission lines. The electrical demand and electric generating units are located on the nodes. Each node consists of its own unique, specified demand, which is met by a pool of resources consisting of the

generating units as well as emergency generation. The interpretation of emergency power includes brownouts, non-utility generation purposed to feeding the grid (such as hospital generation), and other non-standard generation. Each resource has an associated cost, a bid price and a capacity.

Nodes are connected by transmission lines containing a variety of line capacities, transmission efficiencies, and wheeling costs. The actual transmission nodes and lines defined in ENERGY 2100 can be customized; however, the standard version of the model has 110 nodes represented across North America which are typically aggregated into a smaller number of nodes of interest. Figure 4 illustrates an example of a transmission network represented in ENERGY 2100.

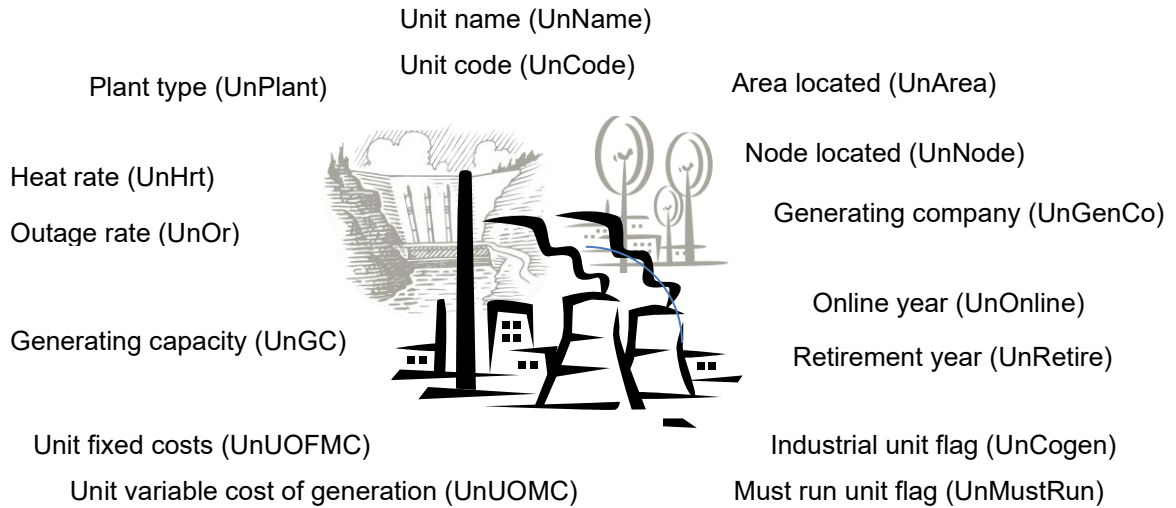
**Figure 4. Sample Transmission Network in ENERGY 2100**



A generating unit is able to send electricity to neighboring nodes via the transmission lines. Each of these inputs is used to find the optimal solution of generation, flows, and the resulting nodal prices. The entire geographic area of the model is dispatched as a single system.

Each electric generating unit is exogenously-specified with defining characteristics. These characteristics include a name, the node in which they are located, the type of plant, the heat rate, the online and retirement years of the unit, and the generating capacity. Additionally, units are assigned an outage rate, a fixed cost, and a variable cost. The units may be flagged as “industrial”, meaning it is self-generating and does not provide generation to the electric grid. Units may also be flagged as “must run”, meaning the unit always runs. The exogenously-specified characteristics of the electric generating units are summarized in Figure 5 below.

**Figure 5: Exogenously-specified characteristics of generating units**



Each individual unit can be simulated although smaller units can also be aggregated by type and transmission mode. Currently the number of electric units represented in the model consists of roughly 2,100 Canadian units and an aggregate representation of US at just over 600 units.

The number of generating companies and retail companies (LSEs) simulated varies by model implementation but range from 10 to 40.

The generating companies have a set of generating units, a capacity expansion strategy, a bidding strategy, contracts with retail companies (LSEs), generation, fuel usage, and emissions. The retail companies have contracts with generating companies, sales to demand areas, and a retail cost structure. The contracts have a generating company and a retail company, a capacity, a variable cost, and a fixed cost. The transmission nodes have generating units and demands from the geographic areas. The demand centers are geographic areas with relationships with transmissions nodes and retail companies. The demand center prices are a function of the retail company prices serving their area.

### Electric Supply Source Code

The files that contain the ENERGY 2100 source code that simulate each of the electric supply submodules are named with an 'E' prefix and are housed within the Engine subdirectory. The file name of each electric supply submodule is listed in Table 1.

**Table 1: Submodule File Names**

File Name	Submodule Model Code
<b>ELoadCurve.jl</b>	Electric load curve submodule
<b>EContractDevelopment.jl</b>	Contract development submodule
<b>ERetailPurchases.jl</b>	Retail purchases submodule
<b>EPollution.jl</b>	Pollution submodule, pollution costs submodule
<b>ECapacityExpansion.jl</b>	Capacity expansion submodule
<b>EDispatch.jl</b>	Electric generation dispatch submodule
<b>EFuelUse.jl</b>	Fuel use submodule
<b>ERetailPowerCosts.jl</b>	Retail power costs submodule
<b>ElectricPrice.jl</b>	Electric pricing submodule

### Load Curve Submodule

The development of the load curve begins in the demand sectors where enduse demands for electricity are multiplied by a load shape factor to produce enduse electric demands for the peak, average and minimum points on the load duration curve for each enduse, economic category and geographic area. These enduse load curves are combined with street lighting and miscellaneous electric demands to produce a system load curve for each area.

The load curve for each economic category is allocated to each retail company using historical relationships or retail prices if retail competition is active. The load curve is also allocated from each geographical area to the transmission nodes. A transmission loss factor is applied to the electric sales from the demand sectors. Once the load curves are allocated to transmission nodes, the peak energy demands for each of six time periods are computed. These six time periods for each season are used to dispatch the generating units. The six time periods are peak, near peak, high intermediate, low intermediate, high base load, and low base load.

### Contract Development Submodule

The purpose of the contract development submodule is to set up the specifications of potential retail company contracts with generating companies. The specifications include minimum and maximum amounts of capacity to be purchased by each retail company, the energy available for each contract, and the cost of capacity and energy purchased through contracts.

## Retail Purchases Submodule

With the contract costs, capacity, and energy limits determined, the model is now ready to create contracts to meet electric demands. In this section, retail companies have the option to purchase electricity either from contracts with generating companies or through the spot market.

Contracts for each retail company are dispatched starting with the lowest cost contracts until all electricity needs are met or no more available contracts exist in which case retail companies purchase electricity from the spot market. The amount of electricity purchased either from contracts or the spot market is saved for use in other parts of the model, including calculating the final electric price.

## Pollution Costs Submodule

The pollution costs submodule determines the amount of emission reductions required by electric utilities and applies reductions at the individual unit level. Marginal and average emissions coefficients are recalculated for the sector as a whole along with an estimation of the costs of the reduction based on abatement curves. The submodule offers the user different options (such as setting the cost of emissions permits or setting a specific reduction target) to determine how reductions for each unit are calculated.

## Capacity Expansion Submodule

The decision to build new generation capacity involves many factors. ENERGY 2100 includes a variety of algorithms to simulate the decision to build new capacity. These algorithms include information on wholesale prices, desired reserve margins, the cost of new capacity (construction, operating, and environmental costs), political and social preferences, regulations, standards, and subsidies. The model is adjusted and customized to simulate the decision-making process at a particular time and location.

One method of building new capacity is to build whenever capacity is lower than a desired reserve margin. The model forecasts the expected demand a few years into the future applying an expected growth rate to current year's peak demand. The number of years which are forecasted depends on the construction time of the units being built. The forecasted peak demand is compared to the existing capacity (derated for intermittent sources), capacity to be retired, and capacity under construction to decide the amount of new capacity.

An alternative method of building new capacity is to build based on the wholesale price of power. As the wholesale price rises, new capacity will be constructed even if the reserve margin is higher than “desired”. Conversely, if the wholesale price is low, then new capacity will not be constructed even if it is “needed”.

After forecasting the construction of new capacity, the submodule passes total available capacity to the electricity generation dispatch submodule and new capacity expenses to the electric pricing submodule.

## Generation Dispatch Submodule

The dispatching of electric units is simulated using a linear program (LP) where the objective function minimizes the cost of production subject to the constraints of meeting demand using available capacity within the limits of the transmission system. The production costs include the bid prices of the units and the cost of moving power across the transmission network. Generating units are derated based on the unit outage rates before they are bid into the system. The full derated capacity is generally bid into the system. Most units are bid at their marginal costs although other options are available. Some units (nuclear) may have a reported fuel cost which overstates their marginal costs. These units can be bid at less than their marginal costs. Fixed cost may also be included in the bid, if desired. Bids can vary by period. Market bids can be set to maximize generating company income. The generating companies modify their bids (up and down) and monitor their net revenue. If net revenues go up, then the generating company continues to modify their bids.

Transmission flows are a function of the dispatch algorithm; however exogenous “contract flows” can be added to force the flow of electricity between specific nodes. This may be needed when significant amount of storage hydro is available since the allocation of the water includes consideration of “contract flows”.

### Imports and Exports

Electricity imports and exports are calculated as part of the LP in the electric generation dispatch submodule. Loading on the transmission lines between nodes comes out of the LP as a variable called HDLLoad. The Function *Flows* in the electric generation submodule stores the flows between the US and Canada obtained from the LP to an import variable (ExpPurchases) and an export variable (ExpSales).

Imports and exports can be specified exogenously (using the variable HDXLoad) for instances where there are known levels of imports or exports, such as through contracts. The exogenously-specified values are added to the flows determined by the LP.

## Self-Generation/Cogeneration

The types of electric generating units represented in the model include all significant generators, including both utility and industrial boilers and generators. By contrast, reported electricity consumption information tends to be based on metered electricity sales, and as such are net of self-generation. Total electricity consumption and generation will generally be slightly higher than reported electricity sales. It is therefore important in calibrating the model with historic electricity consumption that existing generation used as industrial or self-generation be appropriately identified.

The method of calculating cogeneration demands in the model is specified by a switch and may take different forms. In one option, the demand module endogenously calculates sector-level cogeneration demand and costs. Another option available is to calculate cogeneration at the individual unit level rather than the sector level. If this option is selected, the unit-level cogeneration is determined in the electric dispatch submodule is aggregated up to the sector level in the demand module.

When unit-level cogeneration is used, the industrial cogeneration units are represented in the list of electric generating units. Each of the industrial generating units has a capacity, the amount of cogeneration is determined as the portion of that capacity remaining after own use and outage rates. The unit demand then is determined from the capacity available. Then the cogeneration routine in the demand module gets called and aggregates the unit cogeneration demand up to the sector level totals.

## Fuel Usage Submodule

The objective of the fuel usage submodule is to aggregate and summarize several output variables from the previously executed electric supply submodules. In particular, fuel consumed by electric units based on the generation dispatched in the electric generation dispatch submodule is summarized into specific fuel usage variables for use in other submodules, such as pollution.

## Pollution Submodule

The objective of the pollution submodule is twofold. First, the submodule calculates the emissions from electricity generation on the unit level. Second, the submodule applies various pollution related policies. These include but are not limited to offsets, emissions-related capital retirements and construction, Alberta CASA policies, and the assignment of gratis permits. Emissions are a function of fuel usage and an emission factor. The emission factor depends on the emissions reduction technology.

## Retail Power Costs Submodule

The purpose of the retail power costs submodule is to determine the costs of the retail companies' power purchases. With costs and the amount of power purchased from contracts and the spot market determined, the Function *ERetailPowerCosts* calculates the final cost of purchasing power from the generating company for each retail company.

## Electric Pricing Submodule

The purpose of the electric price submodule is to calculate the final price of electricity for each retail company based on the unit cost, delivery charge, and other adjustments. The price calculated in this submodule is the retail price of electricity – the final price that is passed to the customer purchasing power. The retail electric price is key as an input for the decision-making process of consumers in the demand sector.

## 2. Load Curve Submodule

*ELoadCurve.jl* contains the source code and Functions that make up the load curve submodule within the electric supply module. The load curve submodule's main Function is named *ElecLoadCurvesAndSales* and is called from SingleYear Function inside E2100.jl. It is the first submodule called in the electric supply module.

### Submodule Objective

The objective of the load curve submodule is to aggregate and transform outputs from the demand sector into forms more useful to the electric sector. Specifically, hourly load demands which are used in the dispatch of electric generation; aggregate peak, average, and minimum loads used in determining expansion of generation capacity; and electric sales useful in the calculation of electricity prices.

The key input and output variables of the load curve submodule are shown in Figure 7.

Figure 6: Load Curve in Electric Supply

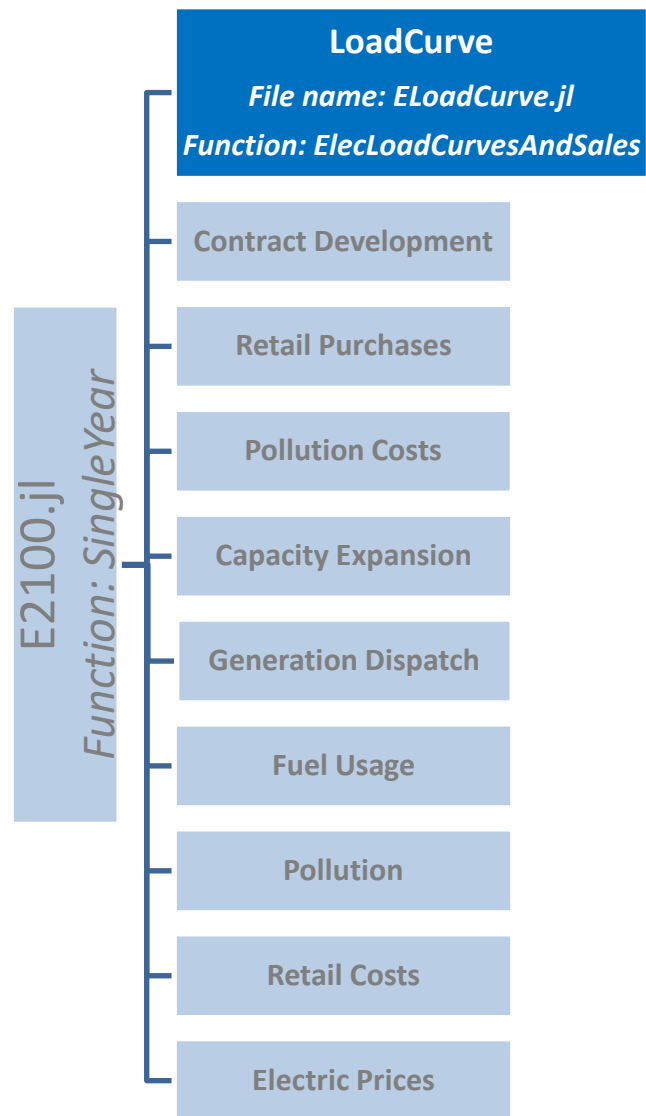
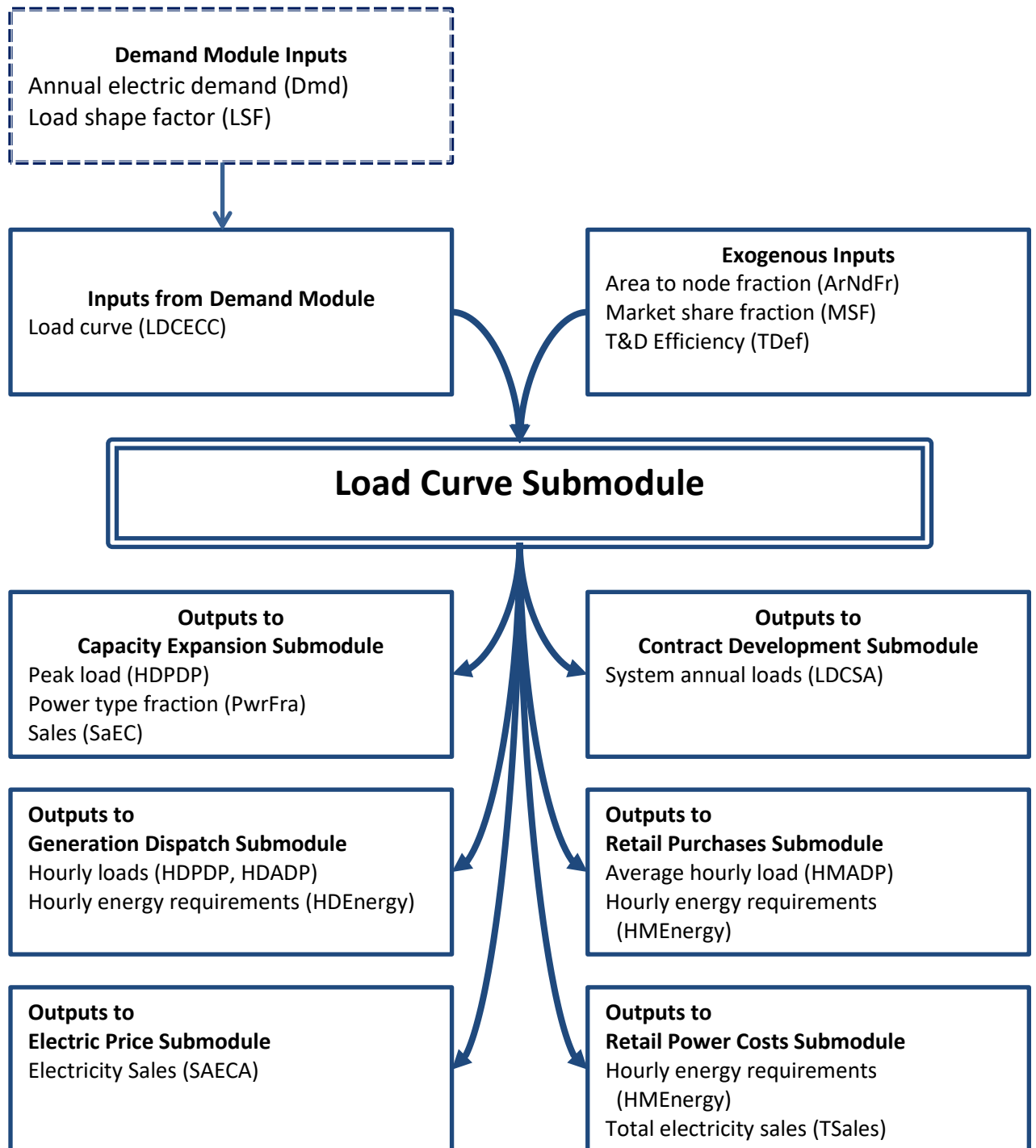


Figure 7. Load Curve Submodule Diagram of Inputs and Outputs

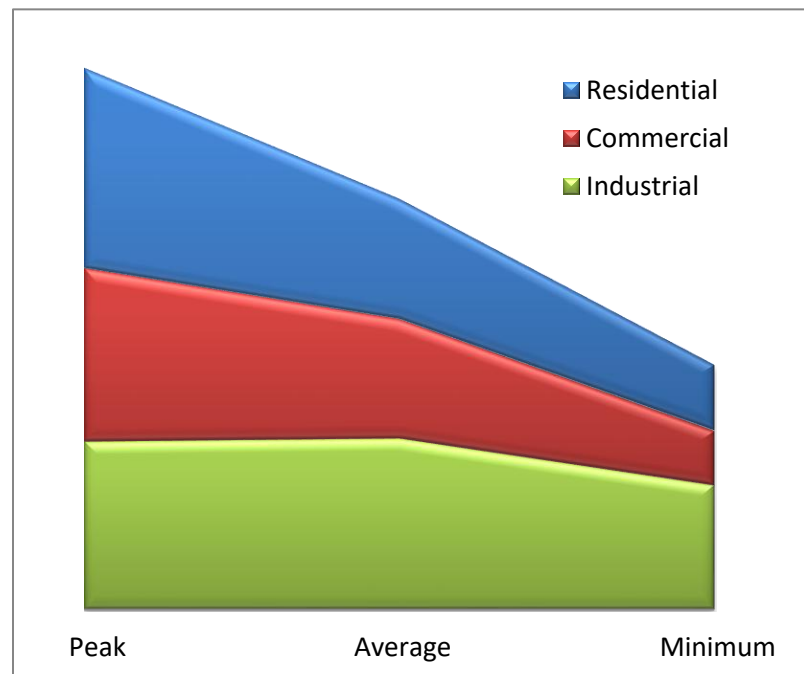


## Submodule Methodology

The development of the load curve begins in the demand modules of the model where enduse demands (Dmd) for electricity are multiplied times a load shape factor (LSF) to produce enduse electric demands (LDCECC). These represent the contribution to peak, average and minimum points on the load duration curve for each enduse, economic category and geographic area. These enduse load curves are combined with a street lighting and miscellaneous electric demands to produce a system load curve for each area.

The load curve submodule in the electric supply module takes these loads for each economic category (LDCECC) and, combining them across industries, allocates the total electric load to each retail company using historical relationships or retail prices if retail competition is active. The load curve is also allocated from each geographical area to the transmission nodes. The load curves based on geographical area and transmission node are used to calculate the total

**Figure 8. Sample LDCECC - 2010 Total Canada Electricity Sales**



electricity sales. These are chiefly used in pricing and as direct outputs. Next, a transmission loss factor is applied to the accumulated load curves in order to create the system load curve (LDCMS). Specifically, this is the system load curve for electricity marketers and retailers, which is a distinction from earlier incarnations of the model where different kinds of electric companies could have their own load curves.

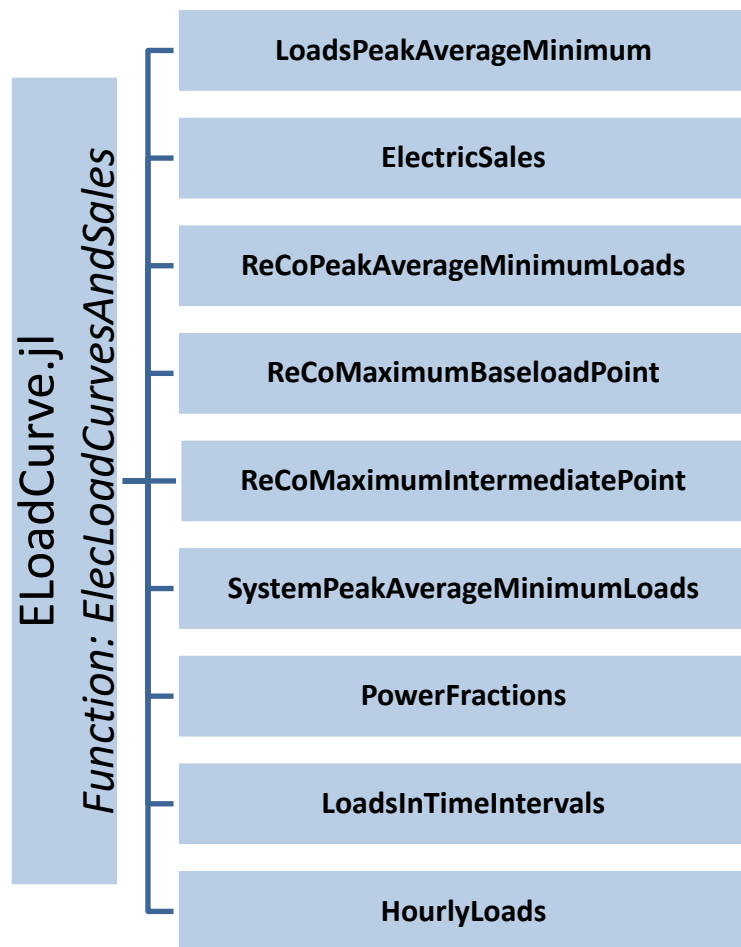
The load curve is then used to create variables relevant to both capacity expansion and generation dispatch. For capacity applications, the load curve submodule creates power fractions (PwrFra), which calculates the percentages of loads in peak, intermediate, and baseload categories. Due to a desire for more granular pricing information, generation dispatch uses more detailed information on load demands. This begins with use of six time periods for each season—peak, near peak, high intermediate, low intermediate,

high base load, and low base load. For each time period, peak and average loads are calculated, as well as the energy requirements.

### Submodule Functions

The Functions described in this section pertain only to those load curve Functions that are housed in the electric supply sector. As discussed in above, at the point when the electric supply sector is executed, the demand sector has allocated the enduse demands (Dmd) to hours, months, and time periods into a load curve (LDECC) using load shape factors (LSF) applied to the demand. The electric load curve submodule in the electric sector performs further transformations of the load curve into retail companies, area, and node splits needed by the other submodules within the electric supply sector, such

Figure 9: Function ElecLoadCurvesAndSales in ELoadCurve.jl



as capacity expansion, generation dispatch, and electric prices. Above, Figure 9 shows the Functions that make up the electric load curve submodule.

Note that peak load is lowered by cogeneration. However, the peak load is not higher due to losses (losses are not included in peak load). We do add back cogeneration and add in losses before we calibrate to the peak, but they are not included in the input peak load variable (xPKLoad).

Table 2 provides a summary description of each of the Functions within the electric load curve submodule along with the key outputs for each.

Table 2: Description of Load Curve Functions

Function Description	Key Outputs
<p><b>1. LoadsPeakAverageMinimum</b></p> <p>Takes the electric load curve from the demand sector (LDCECC) and allocates the loads using the Area/Node split fractions (ArNdFr), the input market share (XMSF) and the T&amp;D efficiency (TDEF) into intermediate version of load curve (LDCDN), Area Electric Load Curve (LDCAr), Node Electric Load Curve (LDCNd), and the marketer system load curve (LDCMS). Area and Node LoadCurves are used in Function <i>ElectricSales</i>; Marketer System Loads (LDCMS) are principally used in Functions <i>ReCoPeakAverageMinimumLoads</i> and <i>LoadsInTime Intervals</i>.</p>	<p>Electric Load Curve (LDCAr), Node Electric Load Curve (LDCNd), and the Marketer System Load Curve (LDCMS)</p>
<p><b>2. ElectricSales</b></p> <p>Calculates the Sales from various Loads; Node Sales (SAECN) from Node Loads (LDCNd), Area Sales (SAECA) from Area Loads (LDCAr), Sales by Economic Category (SAEC) {a different one from[R/C/I/T]Loads.jl} from Node Sales, Class Sales (SACL) from Sales by Economic Category, and Total Sales (TSALES) from Class Sales.</p>	<p>The several sales variables (SAECN, SAECA, SAEC, SACL, TSales) are used in electric price calculations and as outputs.</p>
<p><b>3. ReCoPeakAverageMinimumLoads</b></p> <p>Calculates Minimum, Average, and Peak Demand Loads (MDP, ADP, and PDP) from the marketer system loads (LDCMS).</p>	<p>The loads (MDP, ADP, and PDP) are used for load points below and in the Function <i>PowerFractions</i>.</p>
<p><b>4. ReCoMaximumBaseloadPoint</b></p> <p>Calculates the Maximum Baseload Point on the LDC (MBP) from the Area Loads.</p>	<p>The Maximum Baseload Point (MBP) is used in the Function <i>PowerFractions</i>.</p>

Function Description	Key Outputs
<p><b>5. ReCoMaximumIntermediatePoint</b> Calculates the maximum intermediate load point on the LDC (MILP) from the Area loads.</p>	<p>The maximum intermediate load point (MILP) is used in the Function <i>PowerFractions</i>.</p>
<p><b>6. PowerFractions</b> Calculates the power type fractions (PwrFra) from the minimum, average, and peak demand loads (MDP, ADP, PDP). Currently, a bug switch adjusts how PwrFra is calculated from these loads.</p>	<p>Power fractions (PwrFra) are used in electric capacity expansion.</p>
<p><b>7. SystemAnnualPeakAverageMinimumLoads</b> Calculates the system annual loads (LDCSA) – that is, the highest peak loads from the entire year, lowest minimum loads from the entire year, and the mean average load across the months of the year – from the retail loads (MDP, ADP, PDP).</p>	<p>System annual loads (LDCSA) are used in contract development.</p>
<p><b>8. LoadsInTimeIntervals</b> Interpolates a load curve from the peak, average, and minimum points on the marketer loads. The curve will be sliced into six time periods; for each time period the average load, the peak demand, and the energy requirements are calculated.</p>	<p>Interval loads and energy requirements (HMPDP, HMADP, HMEnergy) are used in the Function <i>HourlyLoads</i>.</p>
<p><b>9. HourlyLoads</b> Takes the interval loads (HMPDP, HMADP) and energy requirements (HMEnergy) of <i>LoadsPerInterval</i>, as well as the Power Sold to Grid (PSoECC) and creates the hourly loads and energy requirements used in the generation dispatch submodule (HDPDP, HDADP, HDEnergy) by incorporating the energy and peak power sold to grid (PSoEnergy, PSoPDP).</p>	<p>Hourly loads and energy requirements (HDPDP, HDADP, HDEnergy) used in the generation dispatch.</p>

Each of the individual Functions summarized in the table above are described in more detail in the following sections.

### Function **LoadsPeakAverageMinimum**

This Function takes as input the electric load curve from the demand sector (LDCECC). It allocates the loads to retail companies and transmission nodes using area/node split fractions (ArNdFr), a market share fraction (MSF). The resulting load curves include an area load curve by economic category, area, and retail company (LDCAr) and a load curve by economic category, transmission node, and retail company (LDCNd); these are used as output and in sales calculations.

A system load duration curve (LDCMS) is also created from the demand sector load curve. The contributions from each economic category are summed and T&D efficiency (TDEF) is incorporated into the final load duration curve. This will be the principal load curve used in calculation of peak, average, and minimum points on the curve, power fractions, and time-period based energy requirements and peaks.

Each of these variables is used by other Functions within the electric supply sector. The LDCMS variable is used as an input to create the Area loads and the Hourly loads in Functions *ReCoPeakAverageMinimumLoads* and *LoadsInTimeIntervals*, as well as in EDispatch for emergency power availability. The LDCAr and LDCNd variables are as input to *ElectricSales* to create Area and Node sales variables, respectively.

Key equations in *LoadsPeakAverageMinimum* are:

**Key Equations:**  
*LoadsPeak*  
*Average Minimum*

*Load curve by ECC, day type, month, node, and retail company:*

$$\text{LDCNd} = \sum_{\text{Area}} (\text{LDCECC} * \text{ArNdFr} * \text{MSF})$$

*Load curve by ECC, day type, month, area, and retail company:*

$$\text{LDCAr} = \sum_{\text{Node}} (\text{LDCECC} * \text{ArNdFr} * \text{MSF})$$

*System load curve by day type, month, node, and retail company:*

$$\text{LDCMS} = \sum_{\text{Area,ECC}} (\text{LDCECC} * \text{ArNdFr} * \text{MSF} * (1 - \text{RofWSw}) / \text{TDEF})$$

$$\text{a. } \text{LDCMS}_{\text{Minimum}} = \min (\text{LDCMS}_{\text{Minimum}} , \text{LDCMS}_{\text{Average}} * 0.90)$$

Table 3 summarizes the variable names and definitions of the inputs and outputs to this Function.

**Table 3: Input and Output Variables - LoadsPeakAverageMinimum**

<b>LoadsPeakAverageMinimum Inputs and Outputs</b>
<p><b>Key inputs</b></p> <p>ArNdFr [Area,Node,Year] = Fraction of the Area in each Node (MW/MW)</p> <p>LDCECC [ECC,Hour,Day,Month,Area,Year] = Electric Load Curve from Demand Sector (MW)</p> <p>MSF [ECC,Area,Node,Area,Year] = Market Share Fraction</p> <p>TDEF [Fuel,Area,Year] = T&amp;D Efficiency (Btu/Btu)</p> <p><b>Key outputs</b></p> <p>LDCAr [ECC,Day,Month,Area,Area,Year] = Electric Load Curve (MW)</p> <p>LDCMS [Day,Month,Node,Area,Year] = Marketer System Load Curve (MW)</p> <p>LDCNd [ECC,Day,Month,Node,Area,Year] = Electric Load Curve (MW)</p>

### Function ElectricSales

This Function calculates the sales from various Loads; node sales (SaECN) from node loads (LDCNd), area sales (SaECA) from area loads (LDCAr), Sales by economic category (SaEC) from node sales, class sales (SaCl) from sales by economic category, and total sales (TSales) from class sales.

While this variable is called SaEC, it is used only in ElectricPrice.jl; elsewhere the version used is SAEC as calculated in the sector loads files RLoad.jl, CLoad.jl, ILoad.jl, and TLoad.jl. SAEC, SAECA, SAEL, and TSales are all used in price calculations in ElectricPrice.jl.

Key equations in *ElectricSales* are:

**Key Equations:**  
*ElectricSales*

Total electricity sales by ECC, retail company, and area:

$$SAECA = \sum_{\text{Month}} (\text{LDCAr} * \text{Hours})/1000$$

Total electricity sales by ECC, retail company, and node:

$$SAECN = \sum_{\text{Month}} (\text{LDCNd} * \text{Hours})/1000$$

Total electricity sales by ECC and retail company:

$$SAEC = \sum_{\text{Node}} SAECN$$

Total electricity sales by class and retail company:

$$SAEL = \sum_{\text{ECC}} (SAEC * \text{ECCCLMap})$$

Total electricity sales by retail company:

$$TSALES = \sum_{Class} SACL$$

Table 4 summarizes the variable names and definitions of the inputs and outputs to *ElectricSales*.

**Table 4: Input and Output Variables - ElectricSales**

<b><i>ElectricSales</i> Inputs and Outputs</b>
<p><b>Key inputs</b>            LDCAr [ECC,Day,Month,Area,Year] = Electric Load Curve (MW)            LDCNd [ECC,Day,Month,Node, Year] = Electric Load Curve (MW)</p>
<p><b>Key outputs</b>            SAEC [ECC,Area,Year] = Electricity Sales (GWh/Yr)            SAECN[ECC,Node,Area,Year] = Electricity Sales (GWh/Yr)            SACL [Class,Area,Year] = Electricity Sales (GWh/Yr)            TSales [Area,Year] = Electricity Sales (GWh/Yr)</p>

### Function *ReCoPeakAverageMinimumLoads*

This Function calculates minimum, average, and peak demand loads (MDP, ADP, PDP) from the marketer system loads (LDCMS). These loads are used in the Functions *ReCoMaximumBaseloadPoint*, *ReCoMaximumIntermediatePoint*, and *PowerFractions*.

Key Equations in *ReCoPeakAverageMinimumLoads* are as follows:

**Key Equations:**  
*ReCoPeak*  
*Average*  
*MinimumLoads*

Annual average load by retail company:

$$ADP = \sum_{Node}(LDCMS_{Average})$$

Annual minimum load by retail company:

$$MDP = \sum_{Node}(LDCMS_{Minimum})$$

Annual peak load by retail company:

$$PDP = \sum_{Node}(LDCMS_{Peak})$$

Table 5 summarizes the names and definitions of the input and output variables to *ReCoPeakAverageMinimumLoads*.

**Table 5: Input and Output Variables - ReCoPeakAverageMinimumLoads**

<b>ReCoPeakAverageMinimumLoads Inputs and Outputs</b>
<p><b>Key inputs</b>                      LDCMS [Day,Month,Node,Area,Year] = Marketer System Load Curve (MW)</p> <p><b>Key outputs</b>                      ADP [Month,Area,Year] = Annual Average Load (MW)                      MDP [Month,Area,Year] = Annual Minimum Load (MW)                      PDP [Month,Area,Year] = Annual Peak Load (MW)</p>

**Function ReCoMaximumBaseloadPoint/ReCoMaximumIntermediatePoint**

The maximum baseload point on the LDC (MBP) and maximum intermediate load point on the LDC (MILP) are calculated from the minimum, average, and peak demand loads (MDP, ADP, PDP). MBP and MILP are used in the Function *PowerFractions*.

Key equations in *ReCoMaximumBaseloadPoint* and *ReCoMaximumIntermediatePoint* are as follows:

**Key Equations:**  
*ReCoMaximumBaseloadPoint* &  
*ReCoMaximumIntermediatePoint*

Maximum baseload point:

$$MBP = MDP + (PDP-MDP)*(1-MBD/8760)^{max(0,(PDP-ADP)/(ADP-MDP))}$$

Maximum intermediate load point:

$$MILP = MDP + (PDP-MDP)*(1-MILD/8760)^{max(0,(PDP-ADP)/(ADP-MDP))}$$

Table 6 summarizes the variable names and definitions of the inputs and outputs to this Function.

**Table 6: Input and Output Variables - ReCoMaximumBaseloadPoint**

<b>ReCoMaximumBaseloadPoint/ReCoMaximumIntermediatePoint Inputs and Outputs</b>
<p><b>Key inputs</b>                      MDP [Month,Area,Year] = Annual Minimum Load (MW)                      ADP [Month,Area,Year] = Annual Average Load (MW)                      PDP [Month,Area,Year] = Annual Peak Load (MW)</p> <p><b>Key outputs</b>                      MBP [Month,Area,Year] = Maximum Baseload Point (MW)                      MILP [Month,Area,Year] = Maximum Intermediate Load Point (MW)</p>

## Function PowerFractions

This Function calculates the power type fractions (PwrFra) from the minimum, average, and peak demand loads (MDP, ADP, PDP). These represent the percentage of demand in the peak, intermediate, and minimum sections of the load curve. Power fraction (PwrFra) is used in ECapacityExpansion.jl, where they are used to simulate load forecasting in capacity expansion planning.

Key Equations in *PowerFractions* are as follows:

**Key Equations:**  
*PowerFractions*

Maximum baseload point:

$$PwrFra = \sum_{Month} (PwrFrAve) / \sum_{Month,Power} (PwrFrAve)$$

Where

$$PwrFrAve_{Minimum} = \sum (MBP) / (\sum (PDP))$$

$$PwrFrAve_{Intermediate} = \sum (MILP-MBP) / \sum (PDP)$$

$$PwrFrAve_{Peak} = \sum (PDP-MILP) / (\sum (PDP))$$

Table 7 summarizes the variable names and definitions of the key inputs and outputs to the PowerFractions Function.

**Table 7: Input and Output Variables - PowerFractions**

<b>PowerFractions Inputs and Outputs</b>
<p><b>Key inputs</b></p> <p>ADP [Month,Area,Year] = Annual Average Load (MW)</p> <p>PDP [Month,Area,Year] = Annual Peak Load (MW)</p> <p>MBP[Month,Area,Year] = Maximum Baseload Point on Load Duration Curve (MW)</p> <p>MDP [Month,Area,Year] = Annual Minimum Load (MW)</p> <p>MILP[Month,Area,Year] = Maximum Base Load Power (MW)</p> <p><b>Key output</b></p> <p>PwrFra [Power,Month,GenCo,Year] = Maximum Baseload Point (MW)</p>

## Function SystemAnnualPeakAverageMinimumLoads

This Function calculates the system annual loads (LDCSA) – that is, the highest peak loads from the entire year, lowest minimum loads from the entire year, and the mean average load across the months of the year – from the retail loads (MDP, ADP, PDP). The system annual loads are used in contract development for determining self-dealing fractions.

Key Equations in *SystemAnnualPeakAverageMinimumLoads* are as follows:

**Key Equations:**  
*SystemAnnual*  
*PeakAverage*  
*MinimumLoads*

Annual system minimum load:

$$LDCSA_{\text{Minimum}} = \min_{\text{Month}} (\text{MDP})$$

Annual system average load:

$$LDCSA_{\text{Average}} = \sum_{\text{Month}} (\text{ADP} * \text{Hours}) / \sum_{\text{Month}} (\text{Hours})$$

Annual system peak load:

$$LDCSA_{\text{Peak}} = \max_{\text{Month}} (\text{PDP})$$

Table 8 summarizes the variable names and definitions of the Functions’ key inputs and outputs.

**Table 8: Input and Output Variables - SystemAnnualPeakAverageMinimumLoads**

<b>SystemAnnualPeakAverageMinimumLoads Inputs and Outputs</b>
<p><b>Key inputs</b></p> <p>MDP [Month,Area,Year] = Annual Minimum Load (MW)</p> <p>ADP [Month,Area,Year] = Annual Average Load (MW)</p> <p>PDP [Month,Area,Year] = Annual Peak Load (MW)</p> <p><b>Key output</b></p> <p>LDCSA [Day,Area,Year] = Annual System Load Curve (MW)</p>

**Function LoadsInTimeIntervals**

This Function converts peak, average, and minimum loads into a load curve by six time periods, which, after adjusting for cogeneration in a later Function, will be used to determine the loads and energy requirements for the generation dispatch submodule.

A load curve is fitted to the peak, average, and minimum points on the marketer loads. These three points represent the peak, average, and minimum hours for each month, node, and retail company. Note that the definition of “month” can vary by model version – some versions have a 12-month representation, and others have a 2-season representation. The current standard version of the model (as of June 2017) uses a 2-season representation (winter and summer) for month. The load curve is fitted by interpolating the three day types—the peak, average, and minimum hours for each month – into a representative six time periods for each month/season. We do this rather than representing each of 8760 hours as a simplification and due to lack of sufficiently detailed data for the load shape behavior of detailed enduses in the demand sector.

From this load curve, we calculate the peak demand (HMPDP), average demand (HMADP), and total energy requirements (HMEnergy) for each of six time periods. The six time periods are peak, near peak, high intermediate, low intermediate, high base load, and low base load. After being adjusted for cogeneration in the Function *HourlyLoads*, these Functions will be used in the generation dispatch submodule.

The key equations in *LoadsInTimeIntervals* are as follows:

**Key Equations:**  
*LoadsInTimeIntervals*

System peak load in interval, with cogeneration:

$$\text{HMPDP} = \text{LDCMS}_{\text{Minimum}} + (\text{LDCMS}_{\text{Peak}} - \text{LDCMS}_{\text{Minimum}}) * (1 - \text{HDHrPk} / \text{Hours})^{**} \\ (\text{LDCMS}_{\text{Peak}} - \text{LDCMS}_{\text{Average}}) / (\text{LDCMS}_{\text{Average}} - \text{LDCMS}_{\text{Minimum}})$$

System energy in interval, with cogeneration:

$$\text{HMEnergy} = \text{HMEnergy}_{\text{Unadjusted}} * (\text{LDCMS}_{\text{Average}} * \text{Hours} / 1000) / \sum(\text{HMEnergy}_{\text{Unadjusted}})$$

- a.  $\text{HMEnergy}_{\text{Unadjusted}} = (\text{HMMDP} * \text{HDHours} / 1000) + \max((\text{HMEGPk} - \text{HMEGMn}) - (\text{HMPDP} - \text{HMMDP}) * \text{HDHrPk} / 1000, 0)$
- b.  $\text{HMMDP} = \text{LDCMS}_{\text{Min}} + (\text{LDCMS}_{\text{Peak}} - \text{LDCMS}_{\text{Min}}) * (1 - \text{HDHrMn} / \text{Hours})^{**} \\ (\text{LDCMS}_{\text{Peak}} - \text{LDCMS}_{\text{Average}}) / (\text{LDCMS}_{\text{Average}} - \text{LDCMS}_{\text{Minimum}})$
- c.  $\text{HMPDP} = \text{LDCMS}_{\text{Min}} + (\text{LDCMS}_{\text{Peak}} - \text{LDCMS}_{\text{Min}}) * (1 - \text{HDHrPk} / \text{HOURS})^{**} \\ \text{LDCMS}_{\text{Peak}} - \text{LDCMS}_{\text{Average}}) / (\text{LDCMS}_{\text{Average}} - \text{LDCMS}_{\text{Minimum}})$
- d.  $\text{HMEGMn} = \text{Hours} / 1000 * (\text{HMMDP})$
- e.  $\text{HMEGPk} = \text{Hours} / 1000 * (\text{HMPDP})$

System average load in interval, with cogeneration:

$$\text{HMADP} = \text{HMEnergy} / \text{HDHours} * 1000$$

Table 9 Table 9 summarizes the key input and output variables in the *LoadsInTimeIntervals* Function.

**Table 9: Input and Output Variables - *LoadsInTimeIntervals***

<b><i>LoadsInTimeIntervals</i> Inputs and Outputs</b>
<p><b>Key inputs</b> LDCMS [Day,Month,Node,Area,Year] = Marketer System Load Curve (MW)</p>
<p><b>Key outputs</b> HMADP [Node,Area,TimeP,Month,Year] = Average Load in Interval (MW) HMPDP [Node,Area,TimeP,Month,Year] = Peak Load in Interval (MW) HMEnergy [Node,TimeP,Month,Year] = Energy In Interval (GWH)</p>

## Function HourlyLoads

This Function adjusts the interval loads (HMPDP, HMADP) and energy requirements (HMEnergy) of *LoadsPerInterval* by incorporating into them cogeneration via power sold to grid (PSoECC). It creates the hourly loads and energy requirements (HDPDP, HDADP, and HDEnergy) with the energy and peak power sold to grid (PSoEnergy, PSoPDP).

The hourly loads and energy requirements (HDPDP, HDADP, and HDEnergy) are used in EDispatch.jl. The energy requirements are also used in the fuel usage submodule.

Note: The Peak Power Sold to Grid is currently zero because of an input without values, the Power Sold to Grid Load Shape Factors (PSoLSF). The new variable, CgLSF might be more accurate.

The key equations in *HourlyLoads* are as follows:

**Key Equations:** Energy in interval, without cogeneration:  
*HourlyLoads* HDEnergy =  $\sum_{Area} (HMEnergy) - PSoEnergy$

Average load in interval, without cogeneration:  
 HDADP =  $HDEnergy / HDHours * 1000$

Peak load in interval, without cogeneration:  
 HDPDP =  $\sum_{Area} (HMPDP) - PSoPDP$   
 Where:  $PSoPDP = (PSoECC * ArNdFr * PSoLSF) / 8760 * 1000$

Table 10 below shows the key input and output variables for the Function *HourlyLoads*.

**Table 10: Input and Output Variables - HourlyLoads**

<b>HourlyLoads Inputs and Outputs</b>
<p><b>Key inputs</b></p> <p>HMPDP [Node,Area,TimeP,Month,Year] = Peak (Highest) Load in Interval (MW)</p> <p>HMEnergy [Node,TimeP,Month,Year] = Energy in Interval (GWh)</p> <p>PSoECC[ECC,Area,Year] = Power Sold to Grid (GWh)</p> <p>PSoLSF[ECC,Area,TimeP,Month,Year] = Load Shape Factor for Power Sold to Grid (MW/MW)</p>
<p><b>Key outputs</b></p> <p>HDADP [Node,TimeP,Month,Year] = Average Load in Interval (MW)</p> <p>HDPDP [Node,TimeP,Month,Year] = Peak Load in Interval (MW)</p> <p>HDEnergy [Node,TimeP,Month,Year] = Energy In Interval (GWh)</p>

### 3. Contract Development Submodule

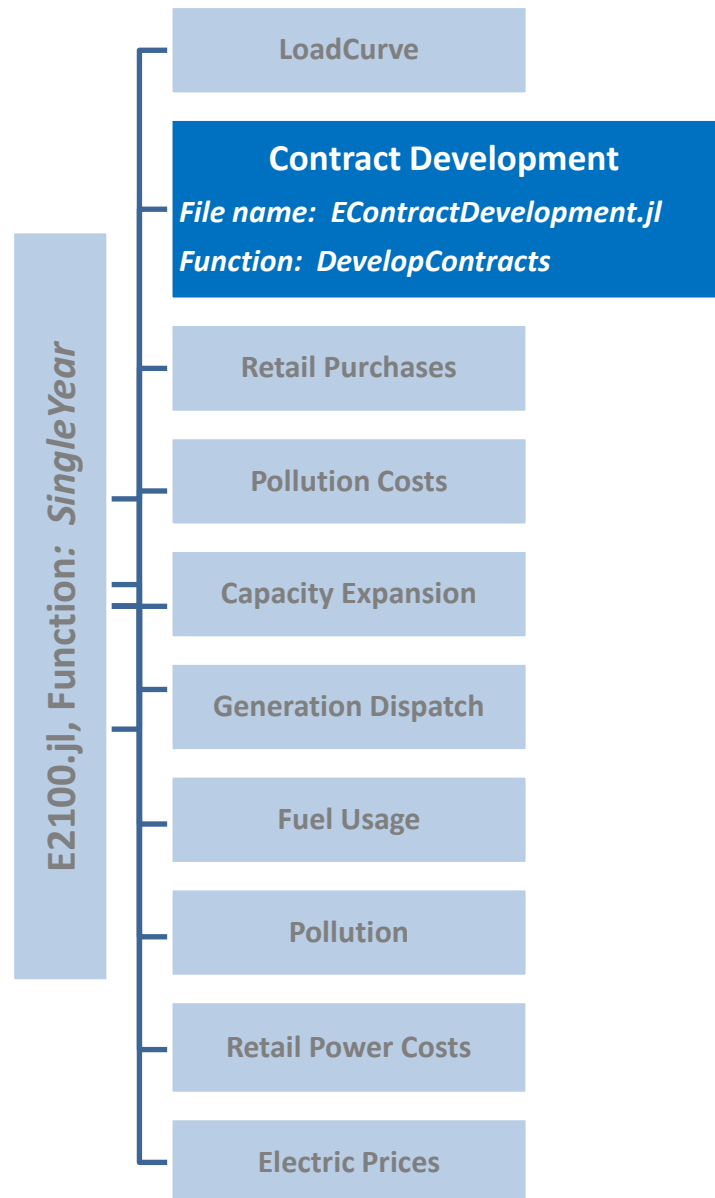
*EContractDevelopment.jl* contains the source code and Functions that make up the retail and generating company contract development submodule. The contract development submodule's main Function is named *DevelopContracts*. The call to *DevelopContracts* is initiated in the *SingleYear* Function inside *E2100.jl* by calling Function *DevelopContracts*. The contract development submodule is executed directly after the load curves have been calculated in the load curve submodule.

#### Submodule Objective

All electric power generated by utilities is sold to retail companies, who have sales to demand areas and a retail cost structure. Retail companies can choose to fulfill the needs of their customers either through contracts made with a specific generating company or by purchasing wholesale power from the market. A contract is a pre-determined purchase agreement between the generating company and retail company. The contract development submodule creates the parameters necessary to simulate the flow of power from generating companies to the customer.

Currently, ENERGY 2100 splits up the simulation of retail companies into several different submodules (contract development, retail purchases, and retail power costs),

Figure 10: Contract Development in Electric Supply



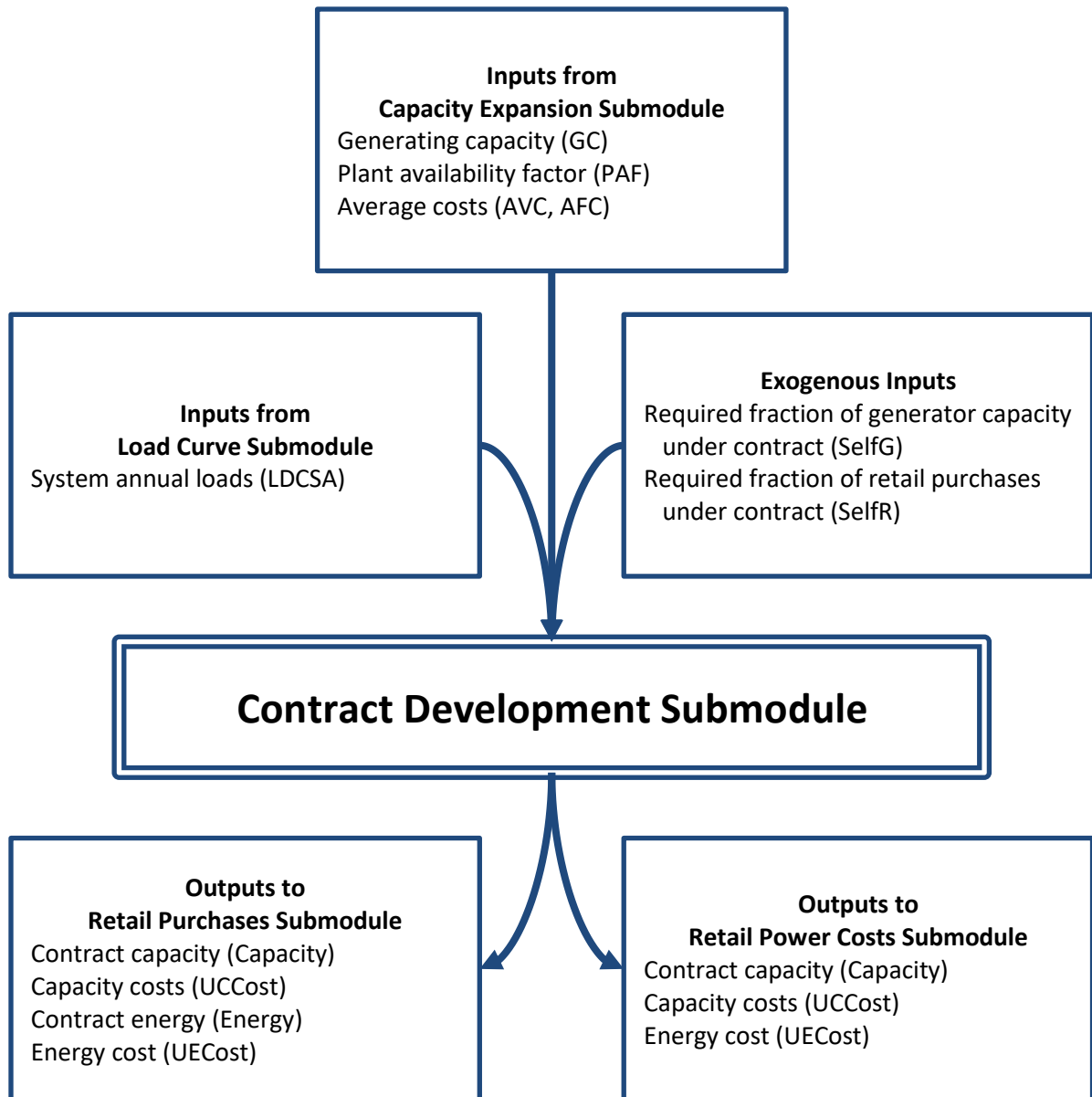
---

each reflecting a different stage of the development of retail company costs and finances. The purpose of the contract development submodule is to set up the specifications of potential retail company contracts with generating companies. The specifications include minimum and maximum amounts of capacity to be purchased by each retail company, the energy available for each contract, and the cost of capacity and energy purchased through contracts. These contract specifications will be used as input to the retail purchases submodule.

### Submodule Methodology

The contract development submodule uses capacity (GC), plant capacity factors (PCF), and average variable and fixed cost (AFC,AVC) inputs to develop contract costs (UECost,UCCost), capacity (Capacity), and generation (Energy) that are used to dispatch contracts in retail purchases. The submodule first calculates the cost of endogenous contracts using other inputs from the electric sector and then adds in any exogenously defined contracts. Figure 11 illustrates the key inputs and outputs of the contract development submodule.

Figure 11: Contract Development Submodule - Inputs and Outputs



## Submodule Functions

The Functions that make up the contract development submodule are shown in Figure 12 below. These are all housed in EContractDevelopment.jl and called from the *DevelopContracts* Function.

Figure 12: Function DevelopContracts in EContractDevelopment.jl

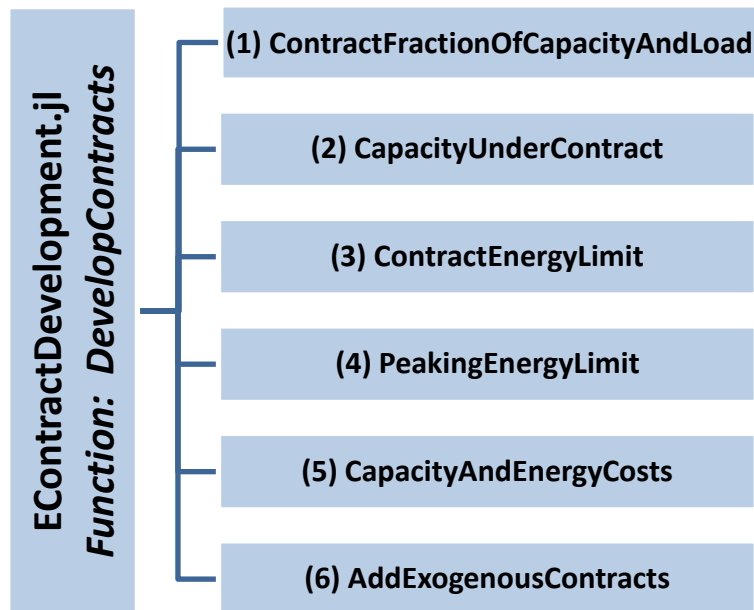


Table 11 provides a brief description of each of contract development Functions and identifies the key output variables. More detailed descriptions of these Functions are provided in the sections following the table.

Table 11: Description of Contract Development Functions

Function Description	Key Outputs
<p><b>1. ContractFractionOfCapacityAndLoad</b></p> <p>This Function determines how much power a retail company will get from contracts with generating companies as a self-dealing fraction (SelfFr). The self-dealing fraction is based on two exogenously-specified variables, therefore allows the user to determine to what extent the retail company utilizes the spot market versus bilateral contracts.</p>	<p>Self-dealing fraction (SelfFr)</p>

Function Description	Key Outputs
<p><b>2. CapacityUnderContract</b></p> <p>This Function applies the self-dealing fraction (SelfFr) determined in <i>ContractFractionOfCapacityAndLoad</i> to capacity (GC) to calculate the amount of capacity under contract (Capacity).</p>	Capacity under contract (Capacity)
<p><b>3. ContractEnergyLimit</b></p> <p>The amount of energy available for each contract (Energy) is determined by applying a plant capacity factor (PCF) to capacity under contract (Capacity).</p>	Energy available by contract (Energy )
<p><b>4. PeakingEnergyLimit</b></p> <p>The amount of energy available for contracts with peaking plants (Energy<sub>Peaking</sub>) is determined based on the plant availability factor (PAF) rather than the plant capacity factor (PCF) used in <i>ContractEnergyLimit</i>.</p>	Energy available for contracts for peaking plants (Energy <sub>Peaking</sub> )
<p><b>5. CapacityAndEnergyCosts</b></p> <p>Calculates the capacity and energy costs (UCCost, UECost) of the contracts. The costs are based on the generating companies' average fixed costs (AFC) and average variable costs (AVC).</p>	Contract capacity and energy costs (UCCost,UECost)
<p><b>6. AddExogenousContracts</b></p> <p><i>AddExogenousContracts</i> allows for the user to use either exogenously specified contract costs (XCapSw = 1) or endogenous costs (XCapSw = 2).</p>	Contract capacity and energy costs (UCCost,UECost)

### Function ContractFractionOfCapacityAndLoad

The electric market in each area modeled can contain a variety of different regulations, including whether electricity is sold exclusively through contracts, through the spot-market, or a combination of both. Each regulatory approach can result in significant difference in the final retail price, so ENERGY 2100 requires the market-type as an input to the model. This Function determines a retail company self-dealing fraction (SelfFr), which is the fraction of power that will be purchased through a contract with a generating company. This fraction is set either by a requirement to purchase power from a specific generating company (SelfG) or the fraction of the retail company peak for which they sign a bilateral contract with the generator (SelfR). Both variables are

exogenous inputs to the model, meaning that the requirement can be customized by the user. This Function allows the user to determine to what extent the retail company utilizes the spot market versus bilateral contracts.

The equations used to create the self-dealing fraction (SelfFr) are listed below.

**Key Equations:** Self-dealing fraction:  
*ContractFractionOfCapacityAndLoad*       $SelfFr = SelfCap / GC$   
 Where  $SelfCap = \text{Maximum}(GC * SelfG, \text{Minimum}(LDCSa_{Peak} * SelfR, GC))$

Table 12 summarizes the variables names and descriptions of the inputs and outputs to *ContractFractionOfCapacityAndLoad*.

**Table 12: Input and Output Variables - ContractFractionOfCapacityAndLoad**

<b><i>ContractFractionOfCapacityAndLoad</i> Inputs and Outputs</b>
<p><b>Key inputs</b></p> <p>GC [Plant,Node,GenCo,Year] = Generation Capacity (MW)</p> <p>SelfG [Area,GenCo,Year] = Minimum Fraction of GenCo Capacity Purchased by Area (MW/MW)</p> <p>SelfR [Area,GenCo,Year] = Minimum Fraction of Area Load Purchased from GenCo (MW/MW)</p>
<p><b>Key output</b></p> <p>SelfFr [Area,GenCo,Year] = Self-Dealing Fraction as Output (MW/MW)</p>

**Function CapacityUnderContract**

With the market type determined, the amount of capacity available for contracts between the generating and retail companies can be set. The amount of generating capacity under contract (Capacity) between a retail company and generating company for each plant type, time period, and month is the capacity available for each generating company (GC) times the self-dealing fraction (SelfFr). This Function calculates the capacity under contract (Capacity) which is then used as input to the retail purchases submodule. If the electric market is determined to be entirely driven by the spot market (SelfFr = 0), then the available capacity for contracts will be zero.

The key equation used in the *CapacityUnderContract* is:

**Key Equation:** Capacity under contract:  
*CapacityUnderContract*       $Capacity = GC * SelfFr$

Table 13 summarizes the key input and output variables from *CapacityUnderContract*.

**Table 13: Input and Output Variables - CapacityUnderContract**

<b>CapacityUnderContract Inputs and Outputs</b>
<p><b>Key inputs</b></p> <p>GC [Plant,Node,GenCo,Year] = Generation Capacity (MW)</p> <p>SelfFr [Area,GenCo,Year] = Self-Dealing Fraction as Output (MW/MW)</p>
<p><b>Key output</b></p> <p>Capacity [Area,GenCo,Plant,TimeP,Month,Year] = Capacity under Contract (MW)</p>

### Function ContractEnergyLimit

For most plant types, the amount of power available for each contract (Energy) is the capacity under contract (Capacity) times the capacity factor for each plant type for the generating company (PCF). The energy available for the contracts (Energy) is calculated in this Function and used as input to the retail purchases submodule which determines the amount of energy that will ultimately be dispatched.

**Key Equation:**  
*ContractEnergyLimit*      Energy available for contracts:  

$$\text{Energy} = \sum (\text{Capacity} * \text{PCF} * \text{HDHours}) / 1000$$

Table 14 summarizes the key input and output variables to *ContractEnergyLimit*.

**Table 14: Input and Output Variables - ContractEnergyLimit**

<b>ContractEnergyLimit Inputs and Outputs</b>
<p><b>Key inputs</b></p> <p>Capacity [Area,GenCo,Plant,TimeP,Month,Year] = Capacity under Contract (MW)</p> <p>PCF [Plant,GenCo,Year] = Plant Capacity Factor (MW/MW)</p>
<p><b>Key outputs</b></p> <p>Energy [Area,GenCo,Plant,Year] = Energy Limit on Contracts (GWh/Yr)</p>

### Function PeakingEnergyLimit

Gas turbine and other types of peaking plants operate sporadically, so using the capacity factor to determine the energy limit can be misleading. The amount of energy available for contracts with peaking plants (Energy<sub>Peaking</sub>) is determined based on the plant availability factor (PAF) rather than the plant capacity factor (PCF) used in *ContractEnergyLimit*.

The equation used to calculate the energy available for peaking plants is listed below.

**Key Equation:**  
*PeakingEnergyLimit*      Energy available for contracts:  

$$\text{Energy} = \sum (\text{Capacity} * \text{PAF} * \text{HDHours}) / 1000$$

Table 15 summarizes the variable names and definitions of the key inputs and outputs to this Function.

**Table 15: Input and Output Variables - PeakingEnergyLimit**

<b><i>PeakingEnergyLimit</i> Inputs and Outputs</b>
<p><b>Key inputs</b>            Capacity [Area,GenCo,Plant,TimeP,Month,Year] = Capacity under Contract (MW)            PAF [Plant,GenCo,Year] = Plant Availability Factor (MW/MW)</p>
<p><b>Key outputs</b>            Energy [Area,GenCo,Plant,Year] = Energy Limit on Contracts (GWh/Yr)</p>

### Function CapacityAndEnergyCosts

The purpose of the CapacityAndEnergyCosts Function is to calculate the cost of capacity and energy for the contracts. The capacity costs for the contract (UCCost) for each plant type are the average fixed costs for each generating company (AFC). The cost of purchasing energy for each contract (UECost) is the average variable costs for each generating company (AVC). The costs are sent to the Retail Purchases Submodule to determine the lowest cost options for dispatching contracts.

**Key Equations:**  
*CapacityAndEnergyCosts*      Contract capacity cost:  

$$\text{UCCost} = \text{AFC} * \text{AFM}$$

Contract energy cost:  

$$\text{UECost} = \text{AVC}$$

Table 16 summarizes the variable names and definitions of the input and outputs from *CapacityAndEnergyCosts*.

**Table 16: Input and Output Variables - CapacityAndEnergyCosts**

<b><i>CapacityAndEnergyCosts</i> Inputs and Outputs</b>
<p><b>Key inputs</b>            AFC [Plant,GenCo,Year] = Average Fixed Costs (\$/KW)            AVC [Plant,GenCo,Year] = Average Variable Costs (\$/MWh)</p>
<p><b>Key outputs</b></p>

<b>CapacityAndEnergyCosts Inputs and Outputs</b>
UCCost [Area,GenCo,Plant,Year] = Contract Capacity Cost (\$/KW)
UECost [Area,GenCo,Plant,Year] = Contract Energy Cost (\$/MWh)

### Function AddExogenousContracts

Depending on the type of analysis being performed, often the user will want specific predetermined contracts used for simulation instead of the endogenously created values. The code in this Function allows the user to directly set up a contract between generating and retail companies and read in the capacity, energy, and cost parameters. Exogenous contracts are typically read into the model via text files in *2020Model*, where the user specifies the parameters of the contract, including the contract capacity and costs. *AddExogenousContracts* allows for the user to use either exogenously specified contract costs ( $XCapSw = 1$ ) or endogenous costs ( $XCapSw = 2$ ).

Table 17 summarizes the variable names and definitions of the key inputs and outputs to this Function.

**Table 17: Input and Output Variables - AddExogenousContracts**

<b>AddExogenousContracts Inputs and Outputs</b>
<p><b>Key inputs</b></p> <p>XCapSw [Area,GenCo,Plant,Year]= Switch for Exogenous Contracts</p> <p>XCapacity [Area,GenCo,Plant,TimeP,Month,Year] = Exogenous Capacity under Contract (MW)</p> <p>XUCCost [Area,GenCo,Plant,Year]= Exogenous Contract Capacity Cost (\$/KW)</p> <p>XUECost [Area,GenCo,Plant,Year]= Exogenous Contract Energy Cost (\$/MWh)</p>
<p><b>Key outputs</b></p> <p>UCCost[Area,GenCo,Plant,Year] = Contract Capacity Cost (\$/KW)</p> <p>UECost[Area,GenCo,Plant,Year] = Contract Energy Cost (\$/MWh)</p> <p>Energy[Area,GenCo,Plant,Year] = Energy Limit on Contracts (GWh/Yr)</p> <p>Capacity[Area,GenCo,Plant,TimeP,Month,Year] = Capacity under Contract (MW)</p>

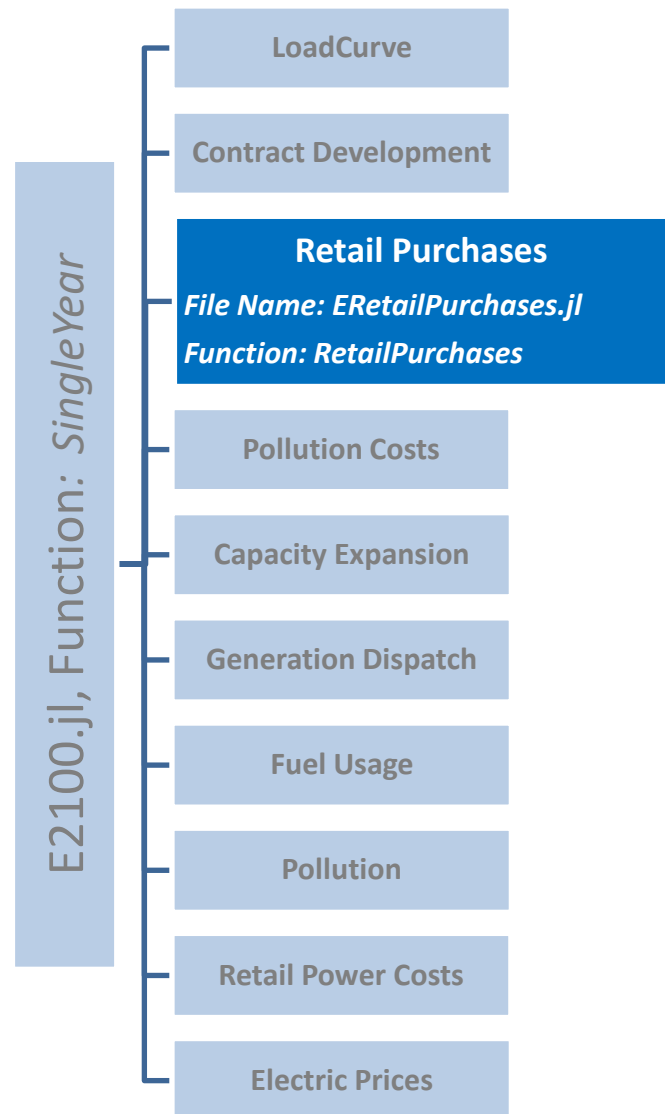
## 4. Retail Purchases Submodule

The *ERetailPurchases.jl* file contains the source code and Functions that make up the retail purchases submodule. Its main Function is named *Retail Purchases* and is called from the *SingleYear* Function inside *E2100.jl* which calls *RetailPurchases*. At the point when the retail purchases submodule is called, the load curves have been created and purchase power contracts have been developed but not dispatched.

### Submodule Objective

With the contract costs (UECost, UCCost), capacity (Capacity), and energy limits (Energy) set between each generating and retail company the model is now ready to create contracts to meet electric demands. In this section, retail companies have the option to purchase electricity either from contracts with generating companies or through the spot market dependent on the regulatory system in place for each market. The retail and generating companies create contracts until either all demands have been met or all energy available for contracts has been exhausted. If the latter occurs, then the retail company meets the remaining demand by purchasing from the spot market. The key output for this Function are the amount of generation coming from either contracts (EGBI) or from the spot market (PPEGA).

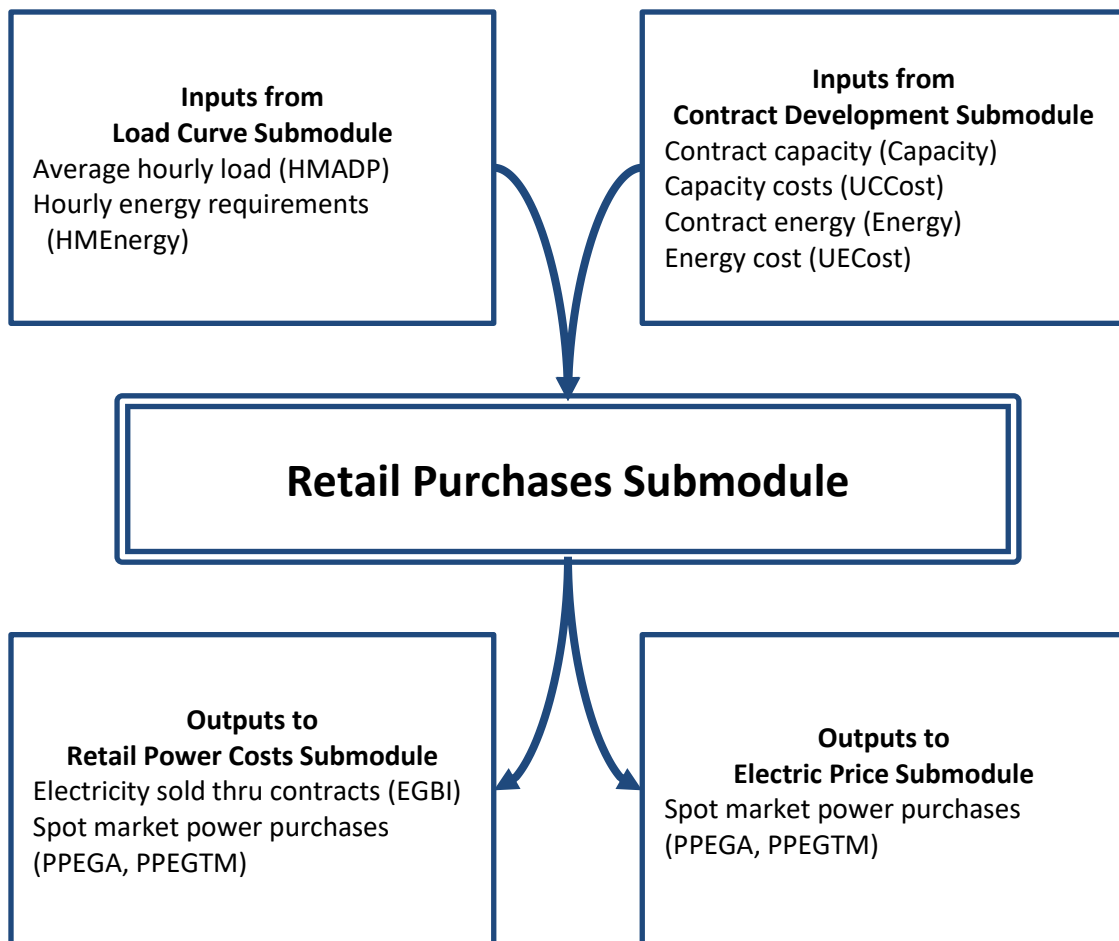
Figure 13: Retail Purchases in Electric Supply



## Submodule Methodology

Contracts for each retail company are dispatched starting with the lowest cost contracts (which are developed in the contract development submodule) until all energy needs are met or there are no more available contracts in which case energy needs are met through electricity purchases from the spot market. The amount of electricity purchased from contracts (EGBI) and the spot market (PPEGA) are output for use in calculating retail power costs and the final electric price. Figure 14 provides a diagram of the key inputs to and outputs from the retail purchases submodule.

**Figure 14. Retail Purchases Submodule Diagram of Inputs and Outputs**



## Submodule Functions

Figure 15 below lists each of the Functions that make up *RetailPurchases* and shows the order of Functions called.

**Figure 15: Function RetailPurchases in ERetailPurchases.jl**

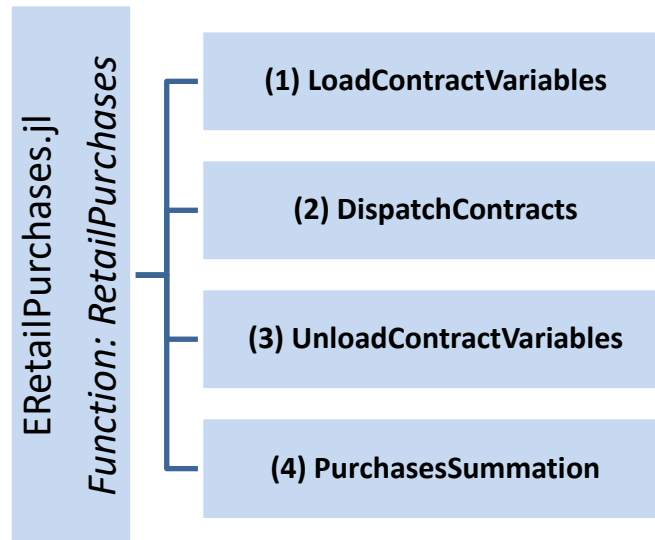


Table 18 describes each Function within the retail purchases submodule along with its key outputs. The sections that follow provide more detailed descriptions of the methodology within the Function.

**Table 18: Description of Retail Purchases Functions**

Function Description	Key Outputs
<b>1. LoadContractVariables</b> Set the capacity, amount of energy, and cost of each contract.	Contract costs and energy usage (CnEnergy, CnUECost)
<b>2. DispatchContracts</b> Dispatch contract generation starting with the lowest cost contracts. If power needed exceeds power available through contracts, then purchase from the spot market.	Contract generation (CnEG)

Function Description	Key Outputs
<p><b>3. UnloadContractVariables</b> Save total contract values for contracts and spot markets.</p>	<p>Electric sold through contracts(EGBITM) and spot market purchases (PPEGTM) by month and time period</p>
<p><b>4. PurchasesSummation</b> This Function aggregates output from Functions above into variables used by other areas of the model. The inputs are electricity sold through contracts (EGBITM) and spot market purchases (PPEGTM) each by month and time period. The saves new variables that are totaled across month and time period for electricity sold through contracts and spot market purchases (EGBI and PPEGA). Electricity sold through contracts is further summed across generating company to get total electricity sold through contracts for each area (EGA).</p>	<p>Electricity sold through contracts (EGBI); Electricity dispatched (EGA); spot market purchases (PPEGA)</p>

### Function LoadContractVariables

This section of code is designed to cycle through each combination of Generating and Retail Company for each plant type, month, and time interval to determine if capacity for a contract is above zero (Capacity). If so, a contract between retail and generator is created (CreateContract). Contract costs, energy limits, capacity, and other characteristics are saved for use in dispatching contracts. After a contract is created the model moves to the next contract slot (CnNumber) until every generating company has been checked. After all contracts have been created then parameters for the spot market for each retail company are saved. Capacity for the spot market is set equal to the peak load for each retail company (HMPDP) and the energy limit equal to the total energy available (HMEnergy). Other parameters are set equal to '99' to identify it as the spot market.

Table 19 summarizes the variable names and definitions of the key inputs and outputs to *LoadContractVariables*.

**Table 19: Input and Output Variables - LoadContractVariables**

<b>LoadContractVariables Inputs and Outputs</b>
<p><b>Key inputs</b></p> <p>Capacity [Area,GenCo,Plant,TimeP,Month,Year] = Capacity under Contract (MW)</p> <p>Energy [Area,GenCo,Plant,Year]= Energy Limit on Contracts (GWh/YR)</p> <p>UECost [Area,GenCo,Plant,Year]= Contract Energy Cost (\$/MWh)</p> <p>HMADP[Node,Area,TimeP,Month,Year] = Average Load in Interval (MW)</p>
<p><b>Key outputs</b></p> <p>CnCap [Contracts,Area,TimeP,Month] = Contract Capacity (MW)</p> <p>CnEnergy[Contracts,Area,TimeP,Month] = Contract Energy (GWh)</p> <p>CnUECost[Contracts,Area,TimeP,Month] = Contract Energy Cost (\$/MWH)</p> <p>Capacity[Area,GenCo,Plant,TimeP,Month,Year] = Capacity under Contract (MW)</p>

**Function DispatchContracts**

With the contracts and spot market for each retail company created, the model is ready to dispatch to meet electric demand. Demand needed to be met by each retail company (HMPDPM) is set equal to the average load (HMADP). Starting with the least expensive contract, demand met by each contract (CnMDS) is calculated as either the maximum capacity, energy limit of the contract, or remaining demand requirements (HMPDPM). Demand met by the contract is then subtracted from demand requirements before moving to the next contract. Total generation for each contract (CnEG) is the demand met by each contract (CnMDS) multiplied by the number of hours in each time interval (HDHours).

The equations used to create the contract generation (CnEG) are listed below.

**Key Equations:**

*Dispatch Contracts*

Contract generation:

1)  $CnEG = CnMDS * HDHours / 1000$

Where, CnMDS = Minimum (CnCap, HMADP, CnEnergy/HDHours\*1000)

Table 20 summarizes the variable names and definitions of the key inputs and outputs to the *DispatchContracts* Function.

**Table 20: Input and Output Variables - DispatchContracts**

<b>DispatchContracts Inputs and Outputs</b>
<p><b>Key inputs</b></p> <p>CnCap [Contracts,Area,TimeP,Month] = Contract Capacity (MW)</p> <p>CnEnergy [Contracts,Area,TimeP,Month] = Contract Energy (GWh)</p> <p>CnUECost [Contracts,Area,TimeP,Month] = Contract Energy Cost (\$/MWH)</p> <p>HMADP [Node,Area,TimeP,Month,Year] =Average Load in Interval (MW)</p>

<b>DispatchContracts Inputs and Outputs</b>
<p><b>Key outputs</b>  CnEG[Contracts,Area,TimeP,Month]= Contract Generation (GWh)</p>

**Function UnloadContractVariables**

This Function checks to see if generation (CnEG) is coming from a contract or the spot market. If generation is from a contract the amount of generation (EGBI) is saved to determine the cost of electricity in other parts of the model. If generation is from the spot market the amount of generation (PPEGA) is saved.

**Key Equations:**  
*Unload ContractVariables*

Electricity sold through contracts:  
EGBITM=CnEG

Power purchases:  
PPEGTM=CnEG

Table 21 summarizes the variable names and definitions of the key inputs and outputs to the *UnloadContractVariables* Function.

**Table 21: Input and Output Variables - UnloadContractVariables**

<b>UnloadContractVariables Inputs and Outputs</b>
<p><b>Key inputs</b>  CnEG[Contracts,Area,TimeP,Month] = Contract Generation (GWh)  CnGenCo [Contracts,Area,TimeP,Month] = Contract GenCo</p> <p><b>Key outputs</b>  EGBITM[Area,GenCo,Plant,TimeP,Month,Year] = Electricity sold thru Contracts (GWh/YR)  PPEGTM[Area,TimeP,Month,Year] = Power Purchases (GWh)</p>

**Function PurchasesSummation**

This Function aggregates output from Functions above into variables used by other areas of the model. The inputs are electricity sold through contracts (EGBITM) and spot market purchases (PPEGTM) each by month and time period. The saves new variables that are totaled across month and time period for electricity sold through contracts and spot market purchases (EGBI and PPEGA). Electricity sold through contracts is further summed across generating company to get total electricity sold through contracts for each retail company (EGA).

The key equations used in the PurchasesSummation Function are as follows:

**Key Equations:**  
*Purchases  
 Summation*

Electricity sold through contracts by plant type, retail company, generating company:

$$EGBI = \sum_{\text{TimeP, Month}} EGBITM$$

Electricity sold through contracts by plant and retail company:

$$EGA = \sum_{\text{GenCo}} EGBI$$

Power purchases through spot market:

$$PPEGA = \sum_{\text{TimeP, Month}} PPEGTM$$

$$TPPEGA = \sum_{\text{Power types}} PPEGA$$

Table 22 summarizes the variable names and definitions of the key inputs to and outputs from *PurchasesSummation*.

**Table 22: Input and Output Variables - PurchasesSummation**

<b><i>PurchasesSummation</i> Inputs and Outputs</b>
<p><b>Key inputs</b>                      EGBITM[Area,GenCo,Plant,TimeP,Month,Year] = Electricity sold thru Contracts (GWh/YR)                      PPEGTM[Area,TimeP,Month,Year] = Power Purchases (GWh)</p> <p><b>Key outputs</b>                      EGBI[Area,GenCo,Plant,Year]= Electricity sold thru Contracts (GWh/YR)                      EGA[Plant,Area,Year] = Electricity Dispatched (GWh/Yr)                      PPEGA[PPSet,Area,Year] = Spot Market Purchases (GWh)                      TPPEGA[Area,Year] = Total Purchase Power (GWh)</p>

## 5. Pollution Costs Submodule

*EPollution.jl* contains the source code and Functions that make up the pollution submodule. The pollution cost submodule's main Function is named *Part1* and is called from the *SingleYear* Function inside *E2100.jl*. It is called after load curve, contract development, and retail purchases submodules have been executed.

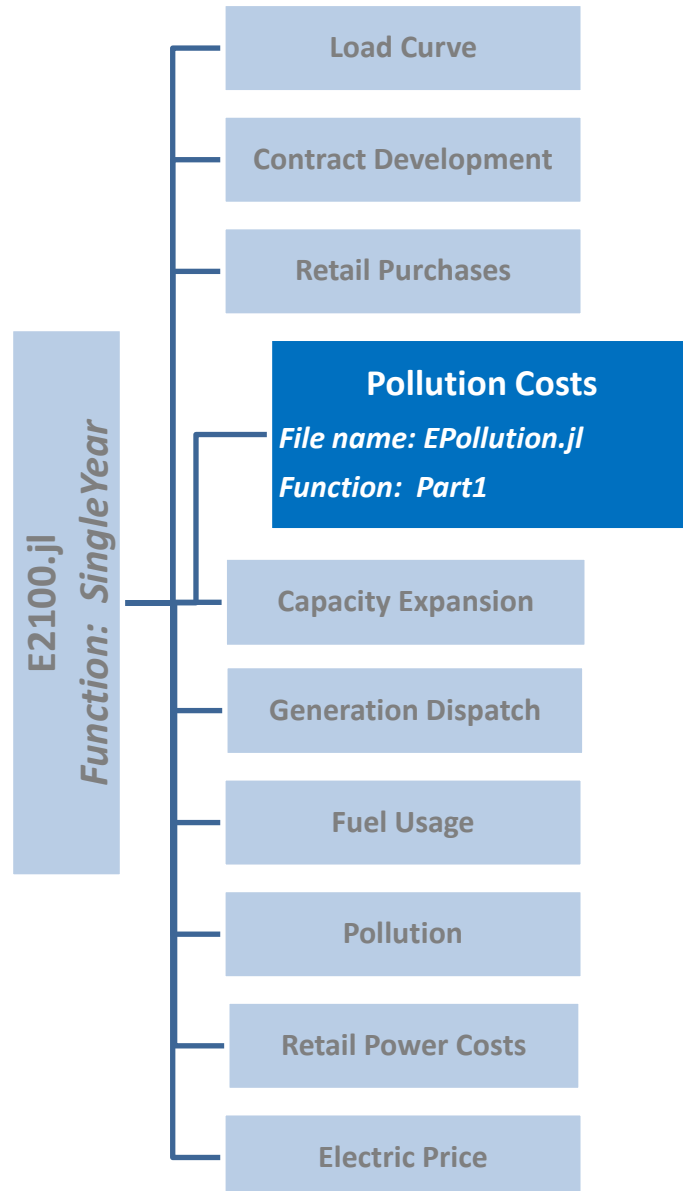
### Submodule Objective

The pollution costs submodule determines the amount of emissions reduction required for the utility generation sector from emission reduction policies and applies reductions at the individual unit level. Marginal and average emissions coefficients are recalculated along with an estimation of the costs of the reduction based on abatement curves. The submodule is designed to be robust and offers the user different options to determine how reductions for each unit are calculated.

### Submodule Methodology

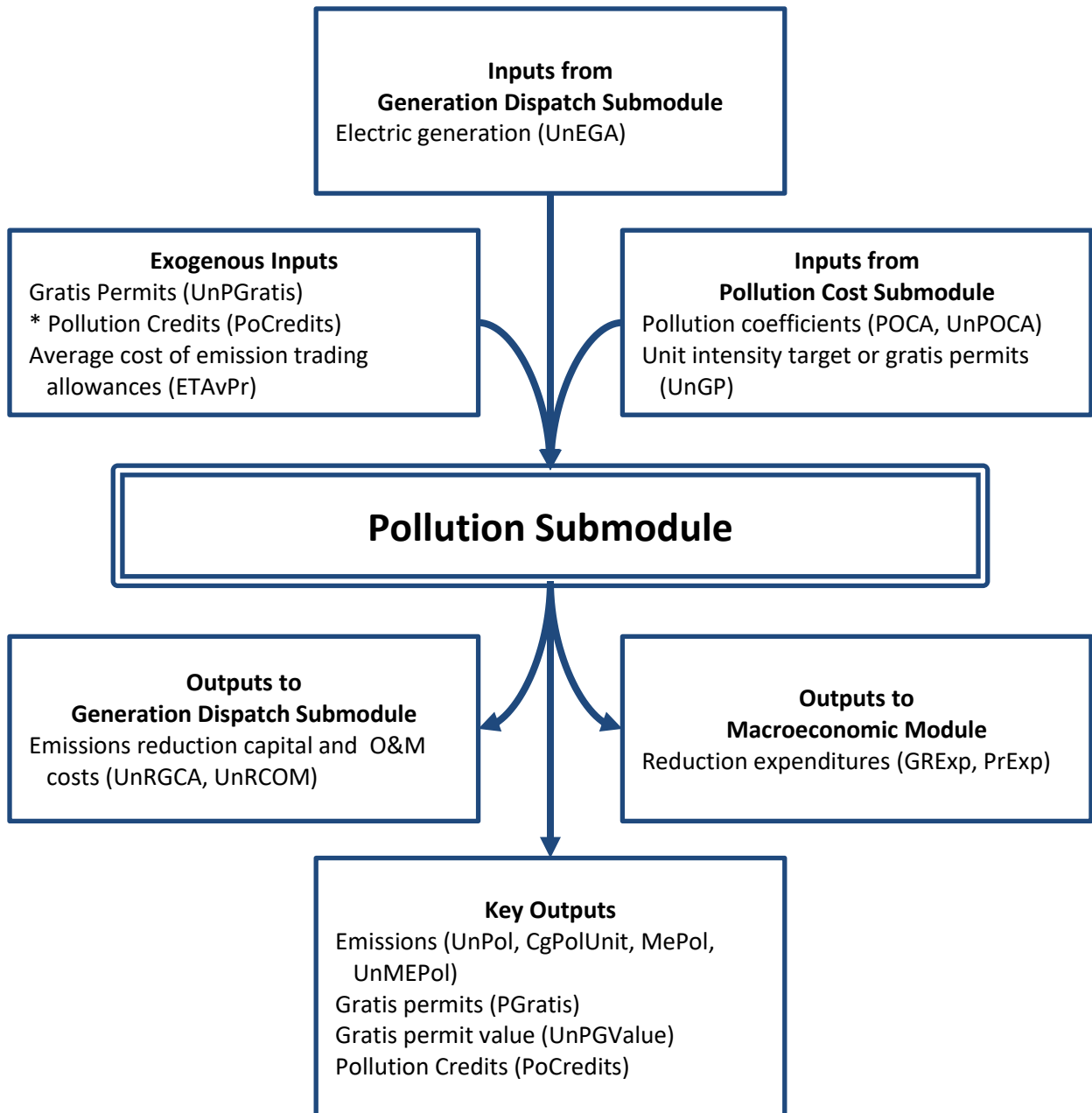
Plant-type and unit-level emissions reductions and reduction capital costs are calculated. These are then used to adjust pollution coefficients and together they are used to determine the impact of emissions reductions on the price of electricity. Key

Figure 16: Pollution Costs in Electric Supply



inputs to *Part1* include the reduction required (RPolicy) or pollution costs (PCostECC), cost curve parameters (EUPCostN, EUPVF), and policy specified model switches to determine the Submodule Methodology (UnPRSw, CapTrade). Key outputs include emission coefficients (POCX, UnPOCX), and reduction costs (UnPoTR). Figure 17 illustrates the key inputs and outputs of the pollution submodule.

**Figure 17. Pollution Submodule Diagram of Inputs and Outputs**



## Submodule Functions

The main Functions called from the *Part1* Function in EPollution.jl are shown in Figure 18 below.

**Figure 18: Function Part1 in EPollution.jl**

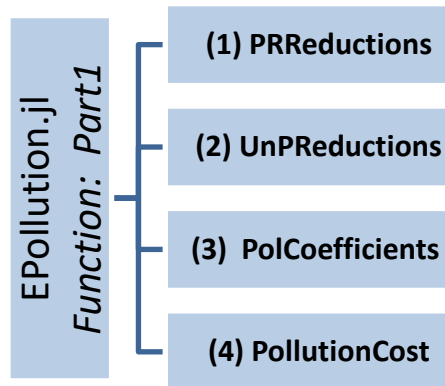


Table 23 describes each of the pollution costs Functions and provides the output variables from each Function. The subsequent sections provide more detailed descriptions of the methodology within each of the pollution Functions.

**Table 23: Description of Pollution Costs Functions**

Function Description	Key Outputs
<b>1. PRReductions</b> Calculates the reductions required and marginal reduction costs at the sector level based on emissions caps or other emissions policies.	Reduction requirements (EURM, EURP) and marginal cost (EURCC)
<b>2. UnPReductions</b> Unit-level emissions reduction requirements and marginal cost.	Reduction requirements (UnRM, UnRP) and marginal cost (UnRCC)
<b>3. PolCoefficients</b> Calculates the average emissions coefficient	Average emitted per unit of energy (POCA, UnPOCA)
<b>4. PollutionCosts</b> Creates marginal and average cost of emitting per megawatt hour based on average emissions coefficient	Marginal and average pollution tax rates (UnPoTxR, UnPoTA <sub>v</sub> )

---

## Function *PRReductions*

The *PRReductions* Function determines the emissions reductions required for the utility sector; there are two main ways emissions reductions are calculated. The method of reduction for the utility sector is dependent on the value of a policy switch (*CapTrade*) allowing the user to specify the method of reduction to be simulated. Either the cost of purchasing emissions permits or an emissions cap to meet an emission reduction policy will be specified by the policy. If the cost is specified, the cost of emissions (*EURCC*) reduction will be used along with abatement curve parameters (*EUPCostN*, *EUPVF*) to calculate a percentage reduction (*EURP*). If the reduction is specified, the intended percentage reductions (*EURP*) will be used with abatement curve parameters to calculate a cost of reduction (*EURCC*). The pollution reduction curve is simply the relationship between the cost of reductions and the amount of reductions.

Policies from different years may use different parameters and slightly different formulas, based on a switch (*EURPCSw*). Most CAC policies are able to use this generic emissions policy logic. Some cap-and-trade policies may work in a different manner and for such policies, the reductions and reduction capital costs from this Function may be treated as zero.

Once a reduction and a cost have been determined, these are used to create an emissions reduction multiplier. *EURM* is the multiplier to the total in order to create the desired percentage reduction (*EURP*), but also incorporating any voluntary reductions (*EUVR*) and exogenously specified reductions (*XEURM*).

### ***PRReductions Inputs and Outputs***

#### **Key Inputs**

CapTrade [Market,Year] = Emission Cap and Trading Switch

EUPCostN [FuelEP,Plant,Poll,Area,Year] = Pollution Reduction Cost Normal (\$/Tonnes)

EUPVF [FuelEP,Plant,Poll,Area,Year] = Pollution Reduction Variance Factor  
 ((\$/Tonnes)/(\$/Tonnes))

EURCD [Poll] = Reduction Capital Construction Delay (Years)

EURCPL [Poll] = Reduction Capital Physical Life (Years)

PCost [FuelEP,Plant,Poll,Area,Year] = Permit Cost (\$/Tonnes)

RPolicy [ECC,Poll,Area,Year] = Pollution Reduction from Limit (Tonnes/Tonnes)

#### **Key outputs**

EURCC [FuelEP,Plant,Poll,Area,Year] = Reduction Capital Cost (\$/Tonnes)

EURM [FuelEP,Plant,Poll,Area,Year] = Reduction Multiplier by Area (Tonnes/Tonnes)

EURP [FuelEP,Plant,Poll,Area,Year] = Pollutant Reduction (Tonnes/Tonnes)

Key equations for the plant-type reductions in Function *PRReductions* include:

**Key Equations:**  
*PRReductions*

*Reduction capital cost and pollutant reduction factor for policies where costs are specified first:*

$EURCC = (PCost * ECoverage + PCostExo) / (EUCCR + EUROCF) * Infla$

$EURP = EURP + (Maximum(EURP, EUIRP) - EURP) / EURCD - EURP / EURCPL$

Actual Reductions (EURP) are increased by changes in Indicated Reductions (EUIRP) and the construction time (EURCD) and reduced by the physical lifetime (EURCPL). Indicated reductions are calculated from costs and curve parameters:

- $EUIRP = 1 / (1 + ((x_{max}((PCost * ECoverage + PCostExo) * PCostM, 0.01) * EURCstM) / EUPCostN) ** EUPVF)$  [2004 Curve Parameters]
- $EUIRP = \ln(EURCC / Infla / (EUPCostN / PCostM / EURCstM)) / EUPVF * ECoverage$  [2009 Curve Parameters]

*Reduction capital cost and pollutant reduction factor for policies where reductions are specified first:*

$EURP = RPolicy * ECoverage$

$EURCC = (1 / RPF_{full} - 1) ** (1 / EUPVF) * EUPCostN / EURCstM / (EUCCR + EUROCF) * Infla$   
 [2004 Curve Parameters]

$EURCC = (EUPCostN / EURCstM) * \exp(EUPVF * RPF_{full}) * Infla$  [2009 Curve Parameters]

Reduction capital cost and pollutant reduction factor for other policies:

EURP = 0

EURCC = 0

Reduction multiplier by area:

EURM = (1-EURP)\*(1-EUVR)\*XEURM

**Function UnPReductions**

With the sector level reduction requirement established, *UnPReductions* determines the reduction amount and emission costs for individual units using methods determined by the value of UnPRSw. The first option (in subFunction *UnCurve*) is for individual units to receive the same reductions levels (UnRP) and costs (UnRCC) as the plant-type based results from the previous Function (EURP, EURCC). Second, the subFunction *UnReduce* compares plant-type reductions to an exogenous value. Then, costs are based on the higher of the two reduction levels and the abatement curve parameters (EUPCostN, EUPVF).

Key Equations:  
*UnPReductions*

<b><i>UnPReductions Inputs and Outputs</i></b>	
<b>Key Inputs</b>	
EUCCR [Area,Year]	= Pollution Reduction Capital Charge Rate (\$/\$)
EURCC [FuelEP,Plant,Poll,Area,Year]	= Reduction Capital Cost (\$/Tonnes)
EURP [FuelEP,Plant,Poll,Area,Year]	= Pollutant Reduction (Tonnes/Tonnes)
RPOCX [PRTech,FuelEP,Poll,Plant,Year]	= Pollution Reduction Multiplier (Tonnes/TBtu/Tonnes/TBtu)
XUnRP [Unit,FuelEP,Poll,Year]	= Pollution Reduction (Tonnes/Tonnes)
XUnRCC [Unit,FuelEP,Poll,Year]	= Pollution Reduction Capital Cost (\$/Tonnes)
<b>Key outputs</b>	
UnRCC [Unit,FuelEP,Poll,Year]	= Pollution Reduction Capital Cost (\$/Tonnes)
UnRM [Unit,FuelEP,Poll,Year]	= Pollution Reduction Multiplier (Tonnes/Tonnes)
UnRP [Unit,FuelEP,Poll,Year]	= Pollution Reduction (Tonnes/Tonnes)

Key equations for unit-level costs and reductions in *UnPReductions* include:

Pollution reduction factor and capital cost for plant-type based cost:

- a. UnRP=EURP
- b. UnRCC=EURCC

Pollution reduction factor and capital cost for semi-exogenous reductions:

- a. UnRP=Maximum(EURP, XUnRP)

- b.  $UnRCC = (1/UnRP - 1) \cdot (1/EUPVF) \cdot EUPCostN / EURCstM / (EUCCR + UnROCF) \cdot Infla$  [2004 Curve Parameters]
- c.  $UnRCC = (EUPCostN / EURCstM) \cdot Exp(EUPVF \cdot UnRP) \cdot Infla$  [2009 Curve Parameters]

**Pollution reduction multiplier:**

$$UnRM = (1 - UnRP) \cdot (1 - UnRP2) \cdot (1 - EUVR)$$

Where: UnRP2 is a variable slot for a second source of reductions, but is currently unused.

### Function PolCoefficients

The *PolCoefficients* Function calculates both the sector-wide and unit-level average emissions coefficients based on the input default marginal emissions coefficient adjusted by the required reduction amount. This allows for the emissions inventory to match the new reduced value. The average coefficient (POCA, UnPOCA) are calculated from the product of the exogenous coefficients (POCX, UnPOCX) and the Emissions Reduction Multipliers (EURM, UnRM). Coefficients for black carbon are calculated from the coefficients for particulate matter smaller than 2.5 micrometers. Additionally, a coefficient for approximate emissions per GWh (UnPOCGWh) is calculated for use in generation dispatch for pollution-related policies.

#### ***PolCoefficients Inputs and Outputs***

##### **Key Inputs**

EURM [FuelEP,Plant,Poll,Area,Year] = Reduction Multiplier by Area (Tonnes/Tonnes)

POCX [FuelEP,Poll,Plant,Area,Year] = Marginal Pollution Coefficient (Tonnes/TBtu)

UnDmd [Unit,FuelEP,Year] = Energy Demand (TBtu)

UnEGA [Unit,Year] = Generation (GWh)

UnPOCX [Unit,FuelEP,Poll,,Year] = Pollution Coefficient (Tonnes/TBtu)

UnRM [Unit,FuelEP,Poll,Year] = Pollution Reduction Multiplier (Tonnes/Tonnes)

##### **Key outputs**

POCA [FuelEP,Plant,Poll,Area,Year] = Average Pollution Coefficients (Tonnes/TBTU)

UnPOCA [Unit,FuelEP,Poll,Year] = Average Pollution Coefficient (Tonnes/TBtu)

UnPOCGWh [Unit,Poll,Year] = Pollution Coefficient (Tonnes/GWh)

Key Equations in *PolCoefficients* include:

**Key Equations:** *Average pollution coefficient by plant type, fuel type, and area:*  
*PolCoefficients* POCA=POCX\*EURM

*Average pollution coefficient by unit and fuel:*  
 UnPOCA=UnPOCX\*UnRM

*Pollution coefficient in tonnes/GWh by unit:*  
 UnPOCGWh =  $[\sum(\text{UnDmd} * \text{UnPOCA})] / \text{UnEGA}$

### Function PollutionCost

*PollutionCost* uses emission coefficients and trading allowances to compute the marginal and average pollution costs (tax rates, fees, or other per MWh costs directly arising out of emissions policies) and transmission pollution costs. The essential construction of the pollution tax rates is the product of the price of emissions trading allowances (ETAPr), the fuel-based pollution coefficient (UnPOCA, POCXNew), the heat rate (UnHRt, HRTM), and fractions of fuel used for each plant type (UnFIFr, FIFrNew), with various adjustments for unit conversion.

Unit Gratis Permits are also calculated for use in later pollution-related Functions. Because many GHG policy schemes link gratis permits to emissions targets, both terms are sometimes used.

#### ***PollutionCost Inputs and Outputs***

##### **Key Inputs**

ETAPr [Market,Year] = Cost of Emission Trading Allowances (\$/Tonnes)

UnPOCA [Unit,FuelEP,Poll,Year] = Average Pollution Coefficient (Tonnes/TBtu)

UnEGA [Unit,Year] = Generation (GWh)

##### **Key outputs**

LLPoTxR [Node,NodeX,Market,Year] = Pollution Costs for Transmission (\$/MWh)

PoTxR [Plant,GenCo,Year] = Pollution Tax Rate (\$/MWh)

PoTRNew [Plant,Area,Year] = Emission Cost for New Plants (\$/MWh)

UnGP [Unit,FuelEP,Poll,Year] = Unit Intensity Target or Gratis Permits (kg/MWh)

UnPoTR [Unit,Year] = Marginal Pollution Tax Rate (\$/MWh)

---

Key equations in *PollutionCost* include:

**Key Equations:**  
*PollutionCost*

*Pollution costs for transmission:*

$$LLPoTxR = LLPOCX * ETAPr / 1000$$

*Marginal pollution tax rate:*

$$UnPoTR = \sum (ETAPr / PolConv * UnPOCA * UnHRt / 1e9 * UnCoverage * UnFIFr)$$

*Emission cost for new plants:*

$$PoTRNew = \sum (ETAPr / PolConv * (POCXNew * HRTM / 1e6 - GPNew) / 1000 * FIFrNew)$$

*Pollution tax rate:*

$PoTxR = PoCstG / GenG$ , where

- a.  $PoCstG = PoCstG + UnPoTR * UnEGA$ , accumulated across units
- b.  $GenG = GenG + UnEGA$ , accumulated across units

## 6. Capacity Expansion Submodule

*ECapacityExpansion.jl* contains the source code and Functions that make up the capacity expansion submodule. It is called after the pollution cost submodule has been executed. The main Function that executes all capacity expansion routines is called *ExpandCapacity*. *ExpandCapacity* is initiated from Function *SingleYear* (within *E2100.jl*) which calls Function *Ctrl* from within *ECapacityExpansion.jl*.

### Submodule Objective

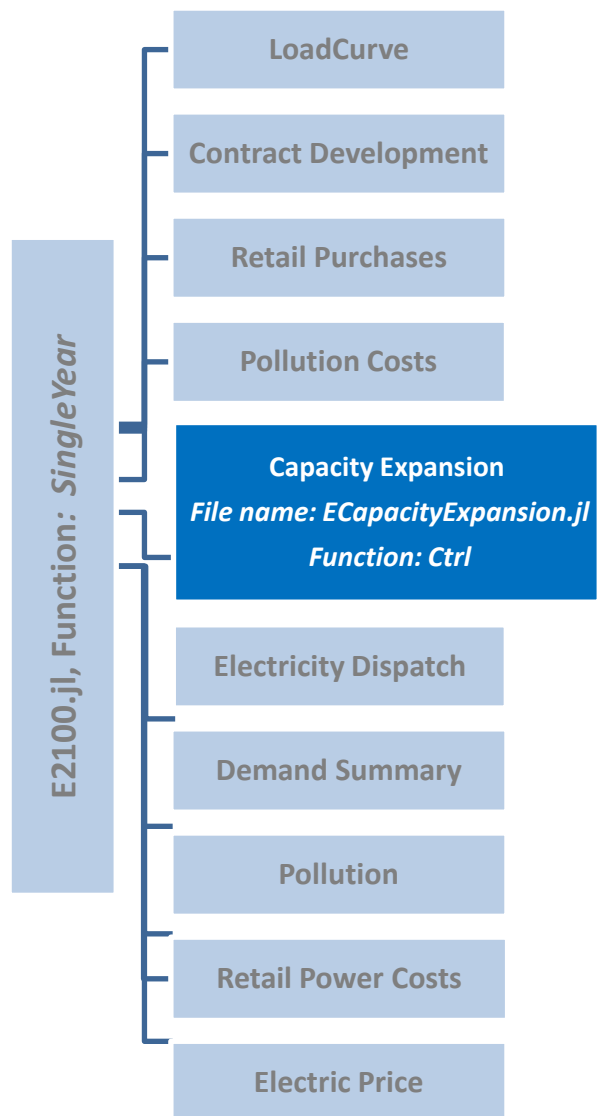
The capacity expansion submodule builds new generating units to meet increases in annual generating capacity demands.

### Submodule Methodology

The decision to build new generation capacity involves many factors. ENERGY 2100 includes a variety of algorithms to simulate the decision to build new capacity. These algorithms include information on wholesale prices, desired reserve margins, the cost of new capacity (construction, operating, and environmental costs), political and social preferences, regulations, standards, and subsidies. The model is adjusted and customized to simulate the decision-making process at a particular time and location.

The first step in capacity expansion is to determine how much capacity to build (the type of plants to build is secondary to determining the total amount of capacity required). Several optional methodologies are available to forecast capacity requirements. These methodologies include:

Figure 19: Capacity Expansion in Electric Supply



**Methods of  
building new  
capacity**

1. **Build new capacity based on reserve margin choosing lowest cost plants:** One method is to build new capacity based on meeting a desired reserve margin, constructing the lowest cost plant type which is feasible. Projections of future demand are estimated based on the current year's peak demand. The number of years which are forecasted depends on the construction time of the units being built. The forecasted peak demand is compared to the existing capacity (for intermittent sources such as wind, capacity is derated), capacity to be retired, and capacity under construction to decide the amount of new capacity.
2. **Build new capacity based on prices:** Another method is to build new capacity based on the wholesale price of power. As the wholesale price rises, new capacity will be constructed even if current capacity is higher than the desired reserve margin. Conversely, if the wholesale price is low, then new capacity will not be constructed even if capacity is lower than the desired reserve margin.
3. **Build new capacity based on prices taking into account the reserve margin:** This method combines wholesale prices and a desired reserve margin. In this case capacity is constructed if the wholesale price is high and capacity is needed. The amount constructed depends on the desired reserve margin.

The actual method used depends on the situation, such as amount of regulation, in the particular area and is a user-specified input to the model. Within a single model some generating companies (GenCo) can use one strategy while another generating company uses a different one.

Any of the methods can generate over-building or under-building although they will correct themselves over time. The model's construction of new capacity depends on simulated forecasts, and do not necessarily match future demand requirements. If too many new units are constructed by generating companies attempting to take advantage of higher prices, this will lead to excess capacity. Likewise, if the costs or barriers to construction are too high, there may be insufficient new units to meet demand requirements. This is the desired result since historically we often see regions in an over-capacity or under-capacity situation.

Once the amount of new capacity is determined, the model then decides the type of capacity. The two primary factors are which types of capacity are politically feasible and the costs of the various types of generating capacity. The relative political feasibility of constructing various plant types is an input to the model.

---

Costs of new capacity include capital costs (including financing), O&M costs, fuel costs, environmental costs, and subsidies. A levelized cost is computed for each type of power (peaking, intermediate, and baseload) and used when selecting the least cost power.

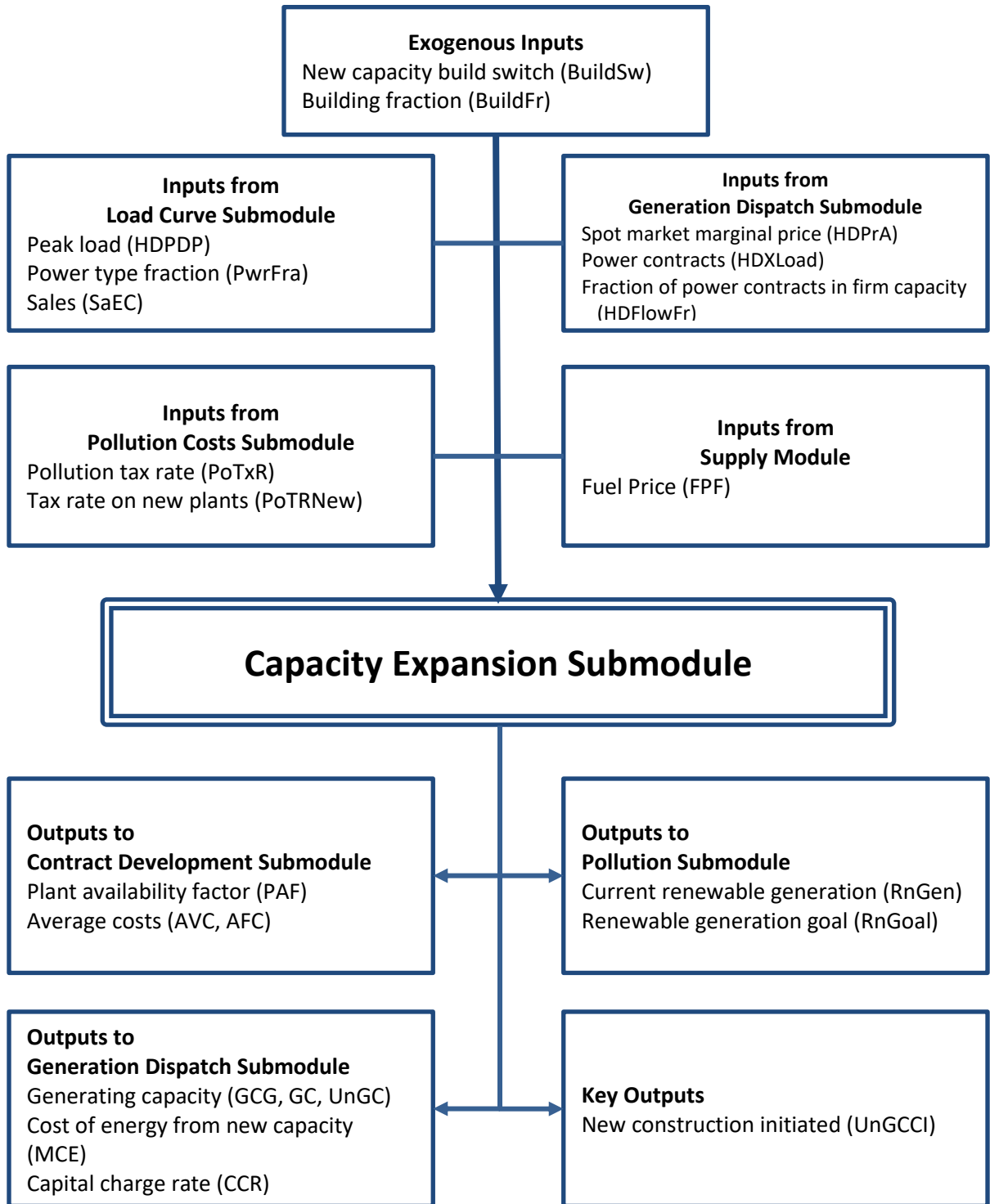
Several exceptions exist that allow for forcing certain types of plants to be built before calling the capacity expansion routines. These exceptions include: 1) building to meet a renewable portfolio standard; 2) building to an exogenously-specified plant type mix; and 3) building intermittent power (as a pre-specified fraction); and 4) building conventional power as back-up for intermittent power.

**Building  
renewable  
power**

Renewable power may be selected as part of the normal plant type selection process, but it can also be built to meet a renewable portfolio standard (RPS). An RPS is expressed as a fraction of sales which must be met with renewable power. The model selects the plant types which qualifies for the RPS and calculates the levelized cost for each plant type. The amount of new capacity is a function of the RPS requirement and the current renewable generation. The type of renewable power developed is determined using a qualitative choice function using the cost of each type of renewable power. These costs are all adjusted for the expected capacity factor of each renewable resource. The capital cost of the renewable resource will increase as the availability of the renewable resource is depleted.

The key inputs to the capacity expansion submodule are listed below with variable names indicated in parenthesis.

Figure 20. Capacity Expansion Submodule Diagram of Inputs and Outputs



The capacity expansion routines called from *ECapacityExpansion.Ctrl* are listed in the table below.

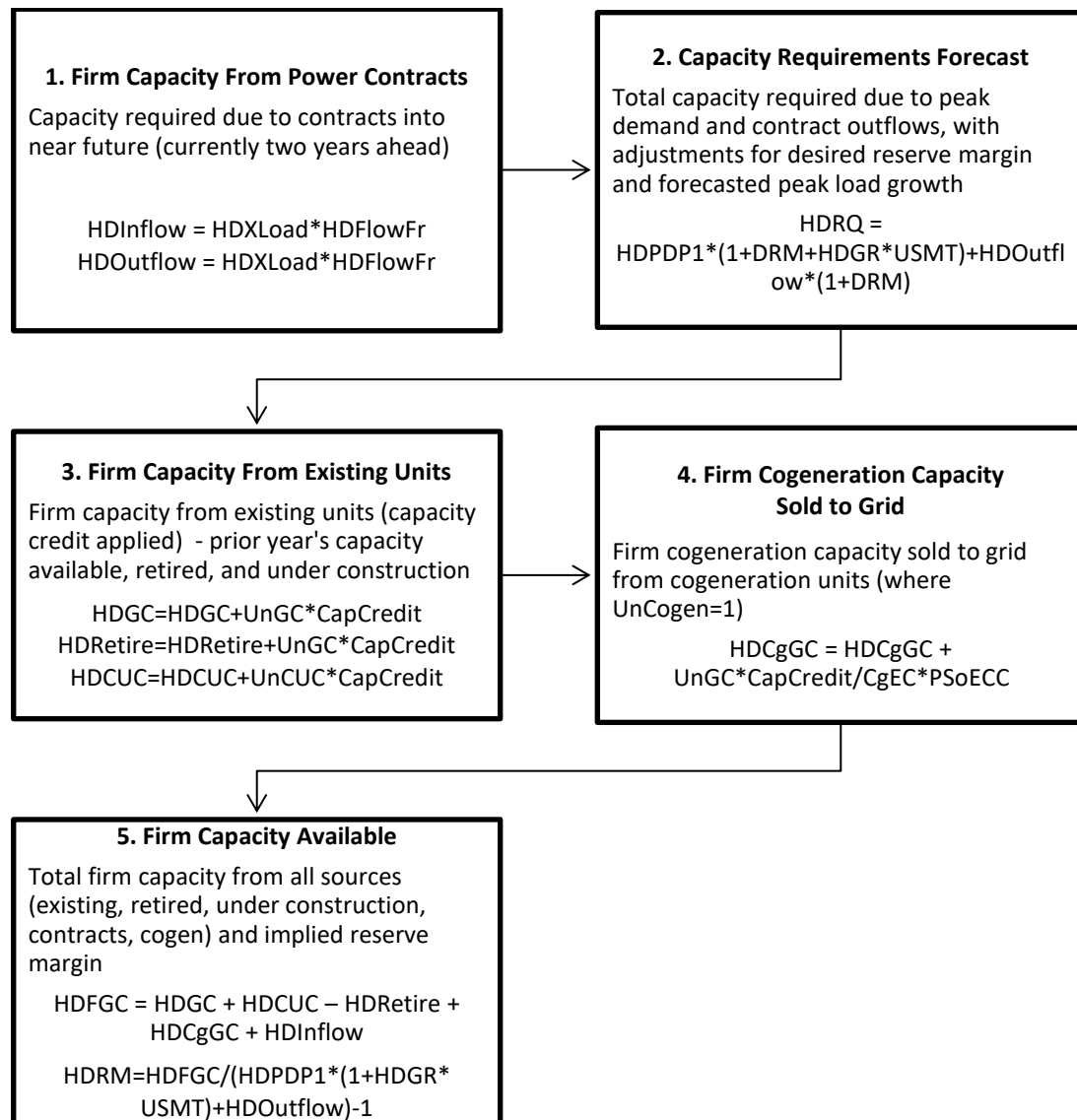
<b>Purpose</b>	<b>Functions Called from Ctrl in ECapacityExpansion.jl</b>
Capacity Requirements	<ol style="list-style-type: none"> <li>1. FirmCapacityFromPowerContracts</li> <li>2. CapacityRequirementsForecast</li> <li>3. FirmCapacityFromExistingUnits</li> <li>4. FirmCogenerationCapacitySoldToGrid</li> <li>5. FirmCapacityAvailable</li> </ol>
Capacity Costs	<ol style="list-style-type: none"> <li>1. CapacityAlreadyDeveloped</li> <li>2. CapitalCostMultiplierFromDepletion</li> <li>3. CapitalCostMultiplierFromETC</li> <li>4. CapacityCapitalCost</li> <li>5. FuelPrices</li> <li>6. ProjectCosts</li> </ol>
Capacity Initiated	<ol style="list-style-type: none"> <li>1. CapacityUnderConstructionMultiplier</li> <li>2. BuildFractionBasedOnClearingPrice</li> <li>3. CapacityNeededBasedOnClearingPrice</li> <li>4. CapacityNeededForReserveMargin</li> <li>5. BuildNewCapacity</li> <li>6. Initiation</li> <li>7. UnitInitiation</li> </ol>
Capacity Construction	<ol style="list-style-type: none"> <li>1. GetUtilityUnits</li> <li>2. UnitConstruction</li> <li>3. ConstructionTotals</li> </ol>

A description of the methodologies to calculate capacity requirements, capacity costs, capacity initiated, and capacity construction are summarized in the following sections.

### **Capacity Requirements**

The capacity expansion routines first calculate projections of capacity requirements at each node ( $HDRQ_{Node,Year}$ ), how much firm generating capacity is available ( $HDFGC_{Node,Year}$ ), and the size of the reserve margin ( $HDRM_{Node,Year}$ ). The capacity required is determined based on peak demand plus contract outflows by node, increased by a desired reserve margin and forecasted peak load growth. Figure 21 illustrates a flowchart of the routines that calculate capacity requirements.

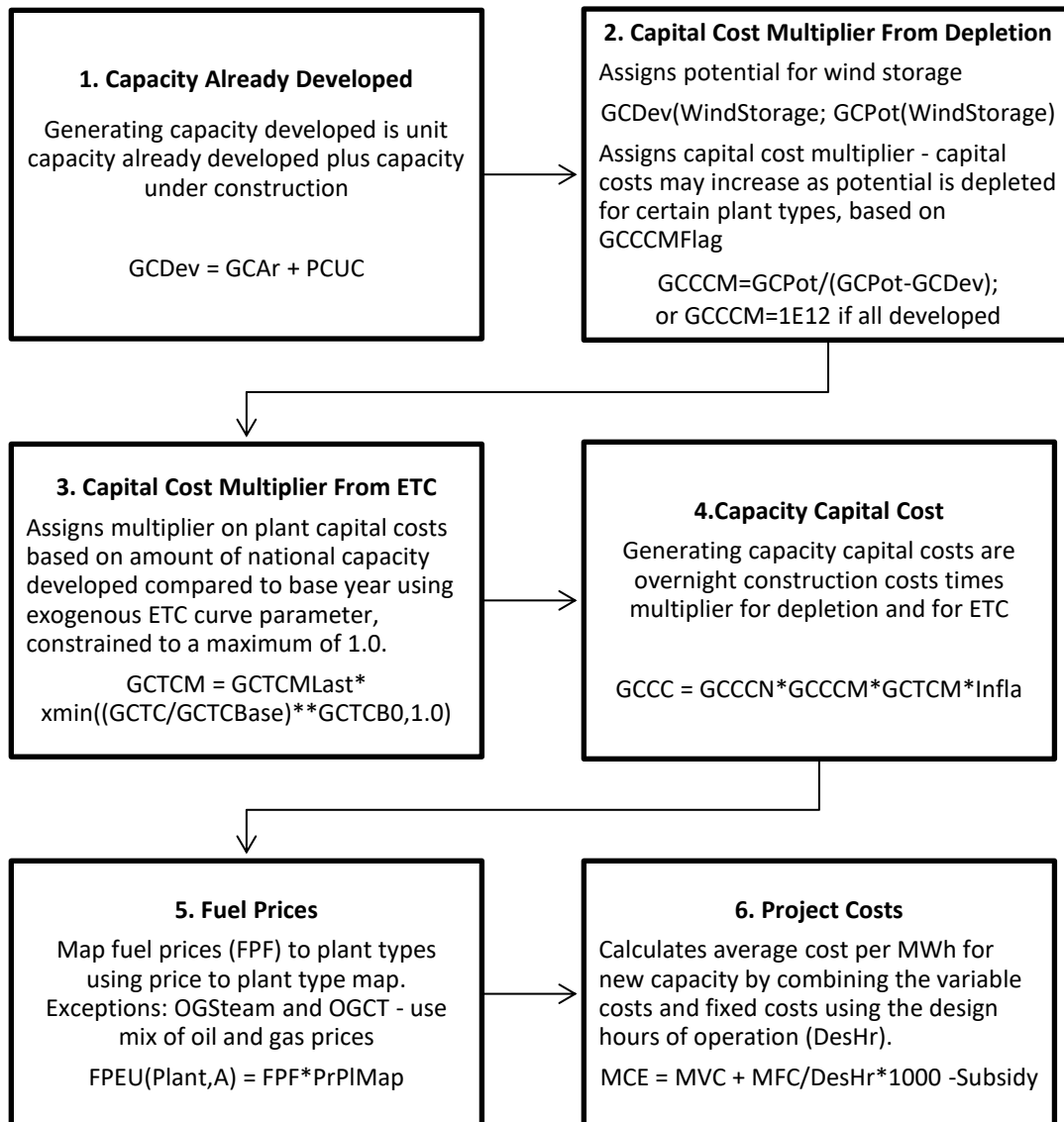
Figure 21. Capacity Expansion Routines – Capacity Requirements



## Capacity Costs

The purpose of the capacity cost routines is to calculate a marginal cost of energy (MCE) for each plant type by type of power (base, intermediate, and peak). Some plant types increase capital costs as generating potential is depleted, such as wind, solar, and biomass. Some other plant types capital costs are increased only after all potential is depleted (such as fuel cells and batteries). Costs can also be decreased due to learning by doing based on an endogenous technology change curve. Figure 22 provides a flowchart of the routines calculating the cost of new capacity.

Figure 22. Capacity Expansion Routines – Capacity Costs



### Capacity Initiated

The capacity-initiated routines ultimately determine how much, when, and where new capacity will be added to existing or new electric generating units. The main output variables are:

- $HDGCCl_{Plant, Node, GenCo, Area, Year}$  - Capacity Initiated Before Renewable Programs (MW)
- $IPGCCl_{Plant, Node, GenCo, Area, Year}$  - Intermittent Power Capacity Initiated (MW)
- $RnGCCl_{Plant, Node, GenCo, Area, Year}$  - Renewable Capacity Initiated (MW)
- $UnGCCl_{Unit, Year}$  - Unit Generating Capacity Initiated (MW)

**Build decision based on clearing price:** New capacity is built (IPGCP<sub>r</sub>) if the decision price (HDPrDP<sub>Power,Node,Year</sub>) is greater than the cost of new capacity (MCE<sub>Plant,Power,Area,Year</sub>) for a given plant type, power type (base, intermediate, or peak), and location. The decision price (HDPrDP) is the price expected for the power from the new plant which is the average clearing price (HDPrA<sub>Node,TimeP,Month,Year</sub>) for the hours in each time period. A Power fraction (PwrFra<sub>Power,GenCo,Year</sub>) determines whether peak, intermediate, or baseload capacity is constructed. The build fraction (BuildFr<sub>GenCo,Year</sub>) determines the amount of new capacity constructed by each company. The build decision is increasingly constrained (by CUCMI<sub>t</sub>) as the fraction of capacity under construction to total capacity (CUCFr) approaches a limit (CUCLimit).

$$IPGCP_r = HDRQ * NdArFr * PjNSw * BuildFr * BFraction * CUCMI_t$$

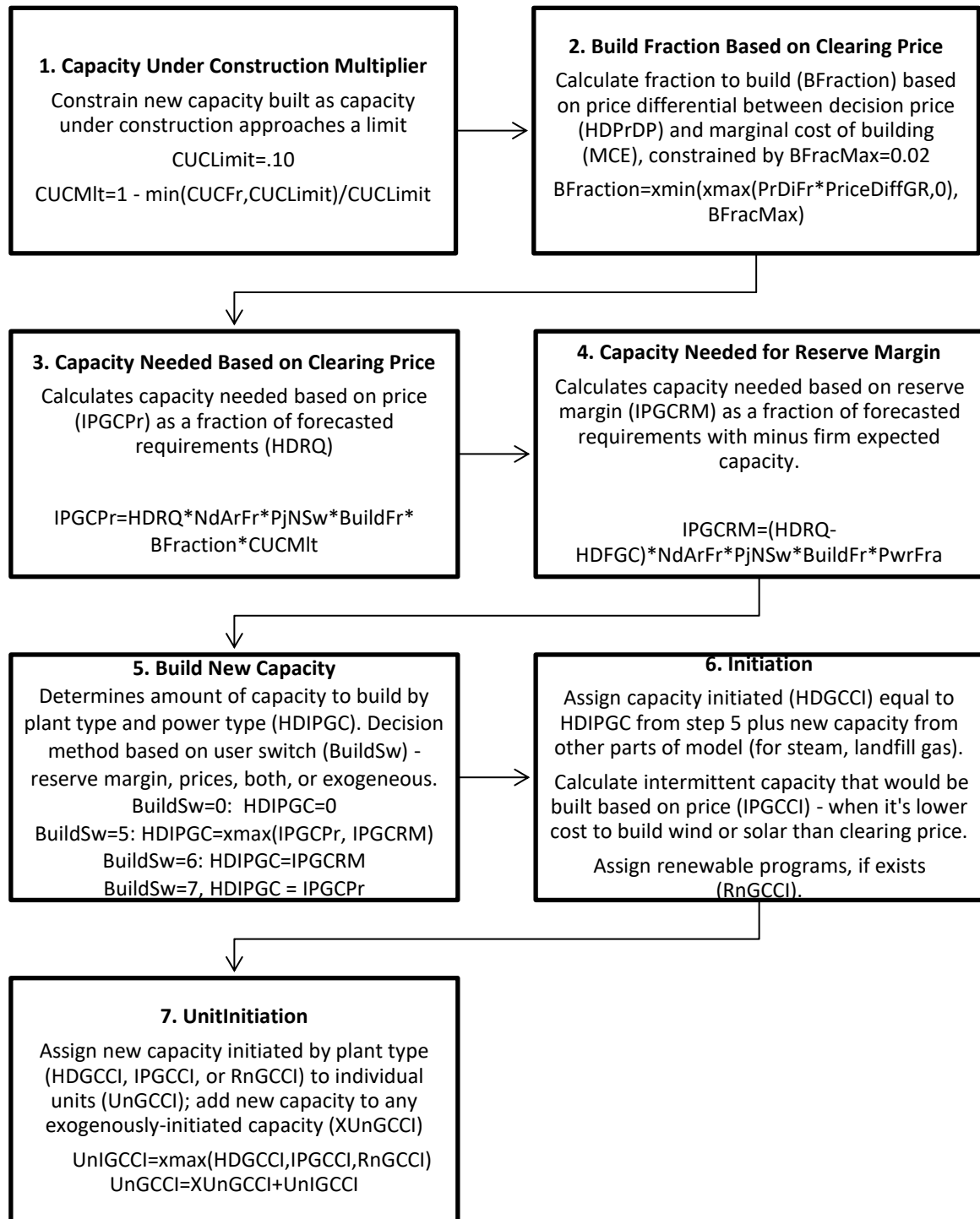
**Build decision based on reserve margin:** New capacity is built due to reserve margin (IPGCRM<sub>Power,Node,GenCo,Area,Year</sub>) if the capacity required (HDRQ) is greater than the firm capacity available (HDFGC). The Area Node Fraction (NdArFr) determines which Area the capacity is constructed. The Power fraction (PwrFra) determines whether peak, intermediate, or baseload capacity is constructed. The build fraction (BuildFr) determines the amount of new capacity constructed by each company. The Project Node Switch (PjNSw) specifies which Nodes the company builds on.

$$IPGCRM = (HDRQ - HDFGC) * NdArFr * PjNSw * BuildFr * PwrFra$$

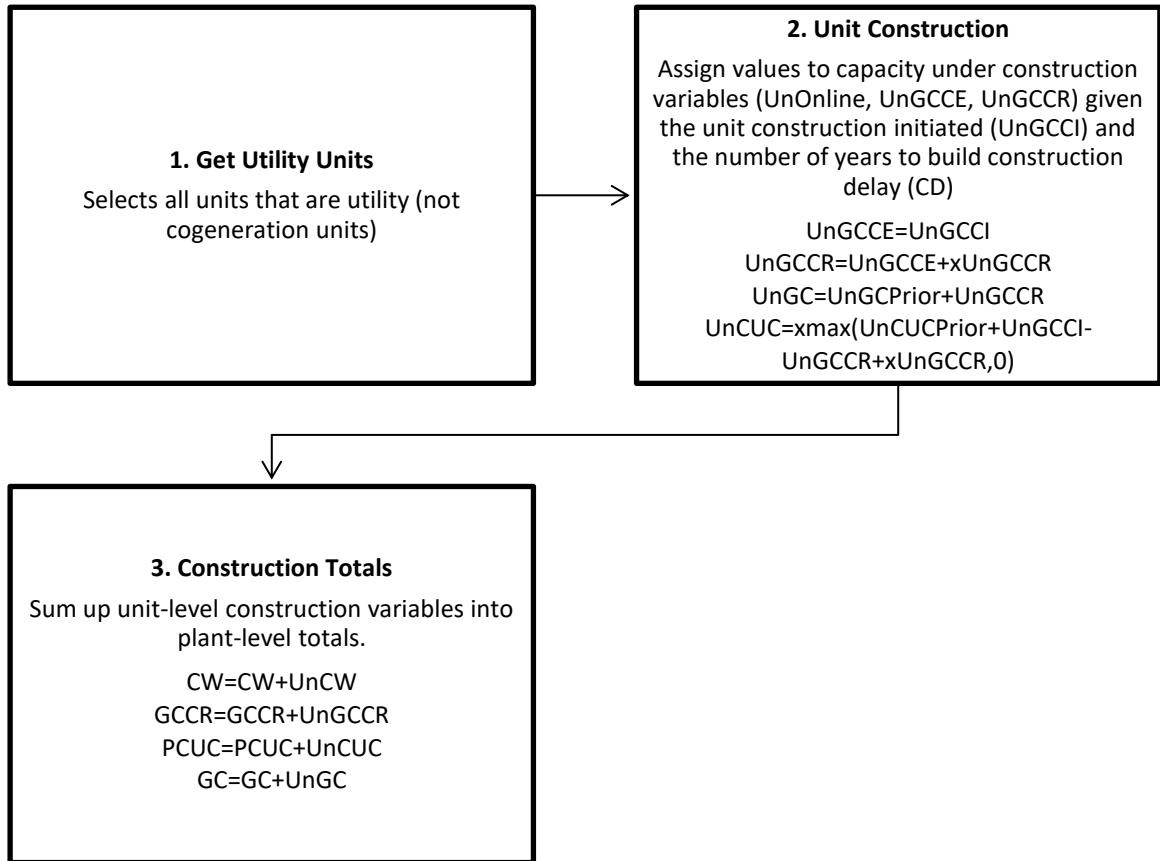
**Build new capacity:** Depending on the value of a user-specified build decision switch (BuildSw), the new capacity indicated (HDIPGC<sub>Power,Node,GenCo,Area,Year</sub>) is assigned

Three types of new capacity initiated: capacity forecasted from expansion routines (HDGCC<sub>I</sub>), intermittent capacity indicated if the cost of building wind or solar is significantly less than the clearing price (IPGCC<sub>I</sub>), and renewable capacity that is indicated by a renewable standard or FIT program (RnGCC<sub>I</sub>).

**Figure 23. Electric Capacity Expansion Routines – Capacity Initiated**



## Capacity Construction



## 7. Electric Generation Dispatch Submodule

Figure 24: Generation Dispatch in Electric Supply

The *EDispatch.jl* file contains the source code and Functions that make up the electric generation dispatch submodule. Its main Function is named *DispatchElectricity* and is called from the *SingleYear* Function inside *E2100.jl*. At the point when the electric generation dispatch submodule gets called, the load curves have been created, purchased power contracts have been developed and dispatched, initial pollution levels have been calculated, and based on demand requirements, new capacity has been built and is available for dispatch.

### Submodule Objective

The objective of the electric generation dispatch submodule is to determine an optimal mix of capacity and generation to be dispatched by existing generating facilities within each month (season), time slice, and node that will meet the current year's electricity demand. Outputs from the submodule additionally include optimal transmission flows and resulting marginal prices.

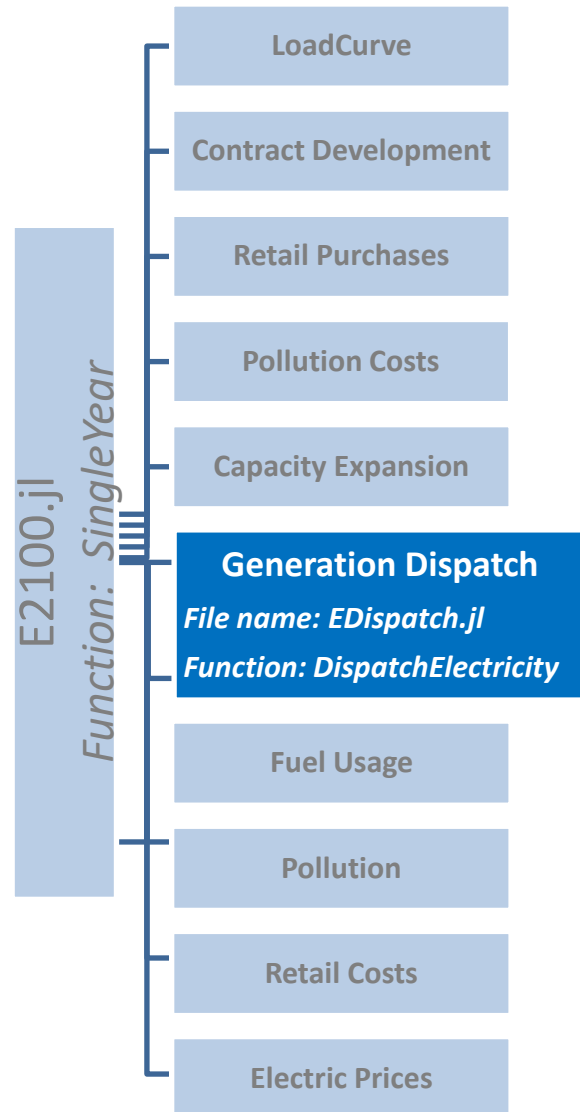
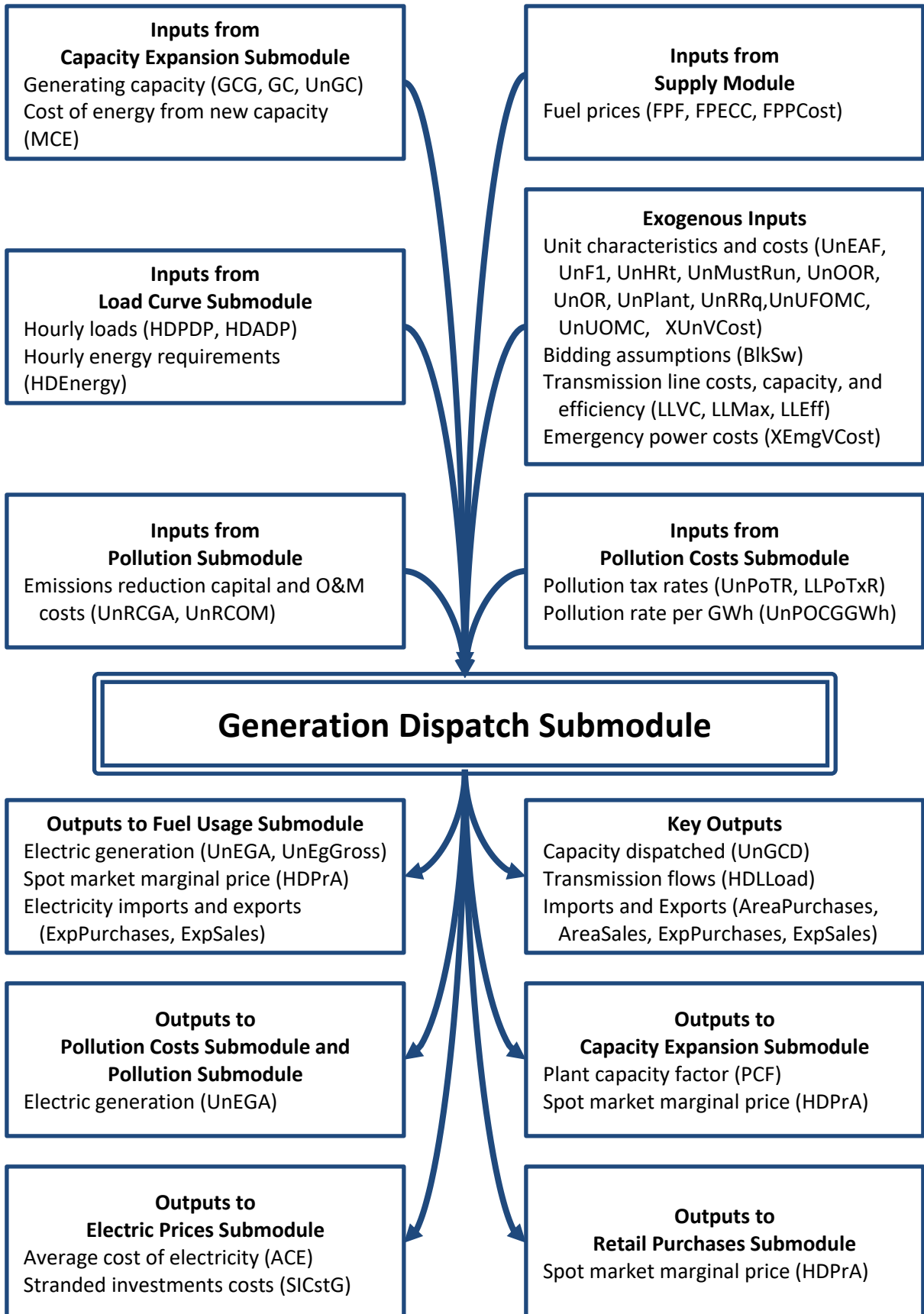


Figure 25 illustrates the relationship of the electricity generation dispatch submodule to the rest of ENERGY 2100.

Figure 25. Generation Dispatch Submodule Diagram of Inputs and Outputs

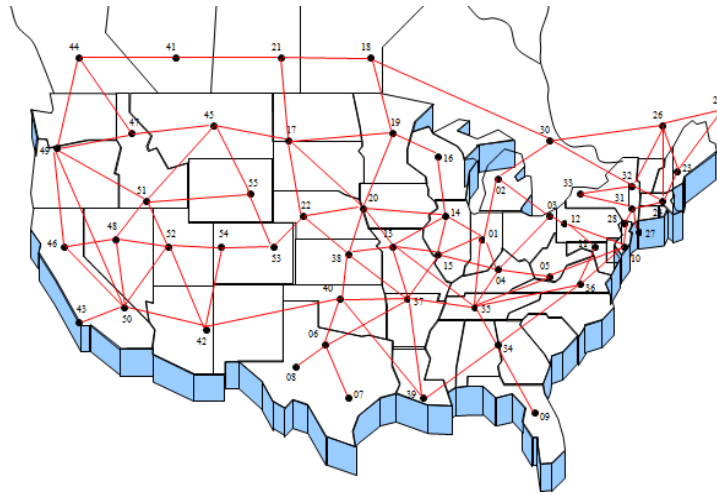


## Submodule Methodology

The term used to describe the process of estimating electricity production levels and the cost of producing that electricity is “production costing”. Within ENERGY 2100, production costing is simulated as a linear programming (LP) problem where the cost of the electric system is minimized subject to the constraints of meeting electric demands (generation balance constraint) while also meeting the overall system requirements. System requirements include the capability of only dispatching up to the generating capacity available (capacity availability constraint) and not exceeding the limits on the transmission lines (transmission constraint). An “emergency power” variable is used in the optimizing equation to ensure the LP is able to solve and is used as a check on the balance of the resulting transmission system. The LP solution provides the most cost effective mix of generating units needed to meet demand, and the outputs include unit capacity dispatched (UnGCD), transmission flows across the lines (HDLLoad), emergency generation (EmEGA), and the resulting marginal prices at each transmission node (HDPRA).

### Electric System

The electric system in ENERGY 2100 is simulated as a set of nodes connected by transmission lines. The transmission lines are assigned line capacities to and from nodes (LLMax), transmission efficiencies (LLEff), and wheeling costs across the lines (LLVC). The actual transmission nodes and lines defined in ENERGY 2100 are able to be customized; however, the standard version of the model has 110 nodes represented across North America which are typically aggregated into a smaller number of nodes of interest. Figure 26 illustrates an example of a transmission network represented in ENERGY 2100.

**Figure 26. Sample Transmission Network in ENERGY 2100**

Each node consists of its own unique, specified demand, which is met by a pool of resources consisting of the generating units as well as emergency generation. The generating units are able to send electricity to neighboring nodes via the transmission lines. Each electric generating unit is exogenously-specified with defining characteristics. These characteristics include a name (UnName), the node in which they are located at (UnNode), the type of plant (UnPlant), the heat rate (UnHRT), the online and retirements years of the unit (UnOnline, UnRetire), and the generating capacity (UnGC). Additionally, units are assigned an outage rate (UnOR), a fixed cost (UnUOFMC), and a variable cost (UnUOMC). The units may be flagged as “industrial” (UnCogen), meaning it is self-generating and does not provide generation to the electric grid. Units may also be flagged as “must run” (UnMustRun), meaning the unit always runs.

### Linear Program Specifications

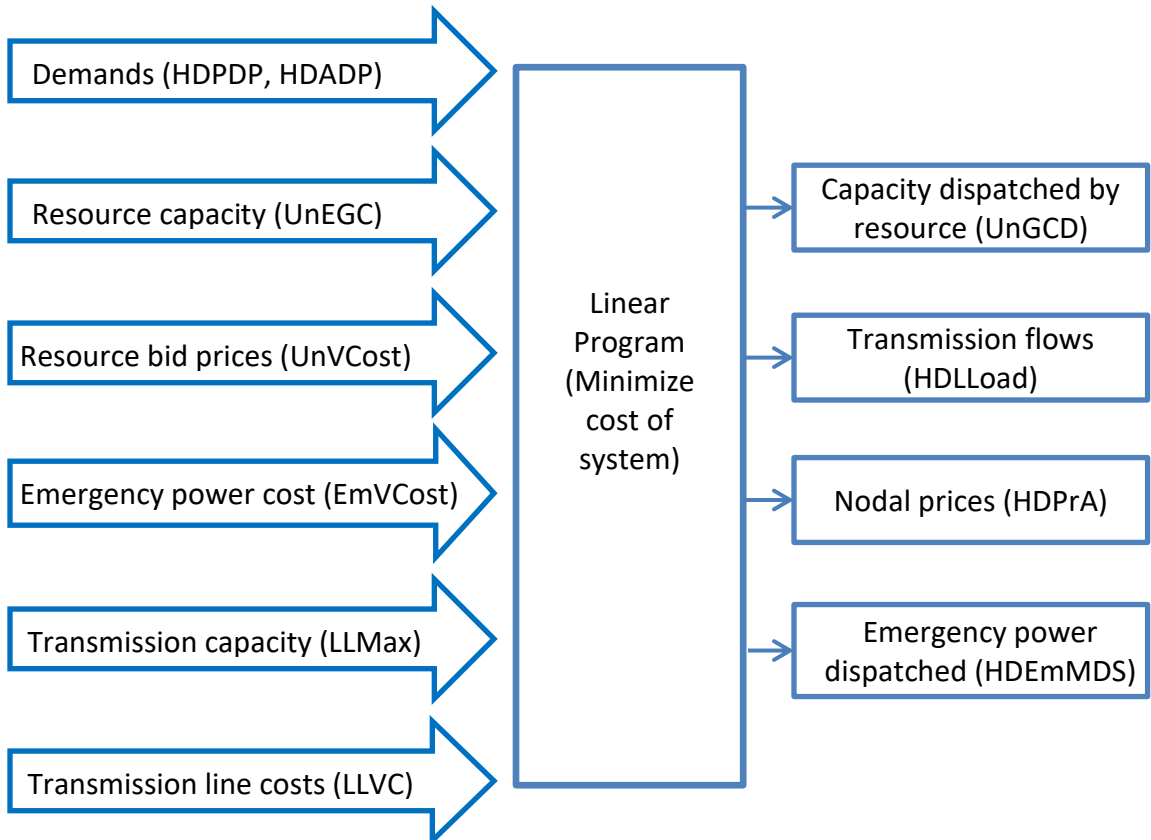
Each of the electric generating unit and transmission network inputs described above is used to find the optimal solution of generation, flows, and the resulting nodal prices. The generating capacity of the individual units and the cost of the units are translated into an amount of unit capacity available and a bid price by node, month, and time period. The entire geographic area of the model is dispatched as a single system each year. Figure 27 illustrates the inputs to and outputs from the LP.

The LP is solved for a single hour of each time period, month, and year of the simulation and includes all load and generation centers (nodes) in the model. The number and definition of the time periods and months may vary across model versions. The standard representation of months consists of two seasons – summer and winter. The standard

number of time periods (or time slices) is six – peak, near peak, high intermediate, low intermediate, high base load, and low base load.

The model dispatches one hour with the goal typically being to meet the average demand (HDADP) for that time period at the lowest cost. For that hour, each node has a load, an amount of capacity available that its units can generate, and a cost of generating each unit. The nodes are part of a transmission system which takes into account the transmission constraints, efficiencies, and costs. The objective function of the LP minimizes the cost of the system to meet the demand at each node within the limitations of the capacity constraints. Units having the same costs are aggregated and represented as a single unit for dispatch. The aggregated units make up the total generating capacity of a node for that time period.

**Figure 27. Linear Program Inputs and Outputs**



The LP must determine the optimal mix to meet total demand at a single node. If there were not enough generating capacity available at that node, then no LP solution would exist for the entire system. To ensure that the objective function solves and provides outputs to the model, each node is assigned the potential to produce emergency back-up power (EmEGA) with no capacity limit. Emergency power does not generate from

---

any specified physical electric units, but it can provide power to its own node if required. Emergency power prices are typically set much higher than generating costs of physical electric units. The resulting emergency power is used as a check on the model results – high emergency power indicates an imbalance in the system which should be investigated.

The **objective function** of the LP minimizes the cost of the system, where the system cost is defined as the sum of capacity dispatched (UnGCD) times unit bid prices (UnVCost), transmission flows (HDLLoad) times transmission costs (LLVC), and emergency power (HDEmMDS) times emergency power costs (HDEmVC). The inputs in this objective function are bid prices (UnVCost), transmission costs (LLVC), and emergency power costs (HDEmVC), and the variables being solved for are capacity dispatched (UnGCD), transmission flows (HDLLoad), and emergency generation (EmEGA).

This optimal solution is subject to a **generation balance constraint** meaning that, at each node, the amount of generation dispatched at or sent to the node minus the electricity sent from the node to another node must meet the demand for electricity at the given node. Since each transmission line has its own efficiency rating (LLEff), the line efficiency ratings are applied to the flows to account for line losses. Therefore generation balance constraint is put in the model as the sum of the generation dispatched at the node (UnGCD), transmission inflows to the node (HDLLoadin), and emergency power on the node (EmEGA) minus transmission line outflows from the node (HDLLoadout) must be great enough to meet the node's peak demand during peak time periods and average demand during non-peak time periods (HDPDP, HDADP), excluding cogeneration demands.

The optimal solution is subject to a **transmission constraint** which is the maximum flow (LLMax), in megawatts, on the line that can be run at any one time between two nodes. Transmission lines may have a requirement of a minimum level of flow between each node (LLMin), or there also may be loading of the transmission lines specified exogenously due to specific knowledge of a given transmission line (HDXLoad). The LP objective function is subject to the transmission constraint that the transmission flows on a given line must be less than or equal to the maximum capacity (LLMax) and it must be greater than or equal to the exogenous loading (HDXLoad).

The optimal solution is subject to a **capacity availability constraint**. The potential generating capacity at each node is the summation of the effective generating capacities (UnEGC) of the individual units. Generating units do not run all of the time, so they are given an outage rate (UnOR) to set the minimum amount of time that the unit must

spend offline. Some renewable plant types have an additional constraint, as their “fuel sources” have limitations on when they are available. These units are given an effective generating capacity (UnEGC, XUnEGC) to take into account the restrictions on their availability. The capacity availability constraint requires the generation dispatched (UnGCD) to be less than or equal to the effective generating capacity of the units (UnEGC).

Below, Figure 28 is a summary of the objective function and its constraints in equation form.

**Figure 28 - Linear Program objective function and constraints**

<b>LP objective function</b>	<p><i>The <b>objective function</b> of the LP is to minimize the value of</i></p> $\text{Cost of Production} = (\sum \text{Capacity Dispatched (UnGCD)} * \text{Bid Price (UnVCost)}) + (\sum \text{Transmission Flow (HDLLoad)} * \text{Transmission Cost (LLVC)}) + (\sum \text{Emergency Power (HDEmMDS)} * \text{Emergency Cost (HDEmVC)})$
<b>Generation balance constraint</b>	<p>Subject to the following constraints:</p> <ol style="list-style-type: none"> <li>1. Nodal capacity dispatched (UnGCD) + transmission inflows (HDLLoad<sub>in</sub>)*Line efficiency (LLEff) – transmission outflows (HDLLoad<sub>out</sub>) + Emergency power (HDEmMDS) <math>\geq</math> Nodal Demand (HDPDP, HDADP)</li> </ol>
<b>Transmission constraint</b>	<ol style="list-style-type: none"> <li>2. Transmission flow (HDLLoad) <math>\leq</math> Maximum transmission line limit (LLMax) Transmission flow (HDLLoad) <math>\geq</math> Exogenous flows (HDXLoad)</li> </ol>
<b>Capacity availability constraint</b>	<ol style="list-style-type: none"> <li>3. Unit capacity dispatched (UnGCD) <math>\leq</math> Unit capacity available (UnEGC)</li> </ol>

---

## Inputs to the Linear Program

The inputs to the LP include the following:

- system demands
- unit capacity available
- unit bid prices
- transmission line capacity and costs
- emergency power capacity and costs

Before being sent to the LP, each of these inputs is split by node, month, and time period, and is adjusted to incorporate differences in dispatching assumptions among the different types of generating plants. The standard plant types available for dispatch in the model include fossil-fueled units (gas/oil peaking, gas/oil combined cycle, gas/oil steam, and coal), nuclear plants, hydroelectric plants (base hydro, peak hydro, and pumped-storage hydro), and other renewable plant types (such as small hydro, wind, solar, geothermal, and biofuels).

**Input to LP #1** ***System Demands:*** Peak and average demand (HDPDP, HDADP) provided by node, month, and time period are used as inputs to the LP and were outputs from the demand module. These demands represent the total electricity demand at each node excluding cogeneration.

Demand is increased to account for recharging pumped hydro units

Before being sent to the LP, the system demands are adjusted to account for any units that require additional electricity for recharging. The storage technology represented in ENERGY 2100 that requires recharging is pumped-storage hydro, referred to as pumped hydro in the model. Pumped-storage hydroelectricity is able to store energy for use during high-peak time periods by pumping water from a lower elevation reservoir to a higher elevation. The pumping process uses electricity to restore its water supply. The model (in Function *RechargingRequirements*) keeps track of the electricity generated by storage technologies for time periods one through five. Then in time period six (the lowest load off-peak time period), additional demand requirements are added to the average system demand (HDADP) to account for restoring the pumped hydro units (in Function *IncreaseLoadsForRecharging*) which are then available to run as any other unit, whenever it is cost effective to do so when compared to the market price.

**Input to LP #2** ***Unit Capacity Available:*** Each generating unit is assigned an exogenously-specified capacity (UnGC) on a given node. To determine the amount of capacity available for

dispatch (UnEGC), the unit capacity is derated, or lowered, based on unit outage rates before it is bid into the system. For most units, the full derated capacity is bid into the system. However, some exceptions exist for plant types having limitations on capacity or alternative dispatching assumptions. In these cases, the capacity available may vary across time periods. The three plant types in which there are limiting factors include peak hydro, pumped hydro, and intermittent resources, such as wind.

<b>Peak hydro is allocated to peak time periods first</b>	<i>Peak Hydro:</i> A peak hydro facility is one which is able to store water and can vary its generation through the day, month, and possibly year. It might also be called storage hydro. These are treated differently than base hydro facilities which are essentially “run-of-the-river” and cannot vary their generation. A base hydro facility generates whenever there is water. Peak hydro units, having flexible storage, are bid to maximize the water used during peak periods and minimize water “spilled” during off peak hours. Water is moved between hours and seasons. Available capacity (water) is allocated first to the peak time periods during the year. These peak time periods are determined based on annual load duration curves created for each node. The adjustments to available capacity for peak hydro units are done in Function <i>AllocatePeakHydroAvailability</i> .
<b>Pumped hydro is set to zero in off-peak hours for recharging</b>	<i>Pumped Hydro (storage units requiring recharging):</i> Pumped-storage hydroelectricity is able to store energy for use during high-peak time periods by pumping water from a lower elevation reservoir to a higher elevation. The pumping process uses electricity to restore its water supply. During time period six (lowest load off-peak time period), pumped hydro units are restored so their available capacity (UnEGC) is set to zero.
<b>Wind optionally can be assigned to off-peak periods and can be assigned reserves</b>	<i>Intermittent Resources (Wind):</i> The treatment of wind units varies, but one option is to assign available capacity of wind units only to off-peak periods to approximate their intermittent generation. Another optional routine exists to simulate the required reserves to back up the intermittent resources. If reserve switches (UnRSwitch) are set within a policy, then the model will force specific units to supply reserves for intermittent resources, such as wind. The switch is turned off by default; however, this feature is available. Adjustments are made to the unit bids of intermittent units and to the effective generating capacity of both base load and wind units to force intermittent resources to be dispatched if there are enough reserves and to force the reserves to be dispatched if there are not enough reserves (Function <i>ReserveUnitsAvailabilityAndBids</i> ).

**Input to LP #3** **Unit Bid Prices:** There are options to be able to select the method used to calculate a bid price (UnVCost) in the model. These options are specified using a switch (BlkSw) with the default being equal to one. As a default, this bid price is based on a fraction of fixed costs (UnAFC) and variable costs (UnAVC). The fractions of fixed and variables costs (HDFCFr, HDVCFr) are specified exogenously. The bid price additionally assumes a 75% capacity factor applied to the fixed costs.

Most units are bid in at their marginal cost. Some units, such as nuclear, may have a reported fuel cost which overstates their marginal costs. These units can be bid at less than their marginal costs. Fixed cost may also be included in the bid, if desired. Bids can vary by period.

The bid price is overwritten for units designated as must-run (UnMustRun=1). Must-run units are user-specified and are the exception, as they are prioritized to always run. The other units then may compete based on their availability and price. To ensure the must-run unit is dispatched, they are assigned a bid price slightly less than the bid price.

**Input to LP #4** **Transmission line capacity and costs:** Transmission line constraints (LLMax) are an exogenous input to the model. The transmission costs (LLVC) are also an exogenous input. Transmission flows are a function of the dispatch algorithm; however exogenous “contract flows” can be added to force the flow of electricity between specific nodes. This may be needed when significant amount of storage hydro is available since the allocation of the water includes consideration of “contract flows”.

**Input to LP #5** **Emergency power capacity and cost:** Emergency power is used in the dispatch algorithm if the node including transmission flows is insufficient to meet nodal demands. The capacity available for emergency power (HDEmGC) is unlimited. The bid price for emergency power on a given node is assigned a slightly larger price than the minimum unit bid price at that node. Emergency power bid prices are higher in order to ensure that emergency power is only dispatched if there is a capacity shortfall.

### Linear Programming solution

Once the LP solves for the model, it provides two solutions for each entity in the equation. Each aggregated unit, transmission line, and emergency power generator has a primal and dual solution. The non-zero values of the primal solution for a single transmission line represent the capacity and generation dispatched provided by the specified resources from node “A” to “B” (UnGCD, UnEG, and EmEGA). If the primal solution for a resource is zero, then that resource did not run in the specified time

---

period. If the primal solution for a single transmission line from a node “A” to a node “B” is zero, then for that hour, no transmission was sent across the transmission line in the specified direction.

The price of generation at a node is determined using the dual solution. Comparing the most expensive unit at a node to the dual solution for a unit that did not run at that node indicates the marginal price at that node. When the primal solution is zero, then a dual solution exists. For a given resource, the dual solution value corresponds to the price (in \$/MWh) at that node that generation must increase in order for this unit to be turned on. For transmission from node A to B, the dual solution indicates the price that generation must increase at node “B”, before node “A” sends generation across that transmission line.

### **Imports and Exports**

Electricity imports and exports are obtained from the transmission flows (HDLLoad) output from the LP. Function *Flows* stores the transmission flows between the US and Canada nodes obtained from the LP for imports (ExpPurchases) and exports (ExpSales). Imports and exports can be specified exogenously (using the variable HDXLoad) for instances where there are known levels of imports or exports, such as through contracts. The exogenously-specified values are added to the flows determined by the LP.

### **Cogeneration**

Cogeneration is not included as part of the electric generation dispatch module. The LP determines the optimal mix of electric utility generating units only and is designed to meet the system energy demands excluding cogeneration (HDADP, HDPDP).

### **Submodule Functions**

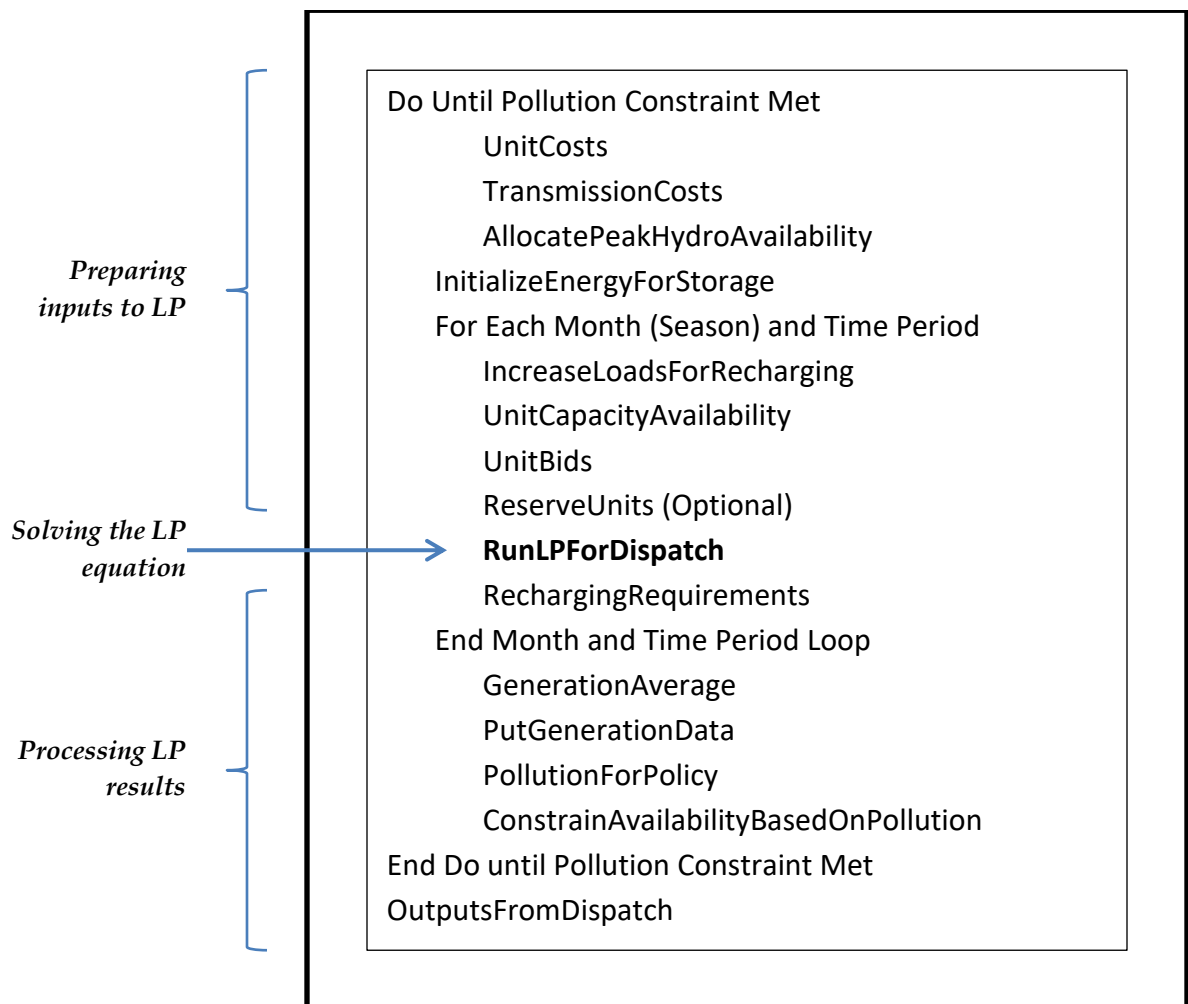
The Functions used to simulate electric generation dispatch are all housed within the *EDispatch.jl* file and called from the *DispatchElectricity* Function. The core of the electric generation dispatch submodule is the linear programming algorithm. All the Functions within this submodule perform one of three functions – 1) prepare inputs to the LP, 2) execute and solve the LP, or 3) process outputs from the LP.

The Functions that make up *DispatchElectricity* and their order of execution are shown in Figure 29.

The Functions that prepare the inputs to the LP assign values and make adjustments to system demand, transmission costs, unit bids, and resource capacity available, and they are listed below.

- Prepare inputs to LP*
- *UnitCosts*
  - *TransmissionCosts*
  - *AllocatePeakHydroAvailability*
  - *InitializeEnergyForStorage*
  - *IncreaseLoadsForRecharging*
  - *UnitCapacityAvailability*
  - *EmergencyPowerAvailability*
  - *UnitBids*
  - *ReserveUnitsAvailabilityAndBids*

**Figure 29. Function DispatchElectricity in EDispatch.jl**



The Function that executes the linear programming algorithm and dispatches the unit generation is:

- Execute LP*
- *RunLPForDispatch.*

The Functions that prepare that process LP results are listed below.

- Process LP results*
- *RechargingRequirements*
  - *GenerationAverage*
  - *PutGenerationData*
  - *PollutionForPolicy*
  - *ConstrainAvailabilityBasedOnPollution*
  - *OutputsFromDispatch*

Table 24 provides a summary description of each of the Functions called from the *DispatchElectricity*, and the key outputs are listed as well. Further details on each of these Functions can be found in the subsequent sections.

**Table 24. Description of Functions within Generation Dispatch**

Function Description	Key Outputs
<p><b>1. UnitCosts</b> Calculates unit fuel prices (UnFP) based on the previous year’s prices then uses fuel prices to calculate unit variable costs (UnAVC). Also calculates other unit financial variables, such as net assets (UnNA), depreciation (UnSLDPR), and unit fixed costs (UnAFC). Sets the variable costs (UnAVC) of must-run plants to zero to ensure they will be dispatched.</p>	<p>Unit costs (UnAVC, UnAFC, UnNA, UnSLDPR) which are used as input to <i>UnitBids</i> Function.</p>
<p><b>2. TransmissionCosts</b> Gets transmission line costs (XLLVC) and adjusts for emissions tax rates (LLPoTxR) if applicable. The transmission line costs are then used directly as input to the <i>RunLPForDispatch</i> Function.</p>	<p>Transmission line costs (LLVC)</p>
<p><b>3. AllocatePeakHydroAvailability</b> Allocates peak hydro plants by prioritizing available capacity (UnGC) to annual peak time periods then to lower load hours. Uses nodal demand (HDPDP) to create annual load duration curve for identifying annual peak time periods. Distributes this available capacity (UnEGC) assigned to annual peak periods back to model’s definitions of months and time periods for dispatch.</p>	<p>Peak hydro effective generating capacity (UnEGC<sub>PeakHydro</sub>)</p>

Function Description	Key Outputs
<p><b>4. IncreaseLoadsForRecharging</b></p> <p>Adjusts nodal demand (HDADP) in low load hours for restoring energy needed to charge pumped-storage hydro facilities. Also saves the amount of electricity required to recharge the storage technologies (StorEnergy). The revised nodal demands are used directly as input to the linear program in <i>RunLPForDispatch</i>. The electricity required (StorEnergy) is saved as output, but not used.</p>	<p>Average demand in low load hours (HDADP)</p> <p>Electricity required to recharge storage technologies (StorEnergy)</p>
<p><b>5. UnitCapacityAvailability</b></p> <p>Determines capacity available (UnEGC) for all units except peak hydro units which were assigned in <i>AllocatePeakHydroAvailability</i>, constrains the available capacity (UnEGC<sub>Storage</sub>) for storage units in low load time periods, assigns capacity available for emergency power (HDEmGC), and overwrites calculated availability with exogenous unit availability (XUnEGC) if specified.</p>	<p>Capacity available (UnEGC)</p> <p>Emergency power available (HDEmGC)</p>
<p><b>6. UnitBids</b></p> <p>Calculates a bid price (UnVCost) for each of the units based on a fraction of fixed and variable costs (HDFCFr, HDVCFr) which are specified exogenously. Must run units are assigned a bid price slightly less than the bid price of other units to ensure those units are chosen for dispatch. Bid prices for emergency power (HDEmVC) are calculated as well. The bid price for emergency power is assigned a higher price than the minimum unit bid price to ensure emergency power is not dispatched unless there is a capacity shortage. These bid prices are direct inputs to the linear programming algorithm in <i>RunLPForDispatch</i>.</p>	<p>Unit bid price/unit variable cost (UnVCost)</p> <p>Emergency power variable cost (HDEmVC)</p>
<p><b>7. ReserveUnits (Optional)</b></p> <p>This Function is only executed for policy analysis, and the equations are dependent upon a renewable switch being turned on to force specific units to supply reserve requirements (UnRSwitch eq 1) for intermittent resources, such as wind. It is also dependent on the assignment of reserves available (UnRAvgt 0). If the renewable switch is turned on, then this Function determines whether or not there are enough reserves available for the intermittent resources to be dispatched. It forces the reserves on when needed and forces the intermittent resources on when reserve requirements are met.</p>	<p>Reserves needed (ResNeed)</p> <p>Capacity available and bid prices (UnEGC<sub>Reserves</sub>, UnEGC<sub>Wind</sub>, UnVCost<sub>Wind</sub>)</p>

Function Description	Key Outputs
<p><b>8. RunLPForDispatch</b> Determines the optimal mix of generation (UnEG) and transmission flows (HDLLoad) that meets nodal energy demands by minimizing overall costs of the system. Key inputs to this Function include nodal demands (HDPDP, HDADP), unit bid prices (UnVCost), transmission line costs (LLVC), transmission limits (LLMax), and capacity available/effective generating capacity (UnEGC). A linear programming (LP) algorithm is used for the optimization with the constraints of meeting nodal demand, staying within transmission line limit constraints, and using the capacity available. Emergency power (HDEmGC) is available as a resource at each node at a cost higher than other resources (HDEmVC) to ensure that the LP solves.</p>	<p>Unit capacity and generation dispatched (UnGCD, UnEG), Transmission flows (HDLLoad), Nodal prices (HDPrA), Emergency capacity and generation dispatched (HDEmMDS, EmEGA)</p>
<p><b>9. RechargingRequirements</b> This Function accumulates the electricity generated by storage technologies for time periods one through five (peak and intermediate time periods, not low-load hours).</p>	<p>Energy required to recharge storage technologies (StorEnergy); Electricity generated by storage technologies (StorEG)</p>
<p><b>10. GenerationAverage</b> Calculates average across dispatch iterations. There currently is only one iteration, so this Function is no longer needed.</p>	<p>Generation by unit, time period, and month (UnEGTM)</p>
<p><b>11. PutGenerationData</b> Simply writes the generation variables to disk (UnEGTM, StorEG, and StorEnergy). Need to revise this Function to put the write statements with the equations.</p>	<p>Function needs revised.</p>
<p><b>12. PollutionForPolicy</b> Calculates emissions based on generation dispatched and saves to policy variable.</p>	<p>Emissions from generation dispatched (PolGHG)</p>
<p><b>13. ConstrainAvailabilityBasedOnPollution</b> This Function is used for policies in which a constraint on emissions exists. If there is a constraint, then the variable PolAvFactor is adjusted and the dispatch is re-executed.</p>	<p>Availability Factor due to pollution constraint (PolAvFactor)</p>

Function Description	Key Outputs
<b>14. OutputsFromDispatch</b> Calculates outputs resulting from dispatch, such as generation totaled across different sets, unit costs, stranded investments for retired units, imports and exports.	Unit costs, imports, exports

### Function UnitCosts

The *UnitsCosts* Function reads in fuel costs (FPF) and unit costs. Unit fuel prices (UnFP) are calculated based on the previous-year's sector-level delivered fuel prices (FPF) determined in the Electric Pricing Submodule. These unit prices (UnFP) are used to calculate a unit-level average variable cost (UnAVC). Net assets (UnNA), depreciation (UnSLDPR), fixed costs (UnAFC) of the units are also calculated.

The values of the variables calculated in this *UnitCosts* Function are used to calculate unit bid prices in the *UnitBids* Function which are ultimately sent to the linear program for dispatch.

The equations for each of these outputs – unit fuel price, net assets, depreciation, unit average fixed costs, and unit average variable costs respectively are listed below.

Key Equations:  
UnitsCosts

Unit fuel price:

$$\text{UnFP} = \text{FPF} * \text{UnFIFr}$$

Unit fuel price of nuclear:

$$\text{UnFP}_{\text{Nuclear}} = \text{XUFC} * \text{Infla} / \text{UnHRT} * 1000$$

Unit net assets:

$$\text{UnNA} = \text{UnNA} + \text{UnCWGA} - \text{UnSLDPR} + \text{UnRCGA}$$

Unit depreciation rate:

$$\text{UnSLDPR} = \text{UnNA} * \text{DPRSL}$$

Unit fixed Costs:

$$\text{UnAFC} = (\text{UnNA} * \text{CCR} + \text{UnSLDPR}) / \text{UnGC} * 1000 + \text{UnUFOMC} * \text{Infla} + \text{UnRCOM}$$

Unit variable costs:

$$\text{UnAVC} = \text{UnFP} * \text{UnHRt} / 1000 + \text{UnUOMC} * \text{Infla} + \text{UnPoTR} + \text{UnPoTRExo} * \text{Infla}$$

Table 25 lists the variable names and definitions of the key input and output variables from the *UnitCosts* Function.

**Table 25: UnitCosts Function - Input and Output Variables**

<b>UnitCosts Inputs and Outputs</b>
<b>Key inputs</b>
CCR (Plant, GenCo, Year) = Capital Charge Rate (1/Yr)
DPRSL (GenCo, Year) = Straight Line Depreciation Rate (1/Yr)
PPF (Prices, Area, Year) = Delivered Fuel Price (\$/mmBtu)
Infla (Year) = Inflation Index (\$/\$)
UnFIFr [Unit, FuelEP, Year) = Unit Fuel Fraction (Btu/Btu)
UnGC [Unit, Year) = Unit Generating Capacity (MW)
UnHRt [Unit, Year) = Heat rate (btu/kWh)
UnPoTRExo [Unit, Year) = Exogenous Pollution Tax Rate (Real\$/MWh)
UnPoTR [Unit, Year) = Pollution Tax Rate (\$/MWh)
UnRCOM [Unit, Year) = Emission Reduction O&M Costs (M\$)
UnUFOMC [Unit, Year) = Fixed O&M Costs (Real \$/KW)
UnUOMC [Unit, Year) = Variable O&M Costs (Real \$/MWh)
XUFCPlant (GenCo, Year) = Unit Fuel Cost (\$/MWh)
<b>Key outputs</b>
UnAFC [Unit, Year) = Average Fixed Costs of Unit (\$/KW)
UnAVC [Unit, Year) = Average Variable Costs of Unit (\$/MWh)
UnFP [Unit, Year) = Fuel Price of Unit (\$/mmBtu)
UnNA [Unit, Year) = Net Asset Value of Unit (M\$)
UnSLDPR [Unit, Year) = Depreciation of Unit (M\$/Yr)

### Function TransmissionCosts

The *TransmissionCosts* Function reads in transmission line costs and adjusts for emissions tax rates if applicable. The costs calculated in this Function (LLVC) are input to the linear program in *RunLPForDispatch* for calculation of the overall system cost of dispatch.

There is only equation used in *TransmissionCosts* which adjusts exogenous transmission costs with any pollution costs or taxes that may exist. The equation is shown below.

**Key Equation:** *Transmission costs:*  
**TransmissionCosts**       $LLVC = (XLLVC + \sum_{Mkt} LLPoTxR + LLPoTxRExo) * Infla$

Table 26 lists the variable names and definitions of the key input and output variables.

**Table 26: TransmissionCosts Function - Inputs and Output Variables**

<b>TransmissionCosts Inputs and Outputs</b>
<p><b>Key inputs</b></p> <p>XLLVC [Node, NodeX, Year) = Transmission rate (\$/MWh)</p> <p>Infla (Year) = Inflation Index (\$/\$)</p> <p>LLPoTxR [Node, NodeX, Market, Year) = Pollution Costs for Transmission (\$/MW)</p> <p>UnPoTR [Unit, Year) = Pollution Tax Rate (\$/MWh)</p>
<p><b>Key outputs</b></p> <p>LLVC[Node, NodeX, Year) Transmission Rate (\$/MWh)</p>

### Function AllocatePeakHydroAvailability

Peak hydro units are able to store energy for use during peak time periods. The *AllocatePeakHydroAvailability* Function simulates an optimal allocation of available water by allocating the capacity of the peak hydro units to peak time periods first then to the base load time periods.

The allocation is done by creating an annual load duration curve for each node as well as multiple time slices (currently set to 12 time slices from the highest peak to the lowest load time period). The capacity and energy is then be mapped back to the months (seasons) and time periods defined for dispatch (by default six time periods).

Inputs to the Function include unit generating capacity (UnGC), own use rate for generating capacity (UnOURGC), unit energy availability factor (UnEAF), and peak load by node, time period, month, and year (HDPDP).

The key output from this Function is the effective generation capacity (UnEGC) of peak hydro units which is used as input to the linear program in *RunLPForDispatch*. Table 27 summarizes the variable names and definitions of the key input and output variables.

**Table 27: Inputs and Output Variables - AllocatePeakHydroAvailability**

<b>AllocatePeakHydroAvailability Inputs and Outputs</b>
<p><b>Key inputs</b></p> <p>HDPDP [Node, TimeP, Month, Year) = Peak Load in Interval (MW)</p> <p>UnGC [Unit, Year) = Unit Generating Capacity (MW)</p> <p>UnOURGC [Unit, Year) = Own Use Rate for Generating Capacity (MW/MW)</p> <p>UnEAF [Unit, Month, Year) = Energy Availability Factor (MWh/MWh)</p>
<p><b>Key output</b></p> <p>UnEGC<sub>Unit,TimeP,Month,Year</sub> = Effective Generating Capacity (MW)</p>

---

## Function IncreaseLoadsForRecharging

Storage technologies, such as pumped storage hydro or batteries, require energy to be recharged. Currently, the only storage technology modeled in ENERGY 2100 is pumped hydro. The pumped hydro units are restored or recharged during off-peak hours and are available to run as any other unit, or in other words, whenever it is cost effective to do so compared to the market price.

The purpose of *IncreaseLoadsForRecharging* is to increase energy demand (HDADP) during the low load hours (time period six) to simulate the pumped hydro units' demand for energy required (StorEnergy) for replenishing hydro storage. The adjusted demands (HDADP) are ultimately used as input to the linear program in *RunLPForDispatch*.

This Function is only called for time period six after time periods one through five have been dispatched. The generation dispatched in time periods one through five from storage units is accumulated in the *RechargingRequirements* Function after the LP execution and stored (StorEG).

Because *IncreaseLoadsForRecharging* is called after time periods one through five have been dispatched, the amount of energy used by the pumped hydro units is known and in time period six, it is replenished. The additional energy needed to recharge the pumped storage hydro units (StorEnergy) is calculated by dividing the storage units' generation (StorEG) by an exogenous recharging efficiency (StorEFF). An adjustment is then made to the average nodal demand (HDADP) for time period six by adding the additional energy needed to recharge storage technologies (StorEnergy).

The key equations used in IncreaseLoadsForRecharging are listed below.

### Key Equations: IncreaseLoadsFor Recharging

#### Storage energy used:

$$\text{StorEnergy} = \text{StorEG} / \text{StorEff}$$

#### Average energy demand:

$$\text{HDADP} = \text{HDADP} + \text{StorEnergy} / \text{HDHours} * 1000$$

The outputs of this Function include the amount of storage energy that was needed (StorEnergy) and a revised average nodal demand (HDADP) in time period six (low load hours). The storage energy needed (StorEnergy) is an output but not used by any other Functions in ENERGY 2100. The adjusted nodal demands (HDADP) are used as input to the linear program in *RunLPForDispatch*. Table 28 summarizes the variable names and definitions of the key input and output variables.

**Table 28: Input and Output Variables - IncreaseLoadsForRecharging**

<b><i>IncreaseLoadsForRecharging Inputs and Outputs</i></b>
<p><b>Key inputs</b></p> <p>HDHours [TimeP6, Month] = Number of Hours in the Interval (Hours)</p> <p>StorEff [Node, Month, Year] = Recharging Efficiency of Storage Techs (kWh/kWh)</p> <p>HDADP [Node, TimeP6, Month, Year] = Average Load in Interval (MW)</p> <p>StorEG [Node, Month, Year] = Electricity Generated from Storage Techs (GWh)</p>
<p><b>Key outputs</b></p> <p>HDADP [Node, TimeP, Month, Year] = Average Load in Interval (MW)</p> <p>StorEnergy [Node, Month, Year] = Electricity Required to Recharge Storage Techs (GWh)</p> <p>HDHours [TimeP6, Month] = Number of Hours in the Interval (Hours)</p> <p>StorEff [Node, Month, Year] = Recharging Efficiency of Storage Techs (kWh/kWh)</p> <p>HDADP [Node, TimeP6, Month, Year] = Average Load in Interval (MW)</p> <p>StorEG [Node, Month, Year] = Electricity Generated from Storage Techs (GWh)</p>

### **Function UnitCapacityAvailability**

The *UnitCapacityAvailability* Function assigns a value for effective generating capacity, or the amount each unit bids into the system by node, time period, and month/season for each of the units of all plant types except peak hydro which previously was assigned in *AllocatePeakHydroAvailability*.

*UnitCapacityAvailability* performs four functions. First it calculates effective generating capacity. The capacity available or capacity bid into the market (UnEGC) is calculated using the unit capacity (UnGC) derated by the outage rates (UnOR, UnOOR) and an availability factor (AvFactor) times a bid fraction (HDGCFr).

Second, this Function constrains the available capacity for storage units (which are energy limited) by using an energy availability factor (UnEAv<sub>Storage</sub>) during different time periods. The energy availability factor is calculated (UnEAv<sub>Storage</sub>) during peak time periods as the unit capacity (UnGC) times the unit's energy availability factor (UnEAF) times the hours (HDHours) in the season. During non-peak periods, the amount of energy available (UnEAv<sub>Storage</sub>) is energy available in the previous time period minus the energy dispatched (UnEG) in the previous time period. Time period one (peak) is dispatched first, so when the equations for time periods two through six are executed, the dispatch for the previous periods are known.

The amount of energy actually bid (UnEGC) is specified as the bid fraction (HDGCFr) constrained by the energy available (UnEAV).

Third, *UnitCapacityAvailability* assigns a capacity available value to emergency power available (HDEmGC). Emergency power is assumed to be available at each node. This Function assigns the emergency power available (HDEmGC) to be the peak load at each node, time period, and month (LDCMS). Having the emergency power available ensures there will always be enough capacity at each node for the LP to solve.

Finally, if any units have exogenously-specified capacity (XUnEGC), then the calculated effective capacities (UnEGC) are replaced by those values.

The key equations in the *UnitCapacityAvailability* Function are listed below.

**Key Equations:**  
*UnitCapacity Availability*

Effective generating capacity:

$$\text{UnEGC} = \text{UnGC} * (1 - \text{UnOURGC}) * (1 - \text{UnOR}) * (1 - \text{UnOOR}) * \text{AvFactor} * \text{PoIAvFactor} * \text{HDGCFr}$$

Energy availability of storage units:

$$\text{UnEAV}_{\text{Peak}} = \text{UnGC} * (1 - \text{UnOUREG}) * \text{UnEAF} * (1 - \text{UnOOR}) * \text{Hours} / 1000$$

$$\text{UnEAV}_{\text{Non-Peak}} = (\text{UnEAV}_{\text{TimeP-1}}) - (\text{UnEG}_{\text{TimeP-1}})$$

Effective generating capacity of storage units:

$$\text{UnEGC}_{\text{Storage}} = \text{Minimum} (\text{UnEGC}_{\text{Storage}} * \text{HDGCFr}, \text{UnEAV}_{\text{Storage}} / \text{HDHours} * 1000)$$

Capacity available for emergency power:

$$\text{HDEmGC} = \sum_{\text{ReCo}} \text{LDCMS}$$

The key outputs are (UnEGC) for all units except peak hydro and the amount of available emergency power (HDEmGC). Table 29 summarizes the variable names and definitions of the key input and output variables.

**Table 29: Input and Output Variables - UnitCapacityAvailability**

<b><i>UnitCapacityAvailability Inputs and Outputs</i></b>
<b>Key inputs</b>
AvFactor (Plant, TimeP, Month, Area, Year) = Availability Factor (MW/MW)
HDGCFr (Plant, GenCo, Node, TimeP, Month, Year) = Fraction of Available Generating Capacity Bid (MW/MW)
HDHours (TimeP, Month) = Number of Hours in Interval (Hours)
Hours (Month) = Hours per Monthly Period (Hours)
LdCMS (Day, Month, Node, Area, Year) = Marketer System Load Curve (MW)
PoIAvFactor (Plant, Area, Year) = Availability Factor from Pollution Constraint (MW/MW)
UnEAV [Unit, TimeP, Month, Year) = Energy Availability Factor (MWh/MWh)
UnOURGC [Unit, Year) = Own Use Rate for Generating Capacity (MW/MW)

**UnitCapacityAvailability Inputs and Outputs**

UnOR [Unit, Year) = Outage Rate (MW/MW)

UnOOR [Unit, Year) = Operational Outage Rate (MW/MW)

XUnEGC [Unit, TimeP, Month, Year) = Exogenous Effective Generating Capacity (MW)

**Key outputs**

UnEGC [Unit, TimeP, Month, Year) = Effective Generating Capacity (MW)

HDEmGC [Node, Month, Year) = Emergency Power Available (MW)

**Function UnitBids**

The *UnitBids* Function calculates a bid price (UnVCost) for each of the units. As a default, the bid price is based on a fraction of fixed and variable costs (HDFCFr, HDVCFr) which are specified exogenously. There are other options for how to calculate a bid price, and those options are specified by changing a switch (BlkSw), where the default is equal to one. Must run units are assigned a bid price slightly less than the bid price of other units to ensure the linear program will choose those units for dispatch. This Function also calculates a bid price for emergency power (HDEmVC) at each node. The bid price for emergency power on a given node is assigned a slightly larger price than the minimum unit bid price at that node. Emergency power bid prices are higher in order to ensure that emergency power is only dispatched if there is a capacity shortfall.

By default, the bid prices (UnVCost) are equal to fixed costs (UnAFC) assuming a 75% capacity factor times an exogenously-specified fraction of fixed costs (HDFCFr) plus variable costs (UnAVC) times an exogenously-specified fraction of variable costs (HDVCFr) plus a very small random number to break ties. If a pollution tax rate is specified (UnPoTR), this rate is added onto the bid price as well.

Must Run units (UnMustRun=1) are bid in at a little bit less than the minimum bid of all the other units to ensure that they are the first dispatched.

Emergency Power is assumed to be available at each node at a cost (HDEmVC) greater than any of the local bids (the maximum of the unit bids plus ten (UnVCost + 10), the marginal cost of electricity for peaking units plus ten (MCE<sub>Peak</sub> + 10), or an exogenously-specified emergency power cost (XEmgVCost).

The key equations used in the *UnitBids* Function are as follows.

**Key Equations:**      *Bid price of the units:*  
*UnitBids*              
$$\text{UnVCost} = \text{UnAVC} * \text{HDVCFr} + (\text{UnAFC} / (8760 * .75) * 1000) * \text{HDFCFr} + \text{abs}(\text{RanNum}) / 10000 + \text{UnPoTR}$$

*Bid price for must-run units:*

$$\text{MinBid} = \text{UnVCostMinimum}$$

$$\text{UnVCost}_{\text{MustRun}} = \text{MinBid} - 0.001$$

*Bid price for emergency power:*

$$\text{HDEmVC} = \text{Maximum} (\text{XEmgVCost} \text{ or } \text{UnVCost} + 10.00 \text{ or } \text{MCE}_{\text{Peak}} + 10.00)$$

Both the unit bid prices (UnVCost) and emergency power prices (HDEmVC) are used as direct inputs to the linear programming algorithm in the *RunLPForDispatch* Function. Table 30 summarizes the variable names and definitions of the key input and output variables.

**Table 30: Input and Output Variables - UnitBids**

<b><i>UnitBids Inputs and Outputs</i></b>
<p><b>Key inputs</b></p> <p>BlkSw [GenCo,Year] = Block Switch Indicating Method of Calculating Bids</p> <p>HDFCFr [Plant,GenCo,TimeP,Year] = Fraction of Fixed Costs Bid</p> <p>HDVCFr [Plant,GenCo,Node,TimeP,Month,Year] = Fraction of Variable Costs Bid</p> <p>UnAFC [Unit, Year] = Unit Average Fixed Costs (\$/KW)</p> <p>UnAVC [Unit,Year] = Unit Average Variable Costs (\$/MWh)</p> <p>UnMustRun [Unit,Year] = Must Run Switch (1=Must Run)</p> <p>UnPoTR [Unit,Year] = Pollution Tax Rate (\$/MWh)</p> <p>XUnVCost [Unit,TimeP,Month,Year] = Exogenous Market Price Bid (\$/MWh)</p> <p><b>Key outputs</b></p> <p>UnVCost [Unit,TimeP,Month,Year] = Bid Price of Power Offered to Spot Market (\$/MWh)</p> <p>HDEmVC [Node,TimeP,Month,Year] = Dispatch Price for Emergency Power (\$/MWh)</p>

### Function ReserveUnits

The *ReserveUnits* Function is only executed for policy analysis, and the equations are dependent upon a renewable switch being turned on to force specific units to be a back-up supply for reserve requirements (UnRSwitch eq 1) for intermittent resources, such as wind. It is also dependent on the assignment of reserves available (UnRAv gt 0). If the renewable switch is turned on, then this Function determines whether or not there are enough reserves available for the intermittent resources to be dispatched. It forces the reserves on when needed and forces the intermittent resources on when reserve requirements are met.

The *ReserveUnits* Function is executed only during the low load time period (time period six) and determines the reserves that are needed and available for intermittent resources that require reserves, such as wind. At the point in the model execution when this Function is called, time periods one through five have already been dispatched. The intermittent resources are only dispatched if there are enough reserves available. This Function calculates whether or not there are enough reserves from base load units left by time period six to run the intermittent resources, such as wind. If there are enough reserves, the intermittent resources will be forced on by setting the unit bid price to be zero ( $UnVCost=0$ ). If there are not enough reserves, then the intermittent resources will be forced off by setting the capacity available to be zero ( $UnEGC=0$ ).

Table 31 summarizes the variable names and definitions of the key input and output variables.

**Table 31: Input and Output Variables - ReserveUnits**

<b><i>ReserveUnits Inputs and Outputs</i></b>
<p><b>Key inputs</b></p> <p>UnRSwitch [Unit] = Switch for Units Forced to Supply Reserve Requirements (1=Yes)</p> <p>ExpUVCost [Month,Year] = Expected Spot Price for Low Load Hour Time Period (\$/MWH)</p> <p>UnAVC [Unit,Year] = Average Variable Costs (\$/MWh)</p> <p>UnEGC [Unit<sub>BaseLoad</sub>,TimeP,Month,Year] = Effective Generating Capacity (MW)</p> <p>UnRAv [Unit] = Reserves Available (MW/MW)</p> <p>UnRRq [Unit] = Reserve Requirements (MW/MW)</p> <p>ResReq [Node,Month,Year] = Low Load Hour Reserve Requirements (MW)</p>
<p><b>Key outputs</b></p> <p>UnEGC [Unit<sub>Wind</sub>,TimeP,Month,Year] = Effective Generating Capacity (MW)</p> <p>UnVCost [Unit<sub>Wind</sub>,TimeP,Month,Year] = Bid Price of Power Offered to Spot Market (\$/MWh)</p> <p>UnResFlag [Unit,TimeP,Month,Year] = Flag to Indicate if Unit is Forced On to Provide Reserves (1=Yes)</p> <p>UnRVCost [Unit,Month] = Effective Cost of Reserves (\$/MWH)</p>

### Function RunLPForDispatch

The *RunLPForDispatch* Function is the heart of the electric generation dispatch submodule. Its purpose is to determine the optimal mix of generation that will meet the energy demand on each of the nodes by minimizing overall costs of the system. A linear programming (LP) algorithm is used to minimize the system costs subject to meeting nodal demand, transmission line limit constraints, and capacity availability

---

constraints. Emergency power is available as a resource at each node to ensure that the LP solves.

The key inputs to this Function and the source of those inputs include the following:

- nodal demands (HDPDP, HDADP) from the Demand Sector,
- unit bid prices (UnVCost) and emergency bids (HDEmVC) from *UnitBids*,
- unit capacity available or effective generating capacity (UnEGC) from *UnitCapacityAvailability* and *AllocatePeakHydroAvailability*,
- transmission costs (LLVC) from *TransmissionCosts*, and
- Transmission line limits (LLMax) which are exogenous inputs.

Outputs from this Function include unit generation and capacity, transmission flows, and nodal prices.

- Capacity dispatched (UnGCD)
- Transmission flow (HDLLoad)
- Emergency power (HDEmMDS)
- Marginal prices (HDPRA)

The source code for the LP is housed in EGen5LP.jl. The *RunLPForDispatch* Function makes a call to *LPCTRL* inside EGen5LP.jl to initiate the LP. The key Functions that make up *LPCTRL* include the following:

1. *AggregateDataForLP*: In order to reduce the number of variables and constraints in the Linear Program which will resolve the transmission constraints, the capacities and costs at each node are combined into aggregate units (UnAgg) with similar costs.
2. *WriteLPInputFile*: This Function writes to a text file the objective function of the LP used as input to the LP.
3. *RunLP*: Executes the LP solver
4. *ReadLPOutputFile*: Reads the output file from the LP to retrieve generation, transmission flows, prices, and emergency power
5. *OutputLPData*: This Function processes the aggregated outputs and splits the results into individual units.

As described in the methodology section, the objective function of the LP minimizes the cost of the system subject to generation balance, transmission, and capacity availability constraints.

**LP Aggregation**

In order to reduce the number of variables and constraints in the linear program, the capacities and costs at each node are combined into aggregate units (UnAgg) with near identical costs. Instead of all of the units in the model, these aggregated units are entered into the external LP solver program to cut down on calculation time. The individual units that make up each aggregated unit are kept track of before sending the aggregated units to the LP solver. The contribution of generating capacity of each unit to the aggregated sum is recorded to disaggregate the LP solution back to the unit level for use in ENERGY 2100.

The aggregation process starts for each node by sorting available units by costs (UnVCost) which are compared to determine whether or not to combine into an aggregate unit. This sorting and aggregating process is carried out by the model on a yearly basis.

In order for a unit to be combined into an aggregate, its cost (UnVCost) must either be less than 1.02 times the aggregate unit's cost (AgVCost) or less than the aggregate unit's cost + 0.001. If this criterion is met, then the effective generating capacity of the aggregate unit (AgEGC) is equal to its current capacity plus the capacity of the unit (UnEGC).

Once the LP has solved, the unit generating capacity dispatched is calculated for the individual units that made up the larger aggregate units. The unit generating capacity dispatched (UnGCD) is equal to the aggregate unit capacity dispatched (AgGCD) times the unit effective generating capacity (UnEGC) divided by the aggregate unit effective generating capacity (AgEGC). In other words, the model keeps track of each individual unit that makes up an aggregated unit. Once the LP has solved, it provides how much generation the aggregated unit provided. This information is used to calculate how much generation each of the individual units provided.

The actual generation of the individual units (UnEG) for that time period can then be calculated from the unit generating capacity dispatched (UnGCD) times the number of hours in the period (HDHours) divided by 1000 to convert MW to GWh.

Key equations in LP aggregation include:

**Key Equations in RunLPForDispatch**     *Aggregated available capacity for units where (UnVCost < AgVCost \* 1.02) or (UnVCost < AgVCost + 0.001)*

$$\text{AgEGC} = \text{AgEGC} + \text{UnEGC}$$

*Unit generating capacity dispatched*

$$\text{UnGCD} = \text{AgGCD} * \text{UnEGC} / \text{AgEGC}$$

Unit generation dispatched

$$\text{UnEG} = \text{UnGCD} * \text{HDHours} / 1000$$

Table 32 summarizes the variable names and definitions of the key inputs and outputs from *RunLPForDispatch*.

**Table 32: Input and Output Variables - RunLPForDispatch**

<b><i>RunLPForDispatch Inputs and Outputs</i></b>
<b>Key inputs</b>
HDADP [Node,TimeP,Month,Year] = Average Load in Interval (MW)
HDPDP [Node,TimeP,Month,Year] = Peak Load in Interval (MW)
HDEmGC [Node,Month,Year] = Emergency Power Available (MW)
HDEmVC [Node,TimeP,Month,Year] = Emergency Power Dispatch Price (\$/MWh)
LLMax [Node,NodeX] = Maximum Loading on Transmission Lines (MW)
LLVC [Node,NodeX] = Transmission Rate (\$/MWh)
UnEGC [Unit,TimeP,Month,Year] = Effective Generating Capacity (MW)
UnVCost [Unit,TimeP,Month,Year] = Bid Price of Power Offered to Spot Market (\$/MWh)
<b>Key outputs</b>
HDEmMDS [Node,TimeP,Month,Year] = Emergency Power Dispatched (MW)
EmEGA [Node,TimeP,Month,Year] = Emergency Generation (GWh)
UnGCD [Unit,TimeP,Month,Year] = Generating Capacity Dispatched (MW)
UnEG [Unit,TimeP,Month,Year] = Unit Generation (GWh)
HDLLoad [Node,NodeX,TimeP,Month,Year] = Loading on Transmission Lines (MW)
HDPrA [Node,TimeP,Month,Year] = Spot Market Marginal Price (\$/MWh)

**Function RechargingRequirements**

This Function accumulates the electricity generated by storage technologies (StorEG), such as pumped hydro, for time periods one through five (time period six represents low load hours for which storage technologies need recharging). Due to the looping nature of the submodule, the output of this Function (StorEnergy) is used in the earlier Function *IncreaseLoadsForRecharging* and increases the amount of energy demand by the amount of electricity that is required for recharging.

Key Equations for RechargingRequirements are as follows:

**Key Equations in Recharging Requirements**      Electricity generated from storage technologies:  
 $\text{StorEG} = \text{StorEG} + \text{UnEG}$ , accumulated across units with  $\text{UnStorage} \text{ equals } 1$

Electricity required to recharge storage technologies:

$StorEnergy = StorEnergy + UnEG / UnEffStorage$ , accumulated across units with  $UnStorage$  equals 1

Table 33 summarizes the variable names and definitions of the key inputs and outputs used in *RechargingRequirements*.

**Table 33: Input and Output Variables - RechargingRequirements**

<b>RechargingRequirements Inputs and Outputs</b>
<p><b>Key inputs</b>  <math>UnEffStorage</math> [Unit] = Storage Efficiency (GWH/GWH)  <math>UnEG</math> [Unit,TimeP,Month,Year] = Generation (GWH)  <math>UnStorage</math> [Unit] = Storage Switch (1=Storage Unit)</p> <p><b>Key outputs</b>  <math>StorEG</math> [Node,Month,Year] = Electricity Generated from Storage Techs (GWh/Year)  <math>StorEnergy</math> [Node,Month,Year] = Electricity Required to Recharge Storage Techs (GWh/Year)</p>

**Function PollutionForPolicy**

This Function is used for policies and calculates emissions based on generation dispatched then saves to policy variable (PolGHG).

PollutionForPolicy accumulates emissions using the equation below.

GHG Emissions from dispatched generation:

$PolGHG = PolGHG + \sum(UnEGTM * UnPOCGWh * PolConv)$ , accumulated across units

Table 34 summarizes the variable names and definitions of the Function’s key inputs and outputs.

**Table 34: Input and Output Variables - PollutionForPolicy**

<b>PollutionForPolicy Inputs and Outputs</b>
<p><b>Key inputs</b>  <math>PolConv</math> [Poll] = Pollution Conversion Factor (convert GHGs to eCO2)  <math>UnEGTM</math> [Unit,TimeP,Month,Year] = Generation (GWH)  <math>UnPOCGWh</math> [Unit,Poll,Year] = Pollution Coefficient (Tonnes/GWh)</p> <p><b>Key outputs</b>  <math>PolGHG</math> [Area,Year] = GHG Emissions from Dispatched Generation (Tonnes)</p>

**Function ConstrainAvailabilityBasedOnPollution**

This Function is used for policies in which a constraint on emissions exists. If there is a constraint, then the availability factor (PolAvFactor) is adjusted and the dispatch is re-executed. The model will re-execute generation dispatch until the GHG Emissions (PolGHG) is within the allowable margin (PolMargin) of the GHG emissions Limit (PolGHGLimit).

The key equations of ConstrainAvailabilityBasedOnPollution follow:

Availability factor from pollution constraint:

$$\text{PolAvFactor} = \text{Minimum}(\text{PolAvFactor} * (\text{PolGHGLimit} / \text{PolGHG})^{**} \text{PolIncr}, 1)$$

rerunning until either:

- a.  $\text{abs}(\text{PolGHG} - \text{PolGHGLimit}) > \text{PolMargin} * \text{PolGHGLimit}$
- b.  $\text{PolGHG} - \text{PolGHGLimit} > 0$  and  $\text{PolAvFactor} \text{ eq } 1.00$

Table 35 summarizes the variable names and definitions of the key input and output variables to the *ConstrainAvailabilityBasedOnPollution* Function.

**Table 35: Input and Output Variables - ConstrainAvailabilityBasedOnPollution**

<b><i>ConstrainAvailabilityBasedOnPollution Inputs and Outputs</i></b>
<p><b>Key inputs</b></p> <p>PolGHG [Area,Year] = GHG Emissions from Dispatched Generation (Tonnes)</p> <p>PolGHGLimit [Area,Year] = Limit on GHG Emissions (Tonnes/Yr)</p> <p>PolIncr [Area,Year] = Increment Parameter for Emission Constraint (Tonnes/Tonnes)</p> <p>PolMargin (Area) = Allowable Margin on Emissions (Tonnes/Tonnes)</p>
<p><b>Key outputs</b></p> <p>PolAvFactor (Plant,Area,Year) = Availability Factor from Pollution Constraint (MW/MW)</p>

### Function OutputsFromDispatch

Calculates outputs resulting from dispatch, such as generation totaled across different sets, unit costs, stranded investments for retired units, imports and exports.

Equations in *OutputsFromDispatch* are listed below:

1.  $\text{UnRes} = \sum(\text{UnEGTM} * \text{UnResFlag})$
2.  $\text{UUnEGA} = \sum(\text{UnEGTM})$
3.  $\text{UnEGA} = \sum(\text{UnEGTM})$ 
  - a. unless  $\text{UnEGA} = \text{XUnEGA}$
4.  $\text{EGNDA} = \text{EGNDA} + \text{UnEGA}$ , accumulated across all units
5.  $\text{EGNDAuu} = \text{EGNDAuu} + \text{UUnEGA}$ , accumulated across all units
6.  $\text{UnOUEG} = \text{UnEGGross} - \text{UnEGA}$

- 
- a.  $UnEGGross = UnEGA / (1 - UnOUREG)$
  - 7.  $UnOUGC = UnGC - UnGCNet$ 
    - a.  $UnGCNet = UnGC * (1 - UnOURGC)$
  - 8.  $EGA = \sum(EGNDA)$
  - 9.  $EGAu = \sum(EGNDAu)$
  - 10.  $UnPCF = UnEGA / (UnGC * 8760 / 1000)$
  - 11.  $UnPCFu = UUnEGA / (UnGC * 8760 / 1000)$
  - 12.  $PCF = EGA / (GCG * 8760 / 1000)$
  - 13.  $FPECC = \sum(UnFP * UnEGA * UnHRt) / \sum(UnEGA * UnHRt)$
  - 14.  $FPPCost = FPECC$
  - 15.  $SICstG = SICstG + (UnNA * CCR + UnSLDPR)$  , *accumulated across all units*
  - 16.  $AFC = AFCCum / GCCum$
  - 17.  $AVC = AVCCum / EGCum$
  - 18.  $ACE = (AFCCum + AVCCum) / EGCum$
  - 19.  $PAF = PAFCum / GCCum$ 
    - a.  $AFCCum = AFCCum + UnAFC * \max(UnGC, 0.000001)$  , *accumulated across all units*
    - b.  $GCCum = GCCum + \max(UnGC, 0.000001)$  , *accumulated across all units*
    - c.  $AVCCum = AVCCum + UnAVC * \max(UUnEGA, 0.000001)$  , *accumulated across all units*
    - d.  $EGCum = EGCum + \max(UUnEGA, 0.000001)$  , *accumulated across all units*
    - e.  $PAFCum = PAFCum + \sum(UnEGC * HDHours) / \sum(HDHours)$  , *accumulated across all units*
  - 20.  $AreaSales = \sum(HDLLoad * HDHours) / 1000$
  - 21.  $ExpSales = \sum(HDLLoad * HDHours) / 1000$
  - 22.  $AreaPurchases = \sum(HDLLoad * LLEff * HDHours) / 1000$
  - 23.  $ExpPurchases = \sum(HDLLoad * LLEff * HDHours) / 1000$

### **OutputsFromDispatch Inputs and Outputs**

#### Key inputs

CCR<sub>Plant,GenCo,Year</sub> = Capital Charge Rate (1/YR)

HDHours<sub>TimeP,Month</sub> = Number of Hours in the Interval (Hours)

HDLLOAD<sub>Node,NodeX,TimeP,Month,Year</sub> = Loading on Transmission Lines (MW)

LLEff<sub>Node,NodeX,Year</sub> = Transmission Line Efficiency (MW/MW)

UnAFC<sub>Unit,Year</sub> = Average Fixed Costs (\$/KW)

UnAVC<sub>Unit,Year</sub> = Average Variable Costs (\$/MWh)

UnEGC<sub>Unit,TimeP,Month,Year</sub> = Effective Generating Capacity (MW)

UnEGTM<sub>Unit,TimeP,Month,Year</sub> = Generation (GWH)

UnFP<sub>Unit,Year</sub> = Fuel Price (\$/mmBtu)

UnGC<sub>Unit,Year</sub> = Gross Generating Capacity (MW)

UnGC<sub>Unit,Year</sub> = Gross Generating Capacity (MW)

UnHRT<sub>Unit,Year</sub> = Heat Rate (BTU/KWh)

UnNA<sub>Unit,Year</sub> = Net Asset Value of Generating Unit (M\$)

UnOUREG<sub>Unit,Year</sub> = Own Use Rate for Generation (MW/MW)

UnOURGC<sub>Unit,Year</sub> = Own Use Rate for Generating Capacity (MW/MW)

UnResFlag<sub>Unit,TimeP,Month,Year</sub> = Flag to Indicate if Unit is Forced On to Provide Reserves (1=Yes)

UnSLDPR<sub>Unit,Year</sub> = Depreciation (M\$/Yr)

#### Key outputs

ACE<sub>Plant,GenCo,Year</sub> = Average Cost of Energy (\$/MWh)

AreaPurchases<sub>Area,Year</sub> = Purchases from Areas in the same Country (GWH/Yr)

AreaSales<sub>Area,Year</sub> = Sales to Areas in the same Country (GWH/Yr)

AVC<sub>Plant,GenCo,Year</sub> = Average Variable Costs (\$/MWh)

EGA<sub>Plant,GenCo,Year</sub> = Electricity Generated (GWh/YR)

EGAu<sub>Plant,GenCo,Year</sub> = Electricity Generated (GWh/YR)

EGNDA<sub>Plant,Node,GenCo,Year</sub> = Electricity Generation (GWh/YR)

EGNDAu<sub>Plant,Node,GenCo,Year</sub> = Electricity Generation (GWh/YR)

ExpPurchases<sub>Area,Year</sub> = Purchases from Areas in a different Country (GWH/Yr)

ExpSales<sub>Area,Year</sub> = Sales to Areas in a different Country (GWH/Yr)

FPECC<sub>Fuel,ECC,Area,Year</sub> = Fuel Prices excluding Emission Costs (\$/mmBtu)

FPPCost<sub>Fuel,ECC,Area,Year</sub> = Fuel Prices including Emission Costs (\$/mmBtu)

PAF<sub>Plant,GenCo,Year</sub> = Plant Availability Fractor (MW/MW)

PCF<sub>Plant,GenCo,Year</sub> = Plant Capacity Factor (MW/MW)

SICstG<sub>GenCo,Year</sub> = Stranded Investment Cost by GenCo (M\$/Yr)

UnEGA<sub>Unit,Year</sub> = Net Generation (GWH)

UnOUEG<sub>Unit,Year</sub> = Own Use Generation (GWh/Yr)

UnOUGC<sub>Unit,Year</sub> = Own Use Generating Capacity (MW)

UnPCF<sub>Unit,Year</sub> = Unit Capacity Factor (MW/MW)

UnPCFuu<sub>Unit,Year</sub> = Unit Capacity Factor (MW/MW)

UnRes<sub>Unit,Year</sub> = Generation while Unit is Forced On to Provide Reserves (GWH)

UUnEGA<sub>Unit,Year</sub> = Endogenous Generation (GWH)

## 8. Fuel Usage Submodule

*EFuelUsage.jl* contains the source code and Functions that make up the fuel usage submodule within the electric supply sector. The fuel usage submodule’s main Function is named *RunFuelUsage* and is called from the *SingleYear* Function inside *E2100.jl*. It is the executed following generation dispatch.

### Submodule Objective

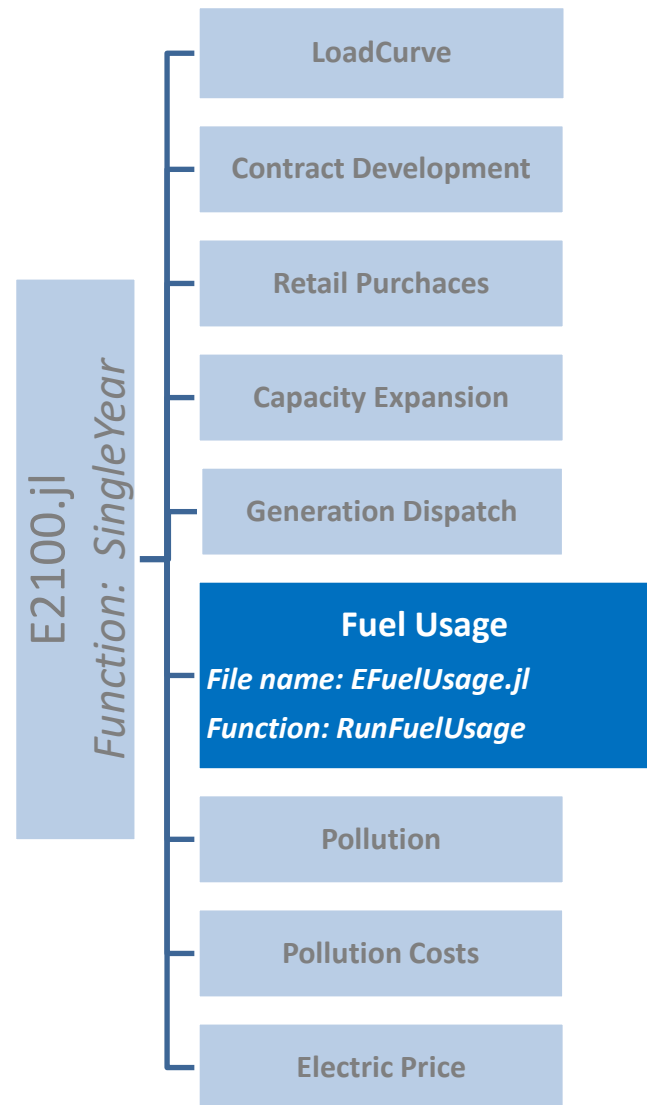
The objective of the fuel usage submodule is to aggregate and summarize various results (in particular fuel consumed by electric units) from the fuel usage submodule into forms suitable for output and use in other sections of the model.

Figure 31 shows the inputs and outputs of the Fuel Usage submodule.

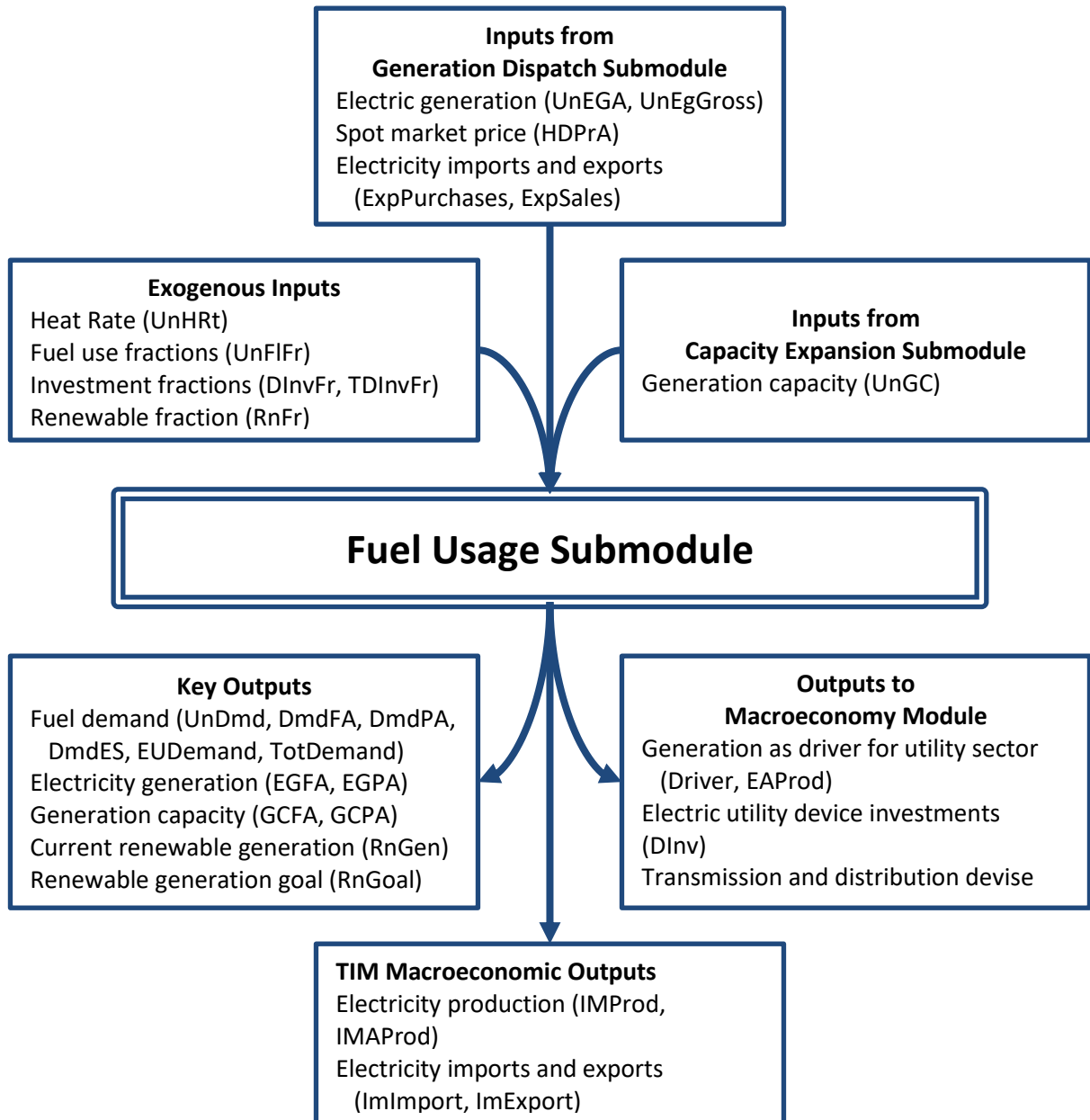
### Submodule Methodology

Creates unit fuel demands (UnDmd) and (UUnDmd) from gross generation (UnEGGross) and the heat rate (UnHRt).

Figure 30: Fuel Usage in Electric Supply



**Figure 31. Fuel Usage Submodule Diagram of Inputs and Outputs**



## Submodule Functions

The Functions used in the demand summary are found in the *EFuelUsage.jl* file and called from the *RunFuelUsage* Function.

Figure 32 provides a diagram listing each of the Functions called in *RunFuelUsage* and illustrates the order of Functions called.

**Figure 32: Function RunFuelUsage in EFuelUsage.jl**

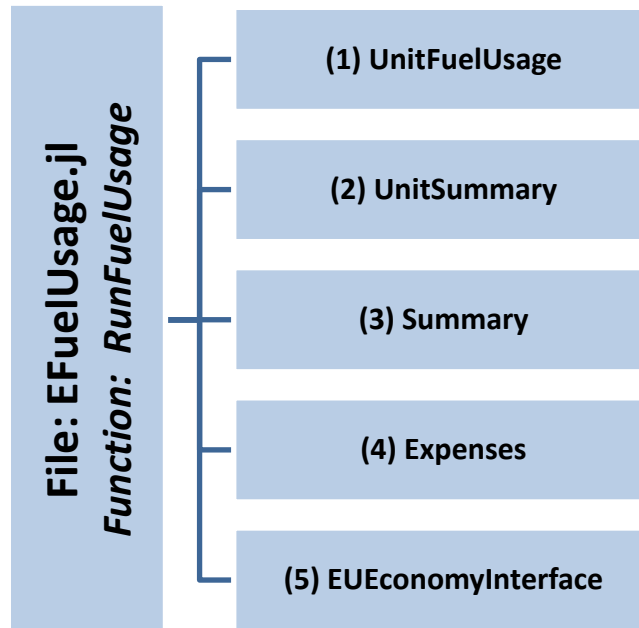


Table 36 below provides a summary of each of the Functions called within the fuel usage submodule.

**Table 36: Description of Functions within Fuel Usage**

Function Description	Key Outputs
<b>1. UnitFuelUsage</b> The Function UnitFuelUsage creates unit fuel demands (UnDmd) and (UUnDmd) from the gross generation (UnEGGross), the heat rate (UnHRt), and the fuel fraction (UnFIFr).	Unit fuel demands UnDmd and UUnDmd

<p><b>2. UnitSummary</b> Accumulates Demand, Generation, and Capacity by fuel and plant type.</p>	<p>DmdFA, DmdPA, EGFA, EGPA, GCFA, GCPA.</p>
<p><b>3. Summary</b> Aggregates Renewable Generation (RnGen) and, depending on BugSw, Renewable Goal (RnGoal), and Demand (DmdFA) into demands for the utility generation sector (DmdES, EuDemand, TotDemand).</p>	<p>RnGen, RnGoal, DmdES, EUDemand, TotDemand</p>
<p><b>4. Expenses</b> Calculates Device Investments, Transmission and Distribution Investments.</p>	<p>DInv, TDIInv</p>
<p><b>5. EUEconomyInterface</b> Exports Driver (Driver) to the rest of the Economy segments of the model.</p>	<p>Driver</p>

**Function UnitFuelUsage**

The Function UnitFuelUsage creates UnDMD and UUnDmd from the Gross Generation (UnEGGross) the heat rate (UnHRt), and the Unit Fuel Fraction (UnFIFr).

UnDmd is used in emissions related formulas in EPollution.jl [and various text files]. UUnDmd is where the model-generated endogenous demand saved in the cases when unit demand is exogenously specified, and is output to serve as a potential check on the model.

<b>UnitFuelUsage Inputs and Outputs</b>
<p><b>Key inputs</b>                      UnEGGross<sub>Unit,Year</sub> = Gross Generation (GWh/Yr)                      UnFIFr<sub>Unit,FuelEP,Year</sub> = Fuel Fraction (BTU/BTU)                      UnHRt<sub>Unit,Year</sub> = Heat Rate (BTU/KWh)</p>
<p><b>Key outputs</b>                      UnDmd<sub>Unit,FuelEP,Year</sub> = Energy Demand (TBtu)                      UUnDmd<sub>Unit,FuelEP,Year</sub> = Endogenous Energy Demand (TBtu)</p>

Key equations in UnitFuelUsage are as follows:

**KeyEquation**  
**UnitFuelUsage**

Unit energy demands  

$$UnDmd = \max(UnEGGross * UnHRt / 1e6 * UnFIFr, 0)$$

## Function UnitSummary

This Function accumulates unit-level Generation (UnEGA) and Capacity (UnGC) into area-level Generation (EGFA, EGPA) and Capacity (GCFA, GCPA) by fuel and plant type, respectively. Demand (DmdFA, DmdPA) is calculated from unit-level Generation (UnEGA), Heat Rate (UnHRt), and Fuel Fraction (UnFIFr).

Fuel-type Demand (DmdFA) is used in emissions calculations in EPollution. Plant-type

Generation (EGPA) is used in various calculations within the Fuel Use submodule. All six variables (DmdFA, DmdPA, EGFA, EGPA, GCFA, GCPA) are widely used as model output.

Key equations in UnitSummary are all accumulated across multiple units either by fuel or by plant type, with each unit adding a particular component to the total:

### UnitSummary Inputs and Outputs

#### Key inputs

UnDmd<sub>Unit,FuelEP,Year</sub> = Energy Demand (TBtu)

UnEGA<sub>Unit,Year</sub> = Generation (GWH)

UnGC<sub>Unit,Year</sub> = Generation Capacity (MW)

UnFIFr<sub>Unit,FuelEP,Year</sub> = Fuel Fraction (BTU/BTU)

UnHRt<sub>Unit,Year</sub> = Heat Rate (BTU/KWh)

#### Key outputs

DmdFA<sub>Fuel,Area,Year</sub> = Energy Demand (TBtu/Yr)

DmdPA<sub>Plant,Area,Year</sub> = Energy Demand (TBtu/Yr)

EGFA<sub>Fuel,Area,Year</sub> = Electricity Generation (GWh/Yr)

EGPA<sub>Plant,Area,Year</sub> = Electricity Generation (GWh/Yr)

GCFA<sub>Fuel,Area,Year</sub> = Generation Capacity (GWh/Yr)

GCPA<sub>Plant,Area,Year</sub> = Generation Capacity (GWh/Yr)

#### KeyEquations UnitSummary

1.  $DmdFA = DmdFA + UnDmd$ , *accumulated across fuels*
  - a.  $DmdFA = DmdFA + UnEGA * UnHRt / 1e6$ , *for units which do not burn emissions-producing fuels*
2.  $DmdPA = DmdPA + \sum_{FuelEP} UnDmd$ , *accumulated across plant types*
  - a.  $DmdPA = DmdPA + UnEGA * UnHRt / 1e6$ , *for units which do not burn emissions-producing fuels*
3.  $EGFA = EGFA + UnEGA * UnFIFr$ , *accumulated across fuels*
4.  $EGPA = EGPA + UnEGA$ , *accumulated across plant types*
5.  $GCFA = GCFA + UnGC$ , *accumulated across fuels*
6.  $GCPA = GCPA + UnGC$ , *accumulated across plant types*

## Function Summary

This Function aggregates Plant-type Generation (EGPA) Renewable Generation (RnGen). Renewable Goal (RnGoal) is aggregated from Sales (SAEC) and the Renewable Fraction (RnFr). The Function also aggregates Demand (DmdFA) into various demands for the utility generation sector (DmdES, EuDemand, TotDemand).

### Renewable Generation

(RnGen) and Renewable Goal (RnGoal) are used in outputs, EPollution.jl equations, and Capacity Expansion (with the caveat that RnGoal is separately calculated within ECapacityExpansion.jl). The various forms of Demand (DmdES, EUDemand, TotDemand) are used in the Supply Sector and as model output.

Key equations in Summary are as follows:

- KeyEquations Summary**
1.  $RnGen = \sum_{Plant} EGPA$
  2.  $RnGoal = (\sum_{ECC} SaEC) * RnFrArea$
  3.  $DmdES = DmdFA$
  4.  $EuDemand = DmdFA$
  5.  $TotDemand = DmdFA$

### Summary Inputs and Outputs

#### Key inputs

$DmdFA_{Fuel,Area,Year} = \text{Energy Demand (TBtu/Yr)}$

$EGPA_{Plant,Area,Year} = \text{Electricity Generation (GWh/Yr)}$

$RnFr_{Area,Year} = \text{Renewable Fraction (BTU/BTU)}$

$SAEC_{ECC,Area,Year} = \text{Sales (GWh/Yr)}$

#### Key outputs

$RnGen_{Area,Year} = \text{Renewable Current Level of Generation (GWh/Yr)}$

$RnGoal_{Area,Year} = \text{Renewable Generation Goal (GWh/Yr)}$

$DmdES_{Fuel,Area,Year} = \text{Energy Demands (TBtu/Yr)}$

$EUDemand_{Fuel,ECC,Area,Year} = \text{Energy Demands (TBtu/Yr)}$

$TotDemand_{Fuel,ECC,Area,Year} = \text{Energy Demands (TBtu/Yr)}$

These investments (DInv, TInv) are aggregated with other industries in the Supply sector for input to the Informatrica Model.

<b><i>Expenses Inputs and Outputs</i></b>
<p>Key inputs</p> <p><math>GCPA_{Plant,Area,Year}</math> = Generation Capacity (MW)</p> <p><math>DInvFr_{Plant,Area,Year}</math> = Device Investments Fraction (\$/KW/Yr)</p> <p><math>TInvFr_{Plant,Area,Year}</math> = Electric Transmission and Distribution Investments Fraction (\$/KW/Yr)</p> <p>Key outputs</p> <p><math>TInv_{ECC,Area,Year}</math> = Device Investments (M\$/Yr)</p> <p><math>TInv_{Area,Year}</math> = Electric Transmission and Distribution Investments (M\$/Yr)</p>

Key equations in Expenses are as follows:

<b>KeyEquations</b> <b>Expenses</b>	1. $DInv = \sum_{Plant} (GCPA * DInvFr) * Infla * 1000 / 1000000$
	2. $TInv = \sum_{Plant} (GCPA * TInvFr) * Infla * 1000 / 1000000$

### Function EUEconomyInterface

This Function calculates two variables from the plant-type Generation (EGPA). First is the Economic Driver (Driver) for the Utility Generation Economic Category and is used in the Macroeconomic sector of the model. The second is the Electric Utility Production (EAProd) which is a copy of EGPA.

<b><i>EUEconomyInterface Inputs and Outputs</i></b>
<p>Key inputs</p> <p><math>EGPA_{Plant,Area,Year}</math> = Electricity Generation (GWh/Yr)</p> <p>Key outputs</p> <p><math>Driver_{ECC,Area,Year}</math> = Economic Driver (Various Millions/Yr)</p> <p><math>EAProd_{Plant,Area,Year}</math> = Electric Utility Production (GWh/Yr)</p>

Key equations in EUEconomyInterface are as follows:

<b>KeyEquations</b> <b>EuEconomy</b> <b>Interface</b>	1. $Driver = \sum_{Plant} EGPA$
	2. $EAProd = EGPA$

## 9. Pollution Submodule

Figure 33. Pollution in Electric Supply

*EPollution.jl* contains the source code and Functions that make up the pollution submodule. The pollution submodule's main Function is named *Part2* and is called from the *SingleYear* Function inside *E2100.jl*. It is called after directly after the Fuel Usage Submodule has been executed.

### Submodule Objective

The objective of the pollution submodule is twofold. First, the submodule calculates the emissions from electricity generation on the unit level. Second, the submodule applies various pollution related policies. These include but are not limited to offsets, emissions-related capital retirements and construction, Alberta CASA policies, and the assignment of gratis permits.

### Submodule Functions

The main Functions called from the *Part2* Function in *EPollution.jl* are shown in Figure 34 below.

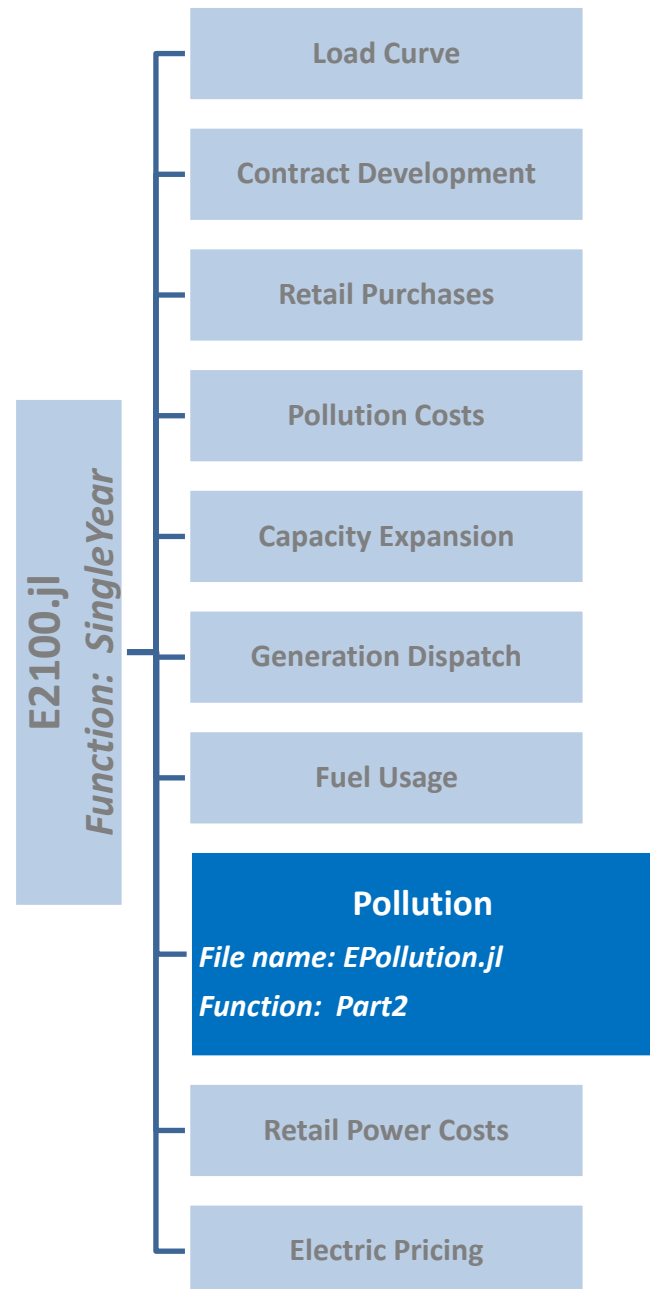
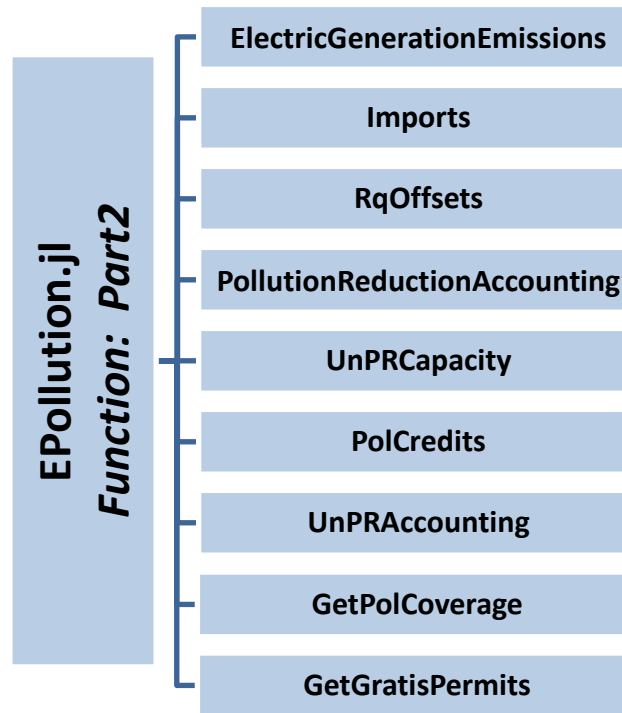


Figure 34. Function Part2 in EPollution.jl



The following Table 37 provides a summary of each of the Functions within the Pollution Submodule (Part 2). It also lists the key outputs from each of those Functions.

Table 37. Description of Functions within Pollution

Function Description	Key Outputs
<b>1. ElectricGenerationEmissions</b> Energy usage and generation are multiplied with emissions coefficients to produce emission inventories	Utility emissions (UnPol, MEPol)
<b>2. Imports</b> Calculation of emission inventories for imported electricity	PolImports
<b>3. RqOffsets</b> Set the number of offsets required for unit if necessary	OffRq
<b>4. PollutionReductionAccounting</b> Simulates the initiation, construction, and retirement of capital to reduce emissions at the sector level	Spending on reduction capital (PRExp)

<p><b>5. UnPRCapacity</b> Set capacity for emission reductions at the unit level using emission targets</p>	Unit reduction capacity (UnRCR)
<p><b>6. PolCredits</b> Simulates emission credit system for Alberta CASA policies</p>	Unit reduction costs and capacity (UnRCC2, UnRCR2)
<p><b>7. UnPrAccounting</b> Simulates the initiation, construction, and retirement of capital to reduce emissions at the unit level</p>	Spending on reduction capital (PRExp)
<p><b>8. GetPolCoverage</b> Calculate emissions covered under regulatory policies</p>	Covered pollution (PolCov)
<p><b>9. GetGratisPermits</b> Determine the number of gratis permits available to be deducted from the covered emissions</p>	Gratis permits (PGratis)

**Function ElectricGenerationEmissions**

This Function utilizes demand, generation, and emissions coefficients to create the inventory of energy-based and process emissions produced from electric generation. Emission inventories are the total amount of emissions produced for a given unit per year. These values are calculated in the model by

<i><b>ElectricGenerationEmissions Inputs and Outputs</b></i>
<p><b>Key Inputs</b>                      UnDmd<sub>Unit,FuelEP,Year</sub> = Energy Demands (TBtu)                      UnEGA<sub>Unit,Year</sub> = Generation (GWH)                      UnPOCA<sub>Unit,FuelEP,Poll,Year</sub> = Average Pollution Coefficient (Tonnes/TBtu)                      UnSqFr<sub>Unit,Poll,Year</sub> = Sequestered Pollution Fraction (Tonne/Tonne)</p>
<p><b>Key outputs</b>                      CgPolUnit<sub>Fuel,ECC,Poll,Area,Year</sub> = Pollution from Industrial Units (Tonnes/Yr)                      GrossPol<sub>ECC,Poll,Area,Year</sub> = Gross Pollution - before any policies (Tonnes/Yr)                      MEPol<sub>ECC,Poll,Area,Year</sub> = Process Pollution (Tonnes/Year)                      UnMEPol<sub>Unit,Poll,Year</sub> = Process Pollution (Tonnes)                      UnPol<sub>Unit,FuelEP,Poll,Year</sub> = Pollution (Tonnes)</p>

multiplying the amount of energy consumed by the marginal rate of emissions produced from consuming energy for each electric unit. For sector level and cogeneration

inventories, the inventories produced by each unit are agglomerated and saved into a final total value.

Key equations in *ElectricGenerationEmissions* are as follows:

**KeyEquations**  
*ElectricGeneration*  
*Emissions*

1.  $UnPol = UnDmd * UnPOCA (1 - UnSqFr)$
2.  $UnMEPol = \max(UnEGA * UnMECX, 0)$
3.  $MEPol = \sum UnMEPol$
4.  $CgPolUnit = CgPolUnit + UnPol$ , *accumulated across Units*
5.  $GrossPol = \sum (EUPol / EURM) * ECoverage + MEPol * ECoverage$

### Function Imports

Emission inventories for electricity purchased from a different area are calculated for use in the total emissions inventory. Some emissions policies also account for “imported emissions” due to higher emission electric generation from an area outside the area covered by the policy.

#### **Imports Inputs and Outputs**

##### Key Inputs

$UnEGA_{Unit,Year} = \text{Generation (GWH)}$

$UnFrImports_{Unit,Area,Year} = \text{Fraction of Unit Imported to Area (GWH/GWH)}$

##### Key outputs

$PollImports_{Poll,Area,Year} = \text{Emissions from Imported Electricity (Tonnes)}$

The Key equation in *Imports* follows:

**KeyEquations**  
*Imports*

1.  $PollImports = PolSplImport + PolOthImports + PolRnImports$ , *where*
  - a.  $PolSplImport = \sum (UnPol * UnFrImports)$
  - b.  $PolOthImports = \sum (GrImports * POCXOthImports) + XPolOthImports$ ,  
*where in sequence*
    - i.  $GrImports = \sum (HDLLoad * HDHours * NdArFr) / 1000$
    - ii.  $GrImpTot = \sum (GrImports)$
    - iii.  $GrImpAdj = \max(GrImpTot - EGSplImports - RnImports, 0) / GrImpTot$
    - iv.  $GrImports = GrImports * GrImpAdj * GrImpMult$
  - c.  $PolRnImports = RnImports * POCXRnImports$ 
    - i.  $RnImports = \max(RnGoal - RnGen, 0)$

### Function RqOffsets

If a unit is instructed to offset its emissions (**OffSw**) then the amount of offsets required is set based on the unit's emissions.

The key equation in RqOffsets is as follows:

**KeyEquations**  
*RqOffsets*

$$\text{OffRq} = \sum(\text{UnPol} * \text{PolConv})$$

#### **RqOffsets Inputs and Outputs**

##### Key Inputs

$\text{OffSw}_{\text{Area,Year}}$  = GHG Electric Utility Offsets Required Switch (1=Required)

$\text{PolConv}_{\text{Poll}}$  = Pollution Conversion Factor (convert GHGs to eCO<sub>2</sub>)

$\text{UnPol}_{\text{Unit,FuelEP,Poll,Year}}$  = Pollution (Tonnes)

##### Key outputs

$\text{OffRq}_{\text{Area,Year}}$  = GHG Electric Utility Offsets Required (Tonnes/Yr)

### Function PollutionReductionAccounting

The total amount spend on emission reduction in the electric generation sector is calculated based on the marginal cost of reduction capital and the amount of capital needed per year. Altering the marginal emissions coefficient in ENERGY 2100 is simulated by developing reduction capital, which would include reduction devices such as smokestack scrubbers. For each given year this Function brings online reduction capital that has finished construction, retires old capital that has reached the end of its lifespan, and initiates new construction if necessary to meet the reduction amount required. The total amount spend on construction by the private sector is saved after accounting for any government subsidies.

#### **PollutionReductionAccounting Inputs and Outputs**

##### Key Inputs

$\text{AGFr}_{\text{ECC,Poll,Area,Year}}$  = Government Subsidy (\$/\$)

$\text{EURCAP}_{\text{FuelEP,Plant,Poll,Area,Year}}$  = Reduction Capacity (Tonnes/Year)

$\text{EURCCEm}_{\text{FuelEP,Plant,Poll,Area,Year}}$  = Embedded Reduction Capital Cost (\$/Tonnes)

$\text{EURCD}_{\text{Poll}}$  = Reduction Capital Construction Delay (Years)

$\text{EURCPL}_{\text{Poll}}$  = Reduction Capital Physical Life (Years)

$\text{EURCR}_{\text{FuelEP,Plant,Poll,Area,Year}}$  = Reduction Capital Completion Rate (Tonnes/Year/Year)

$\text{EURM}_{\text{FuelEP,Plant,Poll,Area,Year}}$  = Reduction Multiplier by Area (Tonnes/Tonnes)

$\text{EUROCF}_{\text{FuelEP,Plant,Poll,Area,Year}}$  = Pollution Reduction O&M Cost Factor (\$/\$)

$\text{EURP}_{\text{FuelEP,Plant,Poll,Area,Year}}$  = Pollutant Reduction (Tonnes/Tonnes)

$\text{PAdCost}_{\text{ECC,Poll,Area,Year}}$  = Policy Administrative Cost (\$/Year)

##### Key outputs

$\text{GReExp}_{\text{ECC,Poll,Area,Year}}$  = Reduction Government Expenses (\$/Year)

$\text{PRExp}_{\text{ECC,Poll,Area,Year}}$  = Reduction Private Expenses (M\$/Yr)

Key equations in *PollutionReductionAccounting* are as follows:

**KeyEquations**  
*PollutionReduction*  
*Accounting*

1.  $GRExp = \sum (EURCR * EURCC * AGFr) / 1e6 + PAdCost$
2.  $PRExp = \sum (EURCR * EURCC * (1 - AGFr) + EURCCEm * EUROCF) / 1e6$

### Function UnPRCapacity

The Function begins by calculating and vintaging the capacity of reequipment used in reducing emissions. This Function produces the amount of emissions needed to be reduced by each unit and construction of reduction capital required to meet the goal. The Indicated Reduction Capacity (UnRICap) is constructed by removing the effects of pollution reduction for the unit (UnRM) from its actual pollution (UnPol), but reapplying the required reduction (UnRP) to find out how many tonnes of pollution could be reduced. This is combined with the Pollution Reduction Capacity (UnRCap) and Pollution Reduction Completion Rate (UnRCR) from the previous year to create the Capacity Initiation Rate (UnRCI). The Reduction Capacity and Reduction Completion (UnRCap, UnRCR) rate are then updated and lagged based on the above variables.

#### **UnPRCapacity Inputs and Outputs**

##### Key Inputs

EURCD<sub>Poll</sub> = Reduction Capital Construction Delay (Years)

EURCPL<sub>Poll</sub> = Reduction Capital Physical Life (Years)

UnPol<sub>Unit, FuelEP, Poll, Year</sub> = Pollution (Tonnes)

UnRM<sub>Unit, FuelEP, Poll, Year</sub> = Pollution Reduction Multiplier (Tonnes/Tonnes)

UnRP<sub>Unit, FuelEP, Poll, Year</sub> = Pollution Reduction Requirement (Tonnes/Tonnes)

##### Key outputs

UnRCR<sub>Unit, FuelEP, Poll, Year</sub> = Pollution Reduction Completion Rate (Tonnes/Yr/Yr)

The key equations in UnPrCapacity are mostly grounded in updating variables from the previous year.

**KeyEquations**  
*PollutionReduction*  
*Accounting*

1.  $UnRICap = UnPol / UnRM * UnRP$
2.  $UnRCI = \text{xmax}( 0, UnRICap - UnRCap - UnRCR * EURCD + UnRCap / EURCPL ) / EURCD$
3.  $UnRCap = UnRCap + DT * (UnRCR - UnRCap / EURCPL)$
4.  $UnRCR = UnRCR + DT * (UnRCI - UnRCR) / EURCD$

### Function PolCredits

The Function *PolCredits* was designed to specifically simulate Alberta CASA regulations. This code only applies to units in Alberta if they are covered by the policy as set by the policy file during the model run. For each covered unit, emission credits are purchased starting with the largest units first. If credits are unavailable, then the unit either retires or constructs reduction capital.

<b><i>PolCredits Inputs and Outputs</i></b>
<b>Key Inputs</b>
UnPol <sub>Unit,FuelEP,Poll,Year</sub> = Pollution (Tonnes)
PoCredits <sub>Unit,Poll,Year</sub> = Pollution Credits Accumulated (Tonnes)
<b>Key outputs</b>
PoCreditsNeeded <sub>Unit,Poll,Year</sub> = Pollution Credits Needed (Tonne/Yr)
UnPRCost <sub>Unit,Year</sub> = Levelized Cost of Unit with Pollution Reduction Equipment (\$/MWh)
UnRCC2 <sub>Unit,Poll,Year</sub> = Pollution Reduction Capital Cost (\$/Tonne)
UnRCR2 <sub>Unit, Poll,Year</sub> = Pollution Reduction Completion Rate (Tonnes/Yr/Yr)

### Function UnPRAccounting

This Function controls the vintaging process for emissions reduction equipment at the unit level. It initiates, completes, and retires emission reduction capital and the amount invested each year is added to the total private and government pollution reduction expenditures.

Key equations in UnPrAccounting are as follows. Some are adjustments to previous variables.

<b><i>UnPRAccounting Inputs and Outputs</i></b>
<b>Key Inputs</b>
UnRCC <sub>Unit,FuelEP,Poll,Year</sub> = Pollution Reduction Capital Cost (\$/Tonne)
UnRCC2 <sub>Unit,Poll,Year</sub> = Pollution Reduction Capital Cost (\$/Tonne)
UnRCCEm <sub>Unit,FuelEP,Poll,Year</sub> = Embedded Pollution Reduction Capital Cost (\$/Tonne)
UnROCF <sub>Unit,Poll,Year</sub> = Pollution Reduction O&M Cost Factor (\$/\$)
UnRCR <sub>Unit,FuelEP,Poll,Year</sub> = Pollution Reduction Completion Rate (Tonnes/Yr/Yr)
PoCredits <sub>Unit,Poll,Year</sub> = Pollution Credits Accumulated (Tonnes)
<b>Key outputs</b>
GRExp <sub>ECC,Poll,Area,Year</sub> = Reduction Government Expenses (\$/Year)
PRExp <sub>ECC,Poll,Area,Year</sub> = Reduction Private Expenses (M\$/Yr)
UnRCGA <sub>Unit,Year</sub> = Emission Reduction Capital Costs (M\$)
UnRCOM <sub>Unit,Year</sub> = Emission Reduction O&M Costs (M\$)

**KeyEquations**  
*PollutionReduction Accounting*

1.  $GRExp = GRExp + (\sum(UnRCR * UnRCC) + UnRCR2 * UnRCC2) * AGFr / 1e6 + PAdCost$
2.  $PRExp = PRExp + [(\sum(UnRCR * UnRCC) + UnRCR2 * UnRCC2) * (1 - AGFr) + UnRCCEm * UnROCF] / 1e6$

3.  $UnRCGA = \sum_{Poll} ((\sum_{Fuel}(UnRCR*UnRCC)+ UnRCR2*UnRCC2)*(1-AGFr))/1e6$
4.  $UnRCOM = \sum(UnRCCEm*UnROCF)/1e6$

### Function GetPolCoverage

Some emissions policies only apply to part of an industry, or not every facility within an industry. As such we may only be interested in the pollution from those facilities which are covered by the policy. This Function simulates this by calculating the emissions covered by regulation as the total quantity of emissions produced in the area covered

#### GetPolCoverage Inputs and Outputs

##### Key Inputs

$PolConv_{Poll}$  = Pollution Conversion Factor (convert GHGs to eCO2)

$PollImports_{Poll,Area,Year}$  = Emissions from Imported Electricity (Tonnes)

$UnCoverage_{Unit,Poll,Year}$  = Fraction of Unit Covered in Emission Market (1=100% Covered)

$UnPol_{Unit,FuelEP,Poll,Year}$  = Pollution (Tonnes)

##### Key outputs

$PolCov_{ECC,Poll,PCov,Area,Year}$  = Covered Pollution (Tonnes/Yr)

by a policy, plus the emissions from electricity imports into that area. This value is saved as an export of the electric sector for use in the broader pollution sector of the model.

The key equation for GetPolCoverage is as follows; it is accumulated across units within covered areas.

1.  $PolCov = PolCov + \sum_{Fuel} (UnPol) * PolConv + PollImports * PolConv * ECoverage$ , accumulated across units in covered areas and active markets

### Function GetGratisPermits

The number of gratis permits is set depending on the type of gratis program. For units covered by an intensity target (FacSw), the value of the permits is accounted for. Otherwise, gratis permits for covered units are either calculated or set to an exogenous value.

#### GetGratisPermits Inputs and Outputs

##### Key Inputs

$ETAvPr_{Market,Year}$  = Average Cost of Emission Trading Allowances (\$/Tonnes)

$FacSw_{Market}$  = Facility Level Intensity Target Switch (1=Facility Target)'

##### Key outputs

$PGratis_{ECC,Poll,PCov,Area,Year}$  = Gratis Permits (Tonnes/Year)

$UnPGratis_{Unit,Poll,Year}$  = Gratis Permits (Tonnes/Yr)

$UnPGValue_{Unit,Poll,Year}$  = Gratis Permit Value (\$M/Yr)

---

The unit gratis permits (UnPGratis) are given their values by applicable policies, and many of the relevant calculations are done partly outside the model within the associated txp files. This allows for greater speed and more flexible modeling, where multiple different policies can be included or excluded at the modeler's discretion.

The key equations for GetGratisPermits follow below:

1.  $\text{UnPGValue} = \text{UnPGratis} * \text{PolConv} * \text{ETA}vPr / 1e6$
2.  $\text{PGratis} = \text{PGratEx} + \text{PGratNew}$ 
  - a.  $\text{PGratEx} = \text{PGratEx} + \text{UnPGratis}$ , *accumulated across online units*
  - b.  $\text{PGratNew} = \text{PGratNew} + \text{UnPGratis}$ , *accumulated across units not yet online*

## 10. Retail Power Costs Submodule

Figure 35. Retail Power Costs in Electric Supply

*ERetailPowerCosts.jl* contains the source code and Functions that make up the Retail Power Costs Submodule. Its main Function is named *ElectricCosts* and is called from the *SingleYear* Function inside *E2100.jl*. At the point when the Retail Power Costs Submodule gets called, the load curves have been created and retail contracts with generating companies have been developed and purchased.

### Submodule Objective

The purpose of the Retail Power Costs Submodule is to use the contract and wholesale market parameters and sales to determine both the total spent by each retail company on power purchases and the cost per unit of purchased power. The latter value is generated as a key input for calculating the final retail electric price.

Figure 36Figure 39 illustrates the key inputs and outputs of the electric price submodule.

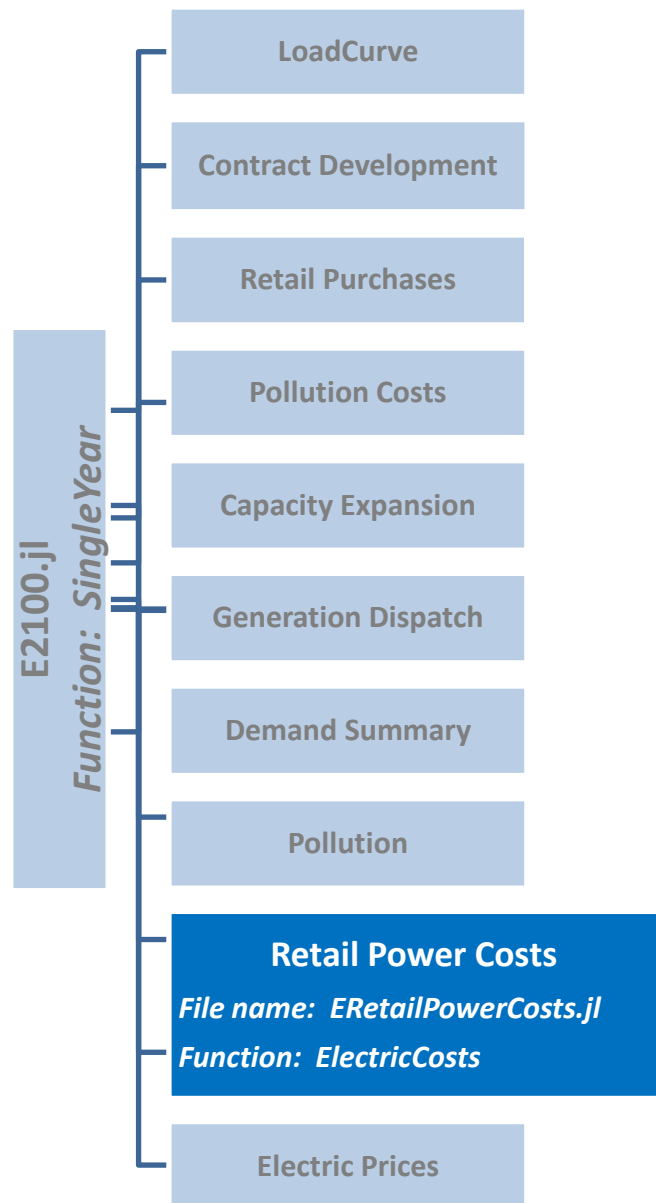
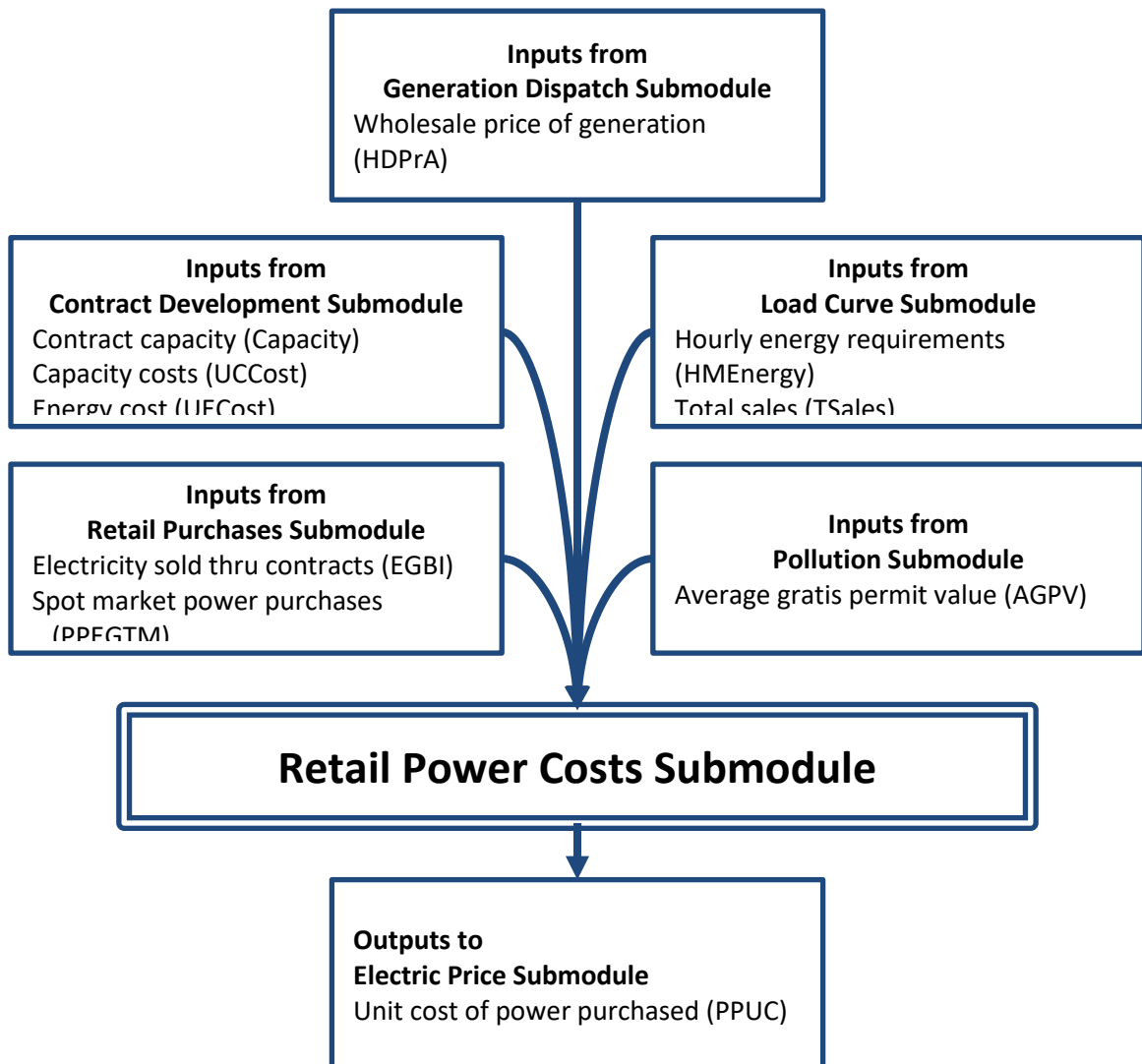


Figure 36. Retail Power Costs Submodule Diagram of Inputs and Outputs



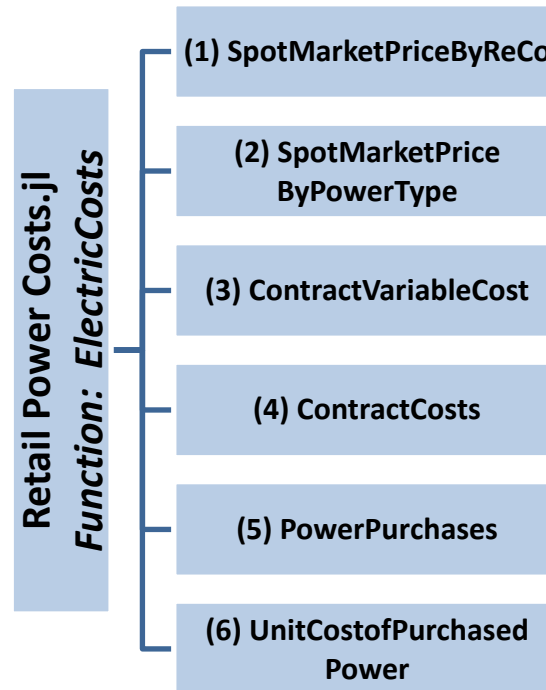
### Submodule Methodology

With costs (UVCost, UCCost, PPVCost) and the amount of power purchased from contracts (EGBI) and the spot market (PPEGTM) determined, ERetailPowerCosts calculates the final cost of purchasing power (PPUC) from the generating company for each retail company.

### Submodule Functions

Below, Figure 37 lists the Functions that make up the Retail Power Purchases Submodule (*ElectricCosts* Function).

**Figure 37. Function ElectricCosts in ERetailPowerCosts.jl**



Each of these Functions is summarized in Table 38 below and the key outputs from each are listed as well. A more detailed description of the Functions is contained in the sections that follow.

**Table 38. Description of Functions within Retail Power Costs**

Function Description	Key Outputs
<b>1. SpotMarketPriceByReCo</b> The retail price of spot market power is a function of the node price weighted by the retail demands at each node	Spot market price per retail company (HMPr, HMPrA)
<b>2. SpotMarketPriceByPowerType</b> The spot market price is saved to the appropriate power type (PPSet = 1) for later use in the model	Price by power type (PPVCost)

Function Description	Key Outputs
<b>3. ContractVariableCosts</b> Energy costs are revised by removing gratis permits if applicable	Energy costs (UVCost)
<b>4. ContractCosts</b> Calculate the total variable and contract costs for each retail company	Contract variable and fixed costs (DVCost, DCCost)
<b>5. PowerPurchases</b> Calculate the total purchases from the spot market from the spot market price and total amount purchased	Total retail spending on power purchases (PUCT)
<b>6. UnitCostOfPurchasedPower</b> The unit cost of purchased power is total amount spent divided by total sales	Spending per unit of power purchased (PPUC)

**Function SpotMarketPriceByReCo**

The calculated price of purchased power for each node from the electric generation dispatch is used to set the price for each retail company to purchasing off the wholesale market, weighted by the percentage of power purchased from each node by the retail company.

<b>SpotMarketPriceByReCo Inputs and Outputs</b>
Key Inputs HDPrA <sub>Node,TimeP,Month,Year</sub> = Spot Market Marginal Price HMEnergy <sub>Node,Area,TimeP,Month</sub> = Energy in Interval (GWH)
Key outputs HMPPr <sub>Area,TimeP,Month,Year</sub> = Spot Market Price (\$/MWh) HMPPrA <sub>Area,Year</sub> = Average Spot Market Price (\$/MWh)

Key Equations  
 SpotMarketPrice  
 ByReCo

Spot market price:  

$$HMPPr = \sum(HDPrA * HMEnergy) / \sum(HMEnergy)$$

### Function SpotMarketPriceByPowerType

The spot market price is saved to the appropriate power type (PPSet = 1) for later use in the model.

Key Equations  
SpotMarketPrice  
ByPowerType

Estimated spot market price:

$$PPVCOST = \frac{\sum(PPEGTM * HMPR)}{\sum(PPEGTM)}$$

#### SpotMarketPriceByPowerType Inputs and Outputs

Key Inputs

HMPR<sub>Area,TimeP,Month,Year</sub> = Spot Market Price (\$/MWh)

PPEGTM<sub>Area,TimeP,Month,Year</sub> = 'Power Purchases (GWH)

Key outputs

PPVCOST<sub>PPSet,Month,Area,Year</sub> = Estimated Spot Market Price (\$/MWh)

### Function ContractVariableCost

The variable cost of the contract is calculated by removing the impact of any gratis permits on the contract energy costs.

Gratis permits are regulatory permits that are freely given to the generating company instead of being purchased. The cost of all

permits is incorporated in the energy cost variable, so if the permits are determined to be gratis then the impact of the permit purchase cost is removed from the variable cost of the contract. The switch to pass costs is set to 1 by default, meaning that gratis permits are refunded unless specified otherwise by the user.

#### ContractVariableCost Inputs and Outputs

Key Inputs

UECost<sub>Area,GenCo,Plant,Year</sub> = Contract Energy Cost (MILLS/kWh)

AGPVS<sub>GenCo</sub> = Is Value of Gratis Permit passed on to Customers (1=yes)

AGPV<sub>Plant,GenCo</sub> = Average Gratis Permit Value (\$/Yr)

Key outputs

UVCost<sub>Area,GenCo,Plant,Year</sub> = Contract Variable Cost (\$/mWh)

Key Equations  
PassGratisTo  
Customers

Contract variable cost:

$$UVCost = UECost - AGPV$$

**Function ContractCosts**

Total contract costs are determined for each retail company by multiplying the amount of capacity or contract sales by the relevant cost per unit. The total variable cost for each retail company is the sum of the variable cost of each contract

(UVCost) times the amount of contract power purchased (EGBI).

The total contract fixed costs (UCCost) is the sum of contact capacity (Capacity) times the contract capacity costs (UCCost).

The amount of purchased power from contracts (EGBI) is an input from the retail purchases submodule. The fixed cost of contracts (UCCost) is an input from the contract development submodule.

<b>ContractCosts Inputs and Outputs</b>
<p><b>Key Inputs</b></p> <p>Capacity<sub>Area,GenCo,Plant,TimeP,Month,Year</sub> = Capacity under Contract (MW)</p> <p>EGBI<sub>Area,GenCo,Plant,Year</sub> = Electricity sold thru Contracts (GWh/Yr)</p> <p>UVCost<sub>Area,GenCo,Plant,Year</sub> = Contract Variable Cost (\$/MWh)</p> <p>UCCost<sub>Area,GenCo,Plant,Year</sub> = Contract Capacity Cost (\$/KW)</p>
<p><b>Key outputs</b></p> <p>DVCost<sub>Area,Year</sub> = Variable Cost (M\$/Yr)</p> <p>DCCost<sub>Area,Year</sub> = Capacity Cost (M\$/Yr)</p>

**Key Equation:**  
**ContractCosts**      Contract variable/energy cost:  

$$DVCost = \sum(EGBI * UVCost) / 1000$$

Contract fixed/capacity cost:  

$$DCCOST = \sum (max(Capacity) * UCCost) / 1000$$

**Function PowerPurchases**

The total cost of purchases from the spot market (**PPCT**) is the sum of the energy purchased from the spot market (**PPEGTM**) times the cost of spot market power (**PPVCost**).

With both the total cost spend on contracts and

<b>PowerPurchases Inputs and Outputs</b>
<p><b>Key Inputs</b></p> <p>PPEGTM<sub>Area,TimeP,Month</sub> = Power Purchases (GWH)</p> <p>PPVCOST<sub>PPSet,Month,Area</sub> = Estimated Spot Market Price (\$/MWh)</p> <p>DVCOST<sub>Area</sub> = Variable Cost (M\$/YR)</p> <p>DCCOST<sub>Area</sub> = Capacity Cost (M\$/YR)</p>
<p><b>Key outputs</b></p> <p>PPCT<sub>PPSet,Area</sub> = Cost of Purchase Power (M\$/YR)</p> <p>PUCT<sub>Area</sub> = Cost of Purchase Power (M\$/YR)</p>

the wholesale market determined, the model can now calculate the final cost of purchasing power for each retail company. Total cost of purchasing power (**PUCT**) is the sum of contract costs (**DVCost, DCCost**) plus the spot market purchases (**PPCT**).

The purchased power set is defined by: BasePP, BaseSpot, IntPP, IntSpot, PeakPP, PeakSpot, EmgPP

**Key Equation:** Cost of purchase power by purchased power set and retail company:  
**PowerPurchases**  $PPCT = \sum(PPEGTM * PPVCOST) / 1000$

Cost of purchase power by retail company:  
 $PUCT = PUCTBI + PUCTSM$   
 Where  $PUCTBI = DVCOST + DCCOST$

**Function UnitCostofPurchasedPower**

The cost of electricity per unit sold (**PPUC**) is the total amount spend on purchases (**PUCT**) divided by the amount of electricity sold (**TSales**)

**Key Equation:** Unit cost of purchased power:  
**UnitCostofPurchasedPower**  $PPUC = PUCT / TSales * 1000$

<b>UnitCostofPurchasedPower Inputs and Outputs</b>
Key Inputs
$PUCT_{Area} = \text{Cost of Purchase Power (M\$/YR)}$
$TSales_{Area} = \text{Electricity Sales (GWh/YR)}$
Key outputs
$PPUC_{Area} = \text{'Unit Cost of Purchase Power (\$/MWH)'$

## 11. Electric Price Submodule

The *ElectricPrice.jl* file contains the source code and Functions that make up the electric price submodule. Its main Function is named *Finance* and it is called from the *SingleYear* Function inside *E2100.jl*. This electric price Function is the last Function that gets called in the electric supply module.

### Submodule Objective

The purpose of the electric price submodule is to calculate the final price of electricity (PECalc, PE) for each retail company based on the unit cost (PPUC), delivery charge (PEDC) and other adjustments.

The generation dispatch submodule of ENERGY 2100 determines the price of producing a unit of electricity. This differs from the final price that is passed to the customer purchasing power since it doesn't include the cost of electric transmission, fuel taxes, and a number of other factors that influence the final retail price.

Simulating the final retail price is key as an input for the decision-making process of consumers in the demand sector.

Figure 39 illustrates the key inputs and outputs of the electric price submodule.

Figure 38: Electric Price in Electric Supply

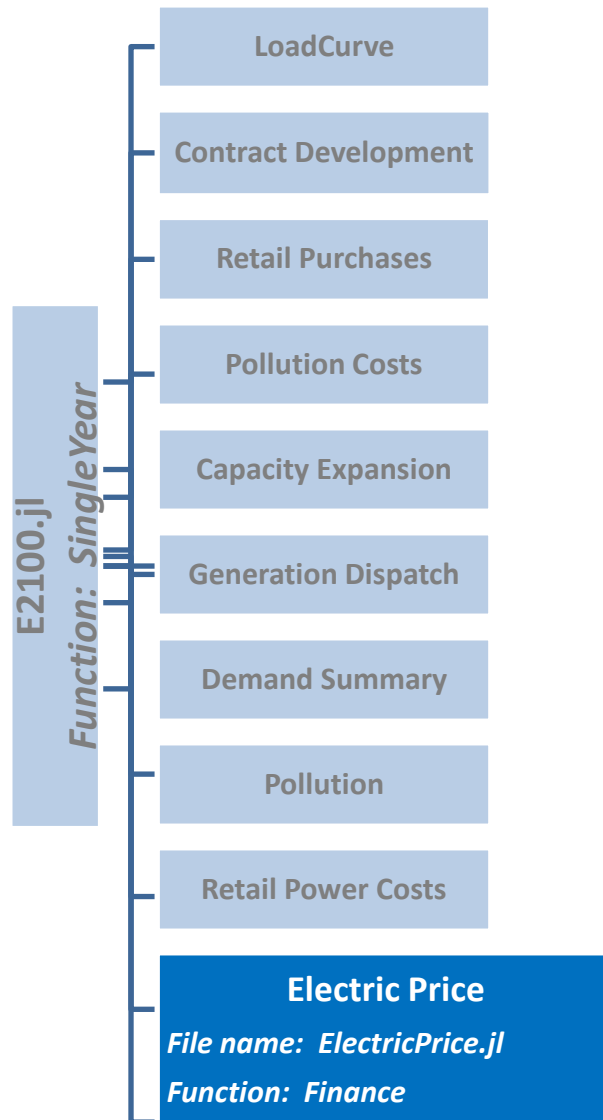
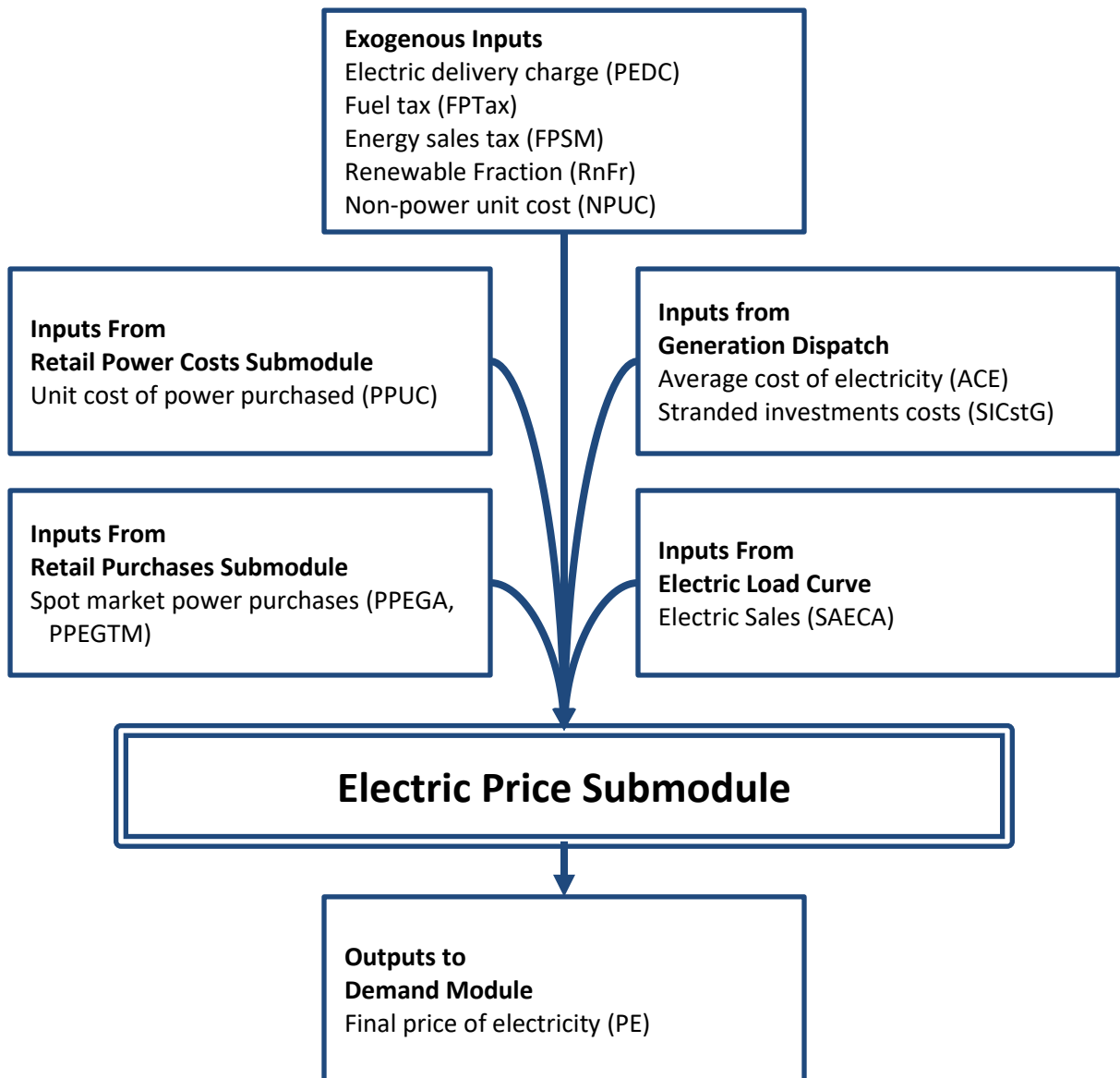


Figure 39. Electric Price Submodule Diagram of Inputs and Outputs



## Submodule Functions

The Functions that make up the electric price submodule are shown in the figure below. These are all housed in ElectricPrice.jl and called from the *Finance* Function.

**Figure 40: Function Finance in ElectricPrice.jl**

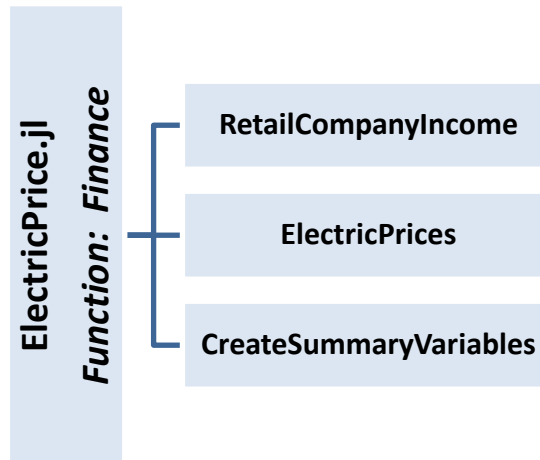


Table 39 provides a brief description of each of electric price Functions and identifies the key output variables. More detailed descriptions of these Functions are provided in the sections following the table.

**Table 39. Description of Functions within Electric Price**

Function Description	Key Outputs
<p><b>1. RetailCompanyIncome</b> Determines total revenue for each retail company and saves the current retail price of electricity by economic class.</p>	Revenue (RV, Rev); Price of electricity by class (PEClass)

Function Description	Key Outputs
<p><b>2. ElectricPrices</b></p> <p>RECSPrice: Calculates impact of any renewable energy credits on retail price</p> <p>NonPowerCosts: Calculates impacts of any non-power costs on the retail price</p> <p>StrandedInvestments: Saves unit-level stranded investments for final price calculation</p> <p>ElectricPriceByECC: Determines final retail electric price as a function of price of power, delivery charges, and adjustments</p>	<p>Renewable impact on price (RnPE), Non-power costs (NPAC), Stranded investments (SICstPE), Calculated price of electricity (PECalc)</p>
<p><b>3. CreateSummaryVariables</b></p> <p>Aggregates and saves various retail company variables for use in the rest of the model and in model outputs</p>	<p>Sum contract costs (DACost), unit contract costs (UCConts)</p>

### Function RetailCompanyIncome

Revenue for each retail company is calculated by multiplying sales by the electric price. This Function saves total revenue for each retail company and the average electric price by economic 'class', which refers to broad economic categories such as Residential, Commercial, etc.

**Key Equation:  
RetailCompany  
Income**

Electricity revenues:

$$ECCRV = SaEC * PE / 1000$$

Price of electricity by class:

$$PEClass = RV / SaCL * 1000$$

<b>RetailCompanyIncome Inputs and Outputs</b>
<p><b>Key Inputs</b></p> <p><math>PE_{ECC,Area,Year}</math> = Marketer Price of Electricity (\$/MWh)</p> <p><math>SaCL_{Class,Area,Year}</math> = Electricity Sales (GWh/Yr)</p> <p><math>SaEC_{ECC,Area,Year}</math> = Electricity Sales (GWh/Yr)</p>
<p><b>Key outputs</b></p> <p><math>ECCRV[ECC,Area,Year]</math> = Electricity Revenues (M\$/Yr)</p> <p><math>Rev_{Area,Year}</math> = Revenue (M\$/Yr)</p>

## Function ElectricPrices

The *ElectricPrices* Function calculates the endogenous price of electricity for use throughout the rest of the model. The price of electricity is a function of the retail unit cost of purchasing electricity (PPUC), the delivery charge estimated during calibration of historical values (PEDC), taxes, and several adjustments dependent on the model parameters. These adjustments are calculated in sub-Functions before the final price estimation.

### **ElectricPrices Inputs and Outputs**

#### Key Inputs

EEConv = Electric Energy Conversion (Btu/kWh)

PPUC<sub>Area</sub> = Unit Cost of Purchase Power (\$/MWh)

PEDC<sub>ECC,Area,Year</sub> = Electric Delivery Charge (\$/MWh)

FPTax<sub>Prices,Area,Year</sub> = Fuel Tax (\$/mmBtu)

FPSM<sub>Prices,Area,Year</sub> = Energy Sales Tax (\$/\$)

#### Key outputs

PECalc<sub>ECC,Area,Year</sub> = Calculated Price of Electricity (\$/MWh)

The delivery charge (PEDC) is calculated as part of the electric sector calibration via a comparison between the historically input values for electric prices versus the endogenous value. This value also contains other factors that aren't explicitly contained in other input variables for the final retail price, such as utility profit

**RefundGratisToCustomers:** If the user specifies that the value of gratis permits are granted to electric customers (GRefSwitch eq 1), then the cost of the emissions credit that is incorporated in the electric purchase price is removed from the final retail price. Gratis permits are permits that are granted freely to firms as part of an emissions permit regulatory system. This is in contrast to fixed price or auctioned permits, where a company would be required to purchase the permits required for emitting. If the simulation contains a permit regulatory system, then ENERGY 2100 adds the cost of purchasing credits to every kilowatt-hour produced. If the user chooses to have the benefit of these gratis permits returned to the customer then the impact of the permits is subtracted back out when calculating the final retail price.

**RECSPrice:** The price impact of renewable energy credits is derived from the renewable requirement policy variable (RnFr), average costs of power for renewable units (ACE), and renewable electric generation (EGAGenCo). In response to a renewable portfolio standard, a retail company can choose to purchase electricity from renewable sources or purchase renewable credits. Activating the switch for this section of code (RECSwitch eq 1) assumes the latter and calculates the cost of the credit based on the renewable requirement (RnFr) to add to the final retail price.

**NonPowerCosts:** Other non-power costs are calculated from electric sales times the non-power unit cost variables (NPUC). This allows the user to specify the costs of producing any non-power goods and services that can be included in the final retail prices. Non-power costs can include overhead costs, human resources, and ancillary services produced by the utility. The unit cost variable is an exogenous input to the model that is set to zero as a default value.

**StrandedInvestments:** The unit level cost of stranded investments (SICstPE) is based on the stranded investment fraction (SICstFr) times the stranded costs per generating company (SICstG). Stranded investments refer to the amount of old debt that the company still needs to service from retired capital expenditures. Traditionally utilities often operated as monopolies with no competition, allowing for investments to assume a long depreciation schedule. Since there was no competition, a firm could plan on keeping a plant indefinitely as there was no reason to invest in new, more efficient generation. The onset of deregulation and regulation resulted in the need to replace old plants in order to produce power competitively. As a result, plants were put out of service before their initial outlay was fully paid meaning the investment is 'stranded', or not producing any power for the firm to sell. The cost of servicing this debt is not captured in the cost of purchasing power, so it is calculated separately and added to the final retail price.

The equation used to calculate the final retail price of electricity is listed below.

**Key Equation:**  
**ElectricPrices**

Calculated electricity price:  

$$PE_{Calc} = (PPUC + NPAC + RnPE + SICstPE - GPURef + PEDC) * Infla + \sum(FPTax * EEConv / 1000 * SaECA) / SaEC * Infla * (1 + \sum(FPSM * SaECA) / SaEC)$$

Model electricity price:  
 Do If ElecPrSw ne Exogenous  
     PE = PE<sub>Calc</sub>  
 Else  
     PE = XPE \* Infla

---

## 12. Electricity Supply Sector Calibration

---

In the electric supply sector the model calibrates electric generation, electric loads (peak, average, and minimum loads), and electricity prices. The calibration changes the values of the electric unit operational outage rate (UnOOR), the load shape factors (LSF through BaseAdj, HPkM, and CLSF), and the electric price differential charge (PEDC). For the operational outage rates the model iteratively executes the electricity dispatch, and then adjusts the outage rates to bring the calculated generation closer to the historical generation. The adjustment of the load shape factors is a straightforward calculation to enable the sum of the enduse demands time their load shape to equal the historical system load values. The electric price differential charge is the difference between the endogenously calculated electricity prices and the historical electricity prices. The future values for the electric unit operational outage rates are dependent on the plant type of the unit:

1. Nuclear, Renewable, and Must Run units are initially set equal to the last value. These units are not difficult to calibrate and often have a near zero operational outage rate with their generation being essentially specified with their outage rate (UnOR).
2. Peak Hydro generation is specified by the energy availability factor (UnEAF) so their operational outage rate is generally zero. A non-zero value generally means the energy availability factor is not specified correctly.
3. Peaking and Combined Cycle Oil and Gas units are set to zero in the forecast period. These units often run for a limited number of hours in the historical period, but they need to be available in the forecast to run if needed.
4. Coal units are the most difficult to specify for the forecast period. Coal units are limited in number of coal units in Canada, but are very important to GHG and CAC emissions, so each unit is carefully reviewed with the outage set in consultation with the provincial electric utilities.