Description and a first application of the TEMOA energy system model:

<u>T</u>ools for <u>E</u>nergy <u>M</u>odel <u>O</u>ptimization and <u>A</u>nalysis

International Energy Workshop 2011
Stanford University, July 7th

Joe DeCarolis, Kevin Hunter, Sarat Sreepathi
Department of Civil, Construction, and Environmental Engineering
North Carolina State University

jdecarolis@ncsu.edu

The TEMOA Project

Tools for Energy Model Optimization and Analysis

Goal: Create a community-driven, technology explicit, energy economy model

Our Approach:

- Open source code (GNU Public License)
- Open source data (GNU Public License)
- No commercial software dependencies
- Input and output data managed directly with a relational DB
- Data and code stored in a web accessible electronic repository
- A version control system
- Programming environment with links to linear, mixed integer, and non-linear solvers
- **Design for sensitivity and uncertainty analysis**
- **Utilize multi-core and compute cluster environments**

Version Control with Subversion

We are using a version control system called Subversion (SVN)
http://subversion.apache.org/
http://svnbook.red-bean.com/

Why? Ensure the integrity, sustainability and traceability of changes during the entire software lifecycle.

SVN enables:

- Multiple developers to work simultaneously on software components; automatic integration of non-conflicting changes
- Display the modifications to model source code
- Create software snapshots (releases) that represent well-tested and clearly defined milestones
- Public access to snapshots of the code and data

You can view our code online: http://svn.temoaproject.org/trac/browser
Most current branch: branches/energysystem-process

Works on all major (Unix, Windows, MacOS) platforms

COmmon Optimization Python Repository (COOPR)

- COOPR is a collection of Python optimization-related packages that supports a diverse set of optimization capabilities for formulating and analyzing optimization models.
- Algebraic model formulation using Python Optimization Modeling Objects (Pyomo)
- Capability to formulate linear, mixed integer, and non-linear model formulations without commercial solvers
- **Part of a rich Python ecosystem**

Developed by the Discrete Math and Complex Systems Department at Sandia National Laboratories: https://software.sandia.gov/trac/coopr/

TEMOA Model Features

- Flexible time slicing by season and time-of-day
- Variable length model time periods
- Technology vintaging
- Separate technology loan periods and lifetimes
- Global and technology-specific discount rates
- All commodity flows balanced at the timeslice level
- Capacity determined by commodity flows at the timeslice level

Commodity Balance

$\forall p$ in time_period

$\forall s$ in season

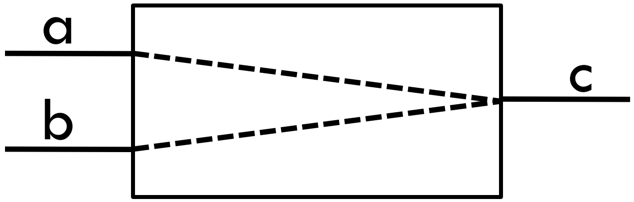
$\forall$d in time_of_day

$\forall c\_i$ in commodity

$\forall t$ in technology

$\forall v$ in vintage

$\forall c\_o$ in commodity



ProcessBalanceConstraint_rule

```
V_FlowOut(p, s, d, c_i, t, v, c_o)
≤
V_FlowIn(p, s, d, c_i, t, v, c_o)
*Efficiency(c_in, t, v, c_o)
```

Activity and Capacity Derived from V_FlowOut

ActivityConstraint_rule

$$\textbf{Activity}(p, s, d, t, v) = \sum_{c\_i} \sum_{c\_o} \texttt{V\_FlowOut}(p, s, d, c\_i, t, v, c\_o)$$

CapacityConstraint_rule

$$\texttt{Activity}(p, s, d, t, v) \le \textbf{Capacity}(t, v) * \texttt{CapacityFactor}(t, v) * \texttt{SegFrac}(s, d) * \texttt{Capacity2Activity}(t)$$

Other Key Constraints

DemandConstraint_rule

$$\sum_{c\_i}\sum_{t}\sum_{v} \texttt{V\_FlowOut(p, s, d, c\_i, t, v, c\_o)}$$

$$\geq$$

$$\texttt{Demand(p, s, d, c\_o)}$$

CommodityBalanceConstraint_rule

$$\sum_{c\_i}\sum_{t}\sum_{v} \texttt{V\_FlowOut(p, s, d, c\_i, t, v, c\_o)}$$

$$\geq$$

$$\sum_{c\_o}\sum_{t}\sum_{v} \texttt{V\_FlowIn(p, s, d, c\_i, t, v, c\_o)}$$

TEMOA Objective Function

$$
\begin{aligned}
&\texttt{Cost} = \\
&\Big(\sum_{p}\sum_{t}\sum_{v} \\
&\texttt{V\_Capacity[t, v]} \\
&\texttt{* (CostInvest[t, v]} \\
&\texttt{*\ \ \ LoanAnnualize[t, v]} \\
&\texttt{+ CostFixed[p, t, v])} \\
&+ \sum_{s}\sum_{d} \\
&\texttt{V\_Activity[p, s, d, t, v]} \\
&\texttt{* CostMarginal[p, t, v]}\ \Big)\ \texttt{* df[p]}
\end{aligned}
$$

MARKAL ‘Utopia’ System Diagram

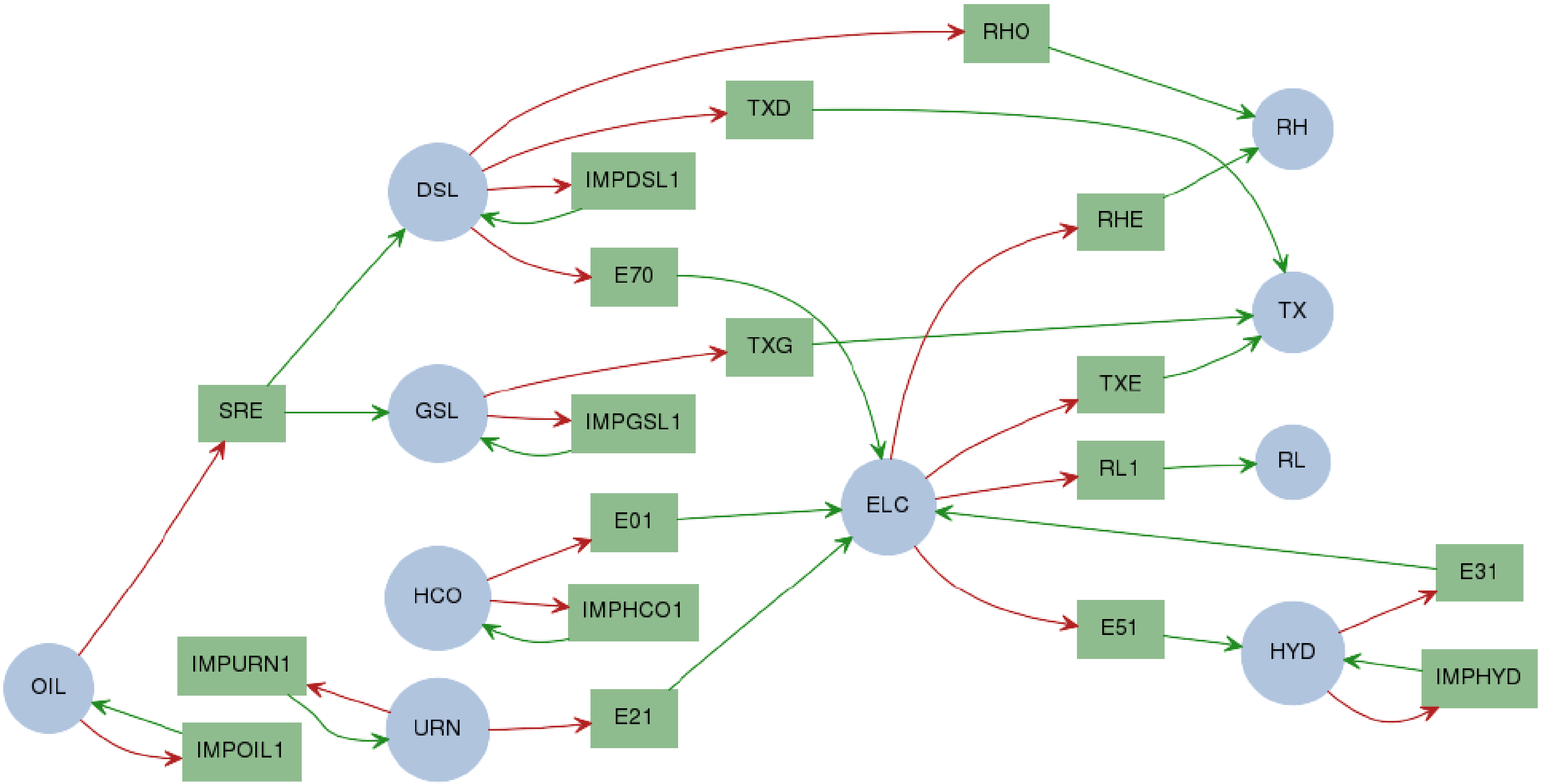
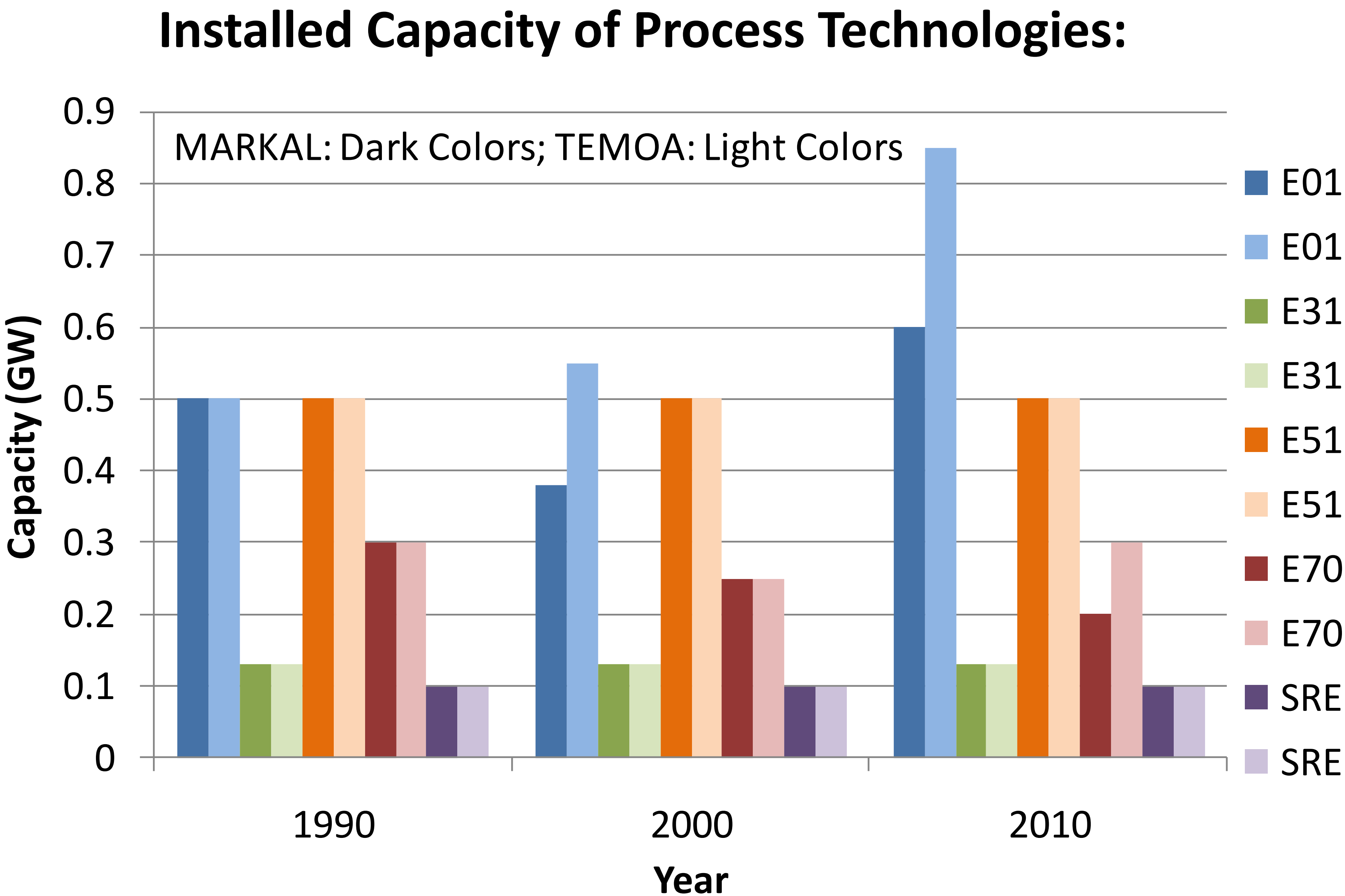


Diagram generated using Graphviz: http://www.graphviz.org/

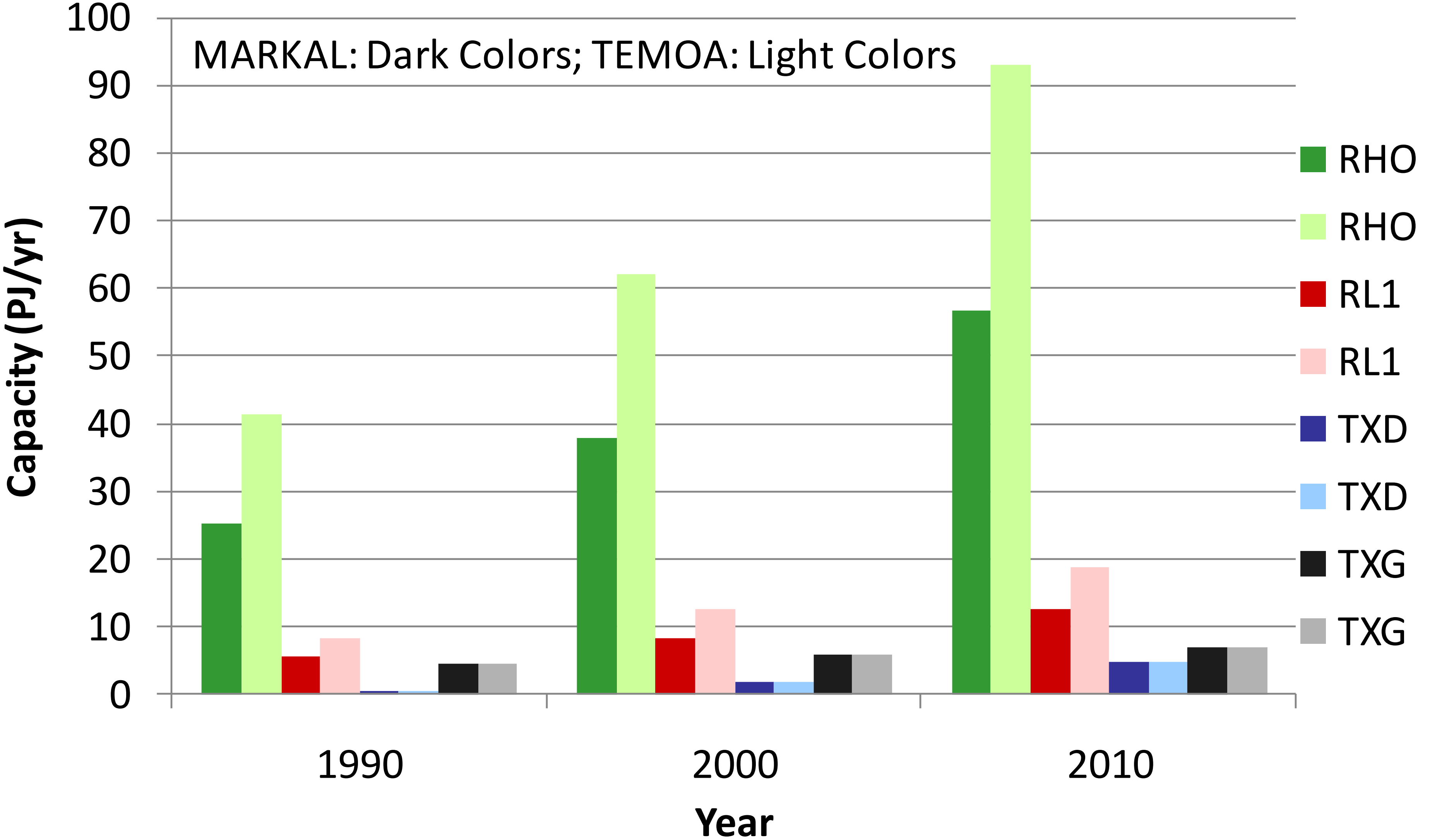
Calibration to Utopia

MARKAL Objective value: 36,821
TEMOA Objective value: 38,502



Calibration to Utopia (continued)

Installed Capacity of Demand Technologies:



Stochastic Optimization

Decision-makers need to make choices before uncertainty is resolved → requires an “act then learn” approach

Need to make short-term choices that hedge against future risk

→ Sequential decision-making process that allows recourse

Stochastic optimization

- Build a scenario tree
- Assign subjective probabilities to future outcomes
- Optimize over all possibilities

Stochastic Optimization of Energy Models

Desirable features for energy models:

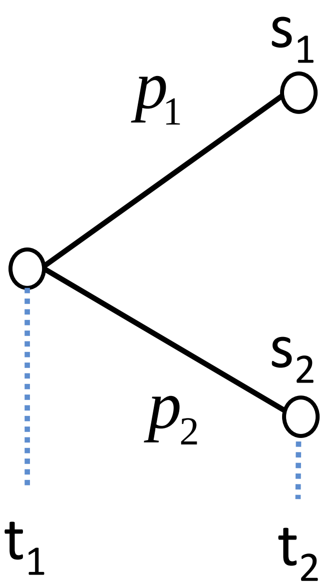
- Multi-stage (greater than 2)
- Multi-objective (e.g., cost, risk, emissions)
- Mixed integer (esp. endogenous tech learning)

Potential stochastic parameters:

- Fuel prices (esp. crude oil, natural gas, coal)
- Policy targets (e.g., CO_2 constraints, subsidies)
- Technology performance (e.g., capital cost, thermal eff)
- End-use demand projections (e.g., heating, cooling)

Simple Example of Stochastic Optimization

Suppose we have two technologies, A and B. Let *x* and *y* represent the installed capacity in Stages 1 and 2, respectively.



Stage 1 Decision Variables:

$$x_A, x_B$$

Stage 2 Decision Variables:

$$y_{A,s_1}, y_{B,s_1}, y_{A,s_2}, y_{B,s_2}$$

$$\text{Minimize: } c^T x + \sum_{s=1}^{N} p_s \cdot d_s^T \cdot y_s$$

Subject To:

$$Ax = b$$

$$T_s x + W_s y_s = h_s \quad for \;\; s = 1, \ldots, N$$

$$x \geq 0$$

$$y_s \geq 0 \qquad for \;\; s = 1, \ldots, N$$

Stochastic Optimization with PySP

Python-based Stochastic Programming (PySP) is part of the COOPR package.

To perform stochastic optimization, specify a Pyomo reference model and a scenario tree

PySP offers two options:

1. **`runef`**: builds and solves the extensive form of the model.
   "Curse of dimensionality" → memory problems
2. **`runph`**: builds and solves using a scenario-based decomposition solver (i.e., "Progressive Hedging) based on Rockafellar and Wets (1991).
   Can be implemented in a computer cluster environment; more complex scenario trees possible.

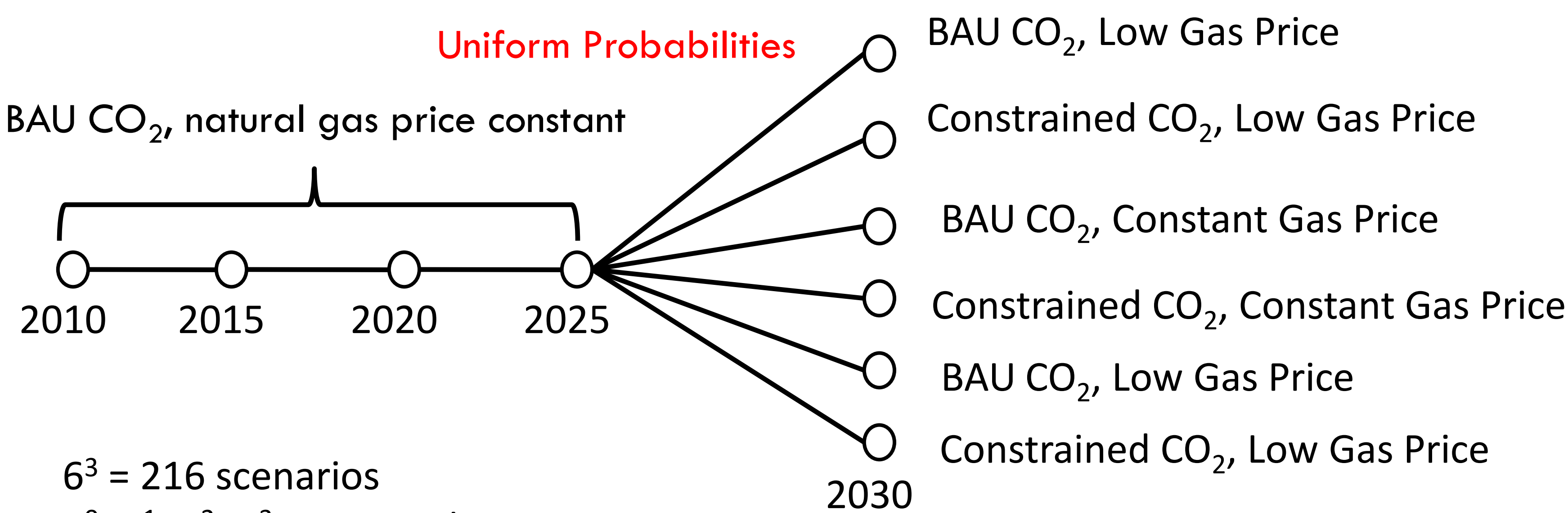
R.T. Rockafellar and R. J-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, pages 119–147, 1991.

A Test Case of the US Electric Sector

Time periods: 2010-2040, 5-year increments
2030 and after, 2 possible CO_2 emissions levels, 3 possible natural gas prices

<u>Electric sector CO_2 emissions in 2010: 2340 MmtCO_2</u>
BAU CO_2: 0.6% annual increase, **CO_2 Constrained**: 4.7% annual decrease
[-50% to +20% change in CO_2 emissions in 2040 relative to 2010]

<u>Natural gas prices in 2010: 4.45 $/GJ</u>
Low: 1.1% annual decrease, **Constant**, **High**: 8.4% annual increase
[Price ranges from 3.8 to 15 $GJ in 2040]



6^3 = 216 scenarios
$6^0+6^1+6^2+6^3$ = 259 nodes

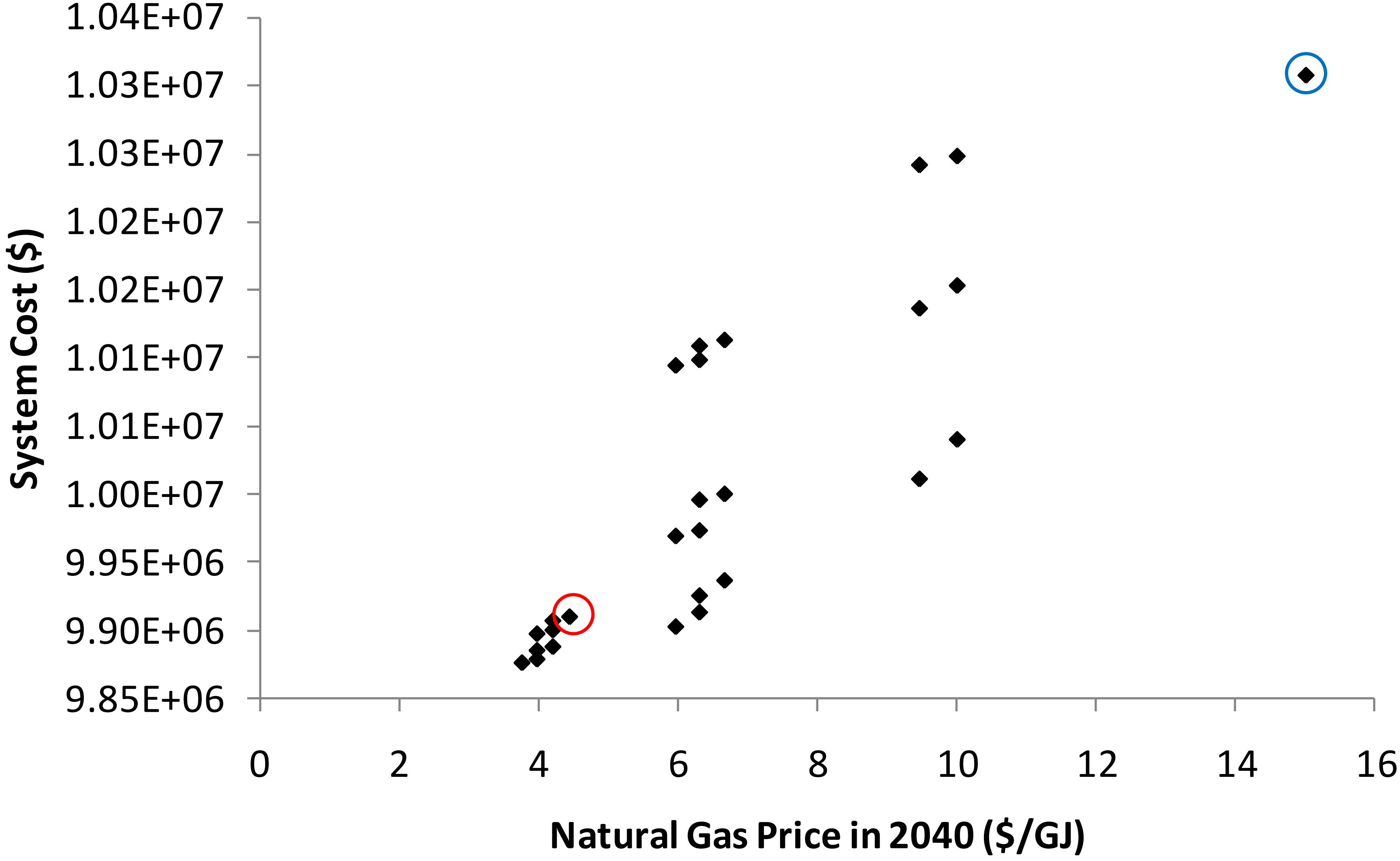
Technology Cost and Performance Characteristics

Annual growth in electricity demand of 0.6% based on the reference case in the *Annual Energy Outlook 2009*.

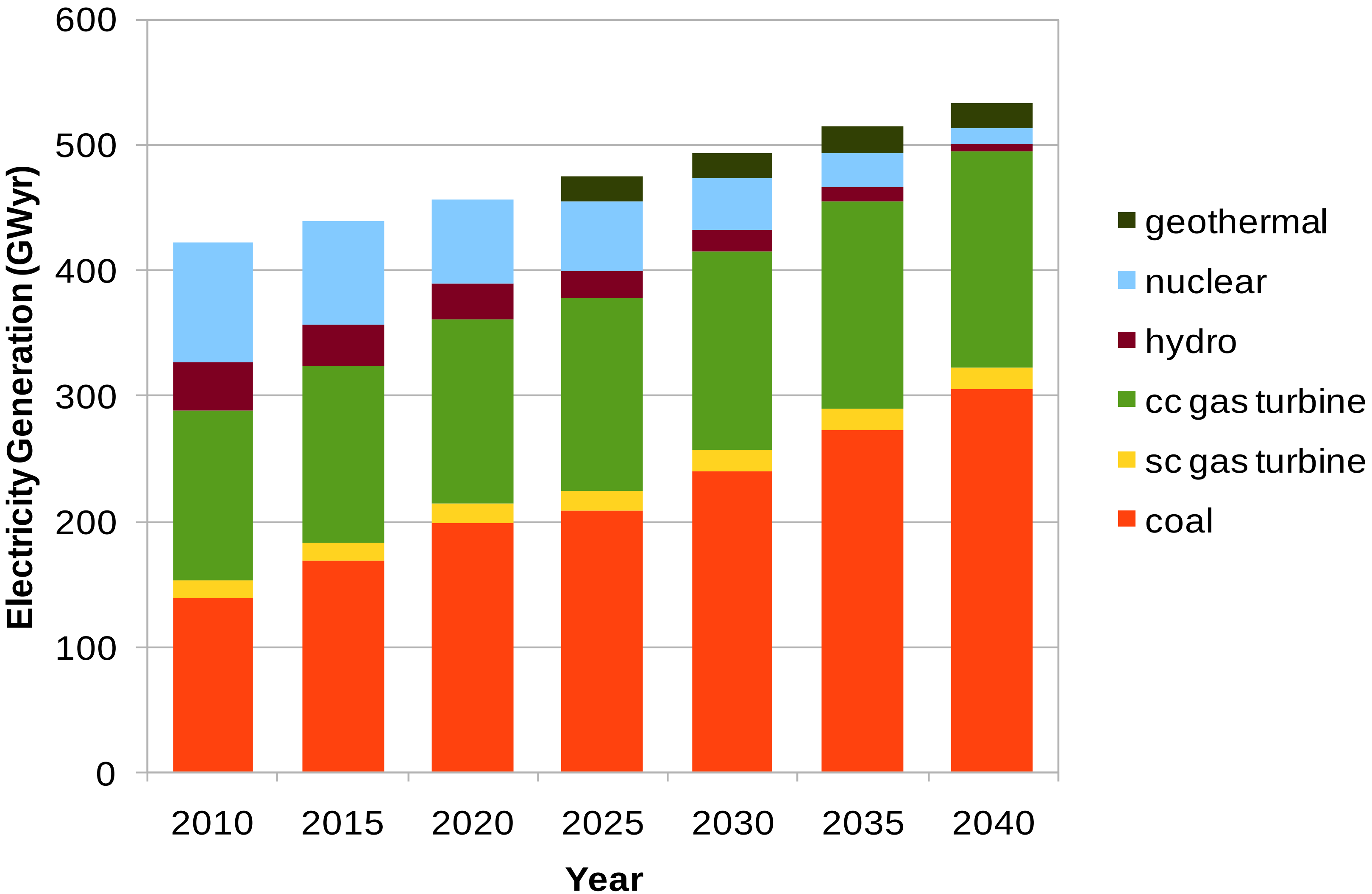
Technology[a]	Capital Cost ($/kW)	Fixed O&M ($/kW·yr)	Variable O&M ($/kWh)	Efficiency (%)	Capacity Factor (%)	Average Cost ($/kWh)	Baseload / Shoulder / Peak (B/S/P)	Capacity Constraint[b] (GW)
Pulverized Coal	2058	27.5	0.0459	39	95	0.043	B	
IGCC	2378	38.7	0.0292	46	90	0.045	B	
IGCC-CCS	3496	46.1	0.0444	41	90	0.066	B	
GTCC-CCS	1890	19.9	0.0294	46	90	0.086	B	
Nuclear	3318	90.0	0.0049	33	95	0.054	B	
Geothermal	1711	165	0.00	11	90	0.044	B	23
GTCC	948	11.7	0.0200	54	95	0.062	Any	
GT	634	10.5	0.0317	40	95	0.076	Any	
Hydro	2242	13.6	0.0243	34	65	0.047	Any	2
Wind-Onshore	1923	30.3	0.00	34	35	0.076	S	8000
Wind-Offshore	3851	89.5	0.00	34	40	0.14	S	800
Solar Thermal	5021	56.8	0.00	34	40	0.17	S	100
Solar PV	6038	11.7	0.00	34	30	0.25	S	

Source: EIA (US Energy Information Administration), Office of Integrated Analysis and Forecasting, US Department of Energy. *Assumptions to the Annual Energy Outlook 2009*. DOE/EIA-0554(2009); Washington DC; US Government Printing Office; 2009b.

Natural Gas Price in 2040 vs. Total Cost



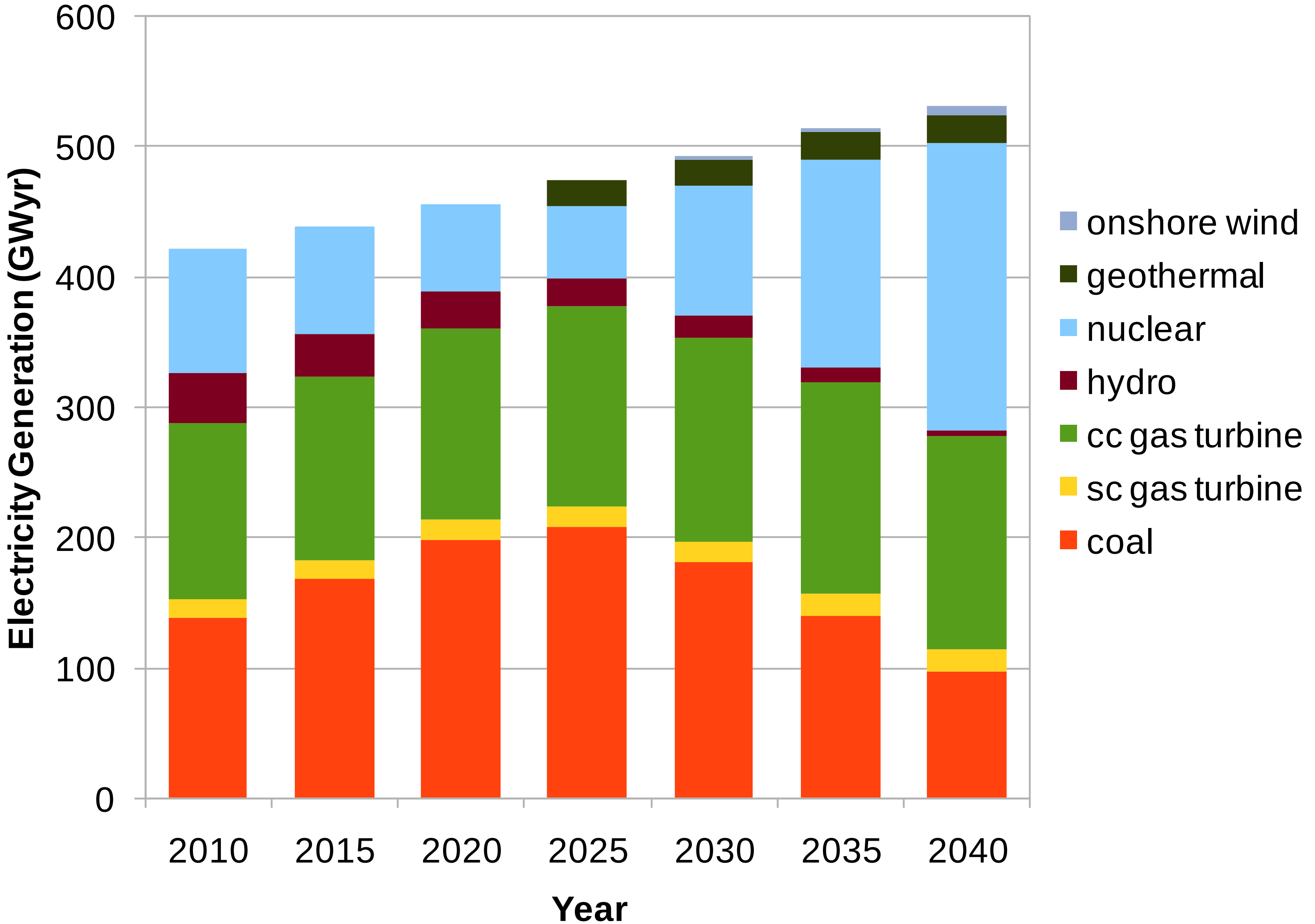
Constant Nat Gas Prices, Increasing CO_2



CO_2 emissions allowed to grow 0.6% annually

Natural gas prices remain constant at 4.5 $/GJ

High Nat Gas Prices, Decreasing CO_2



CO_2 emissions decrease 4.6% annually from 2030-2040

Natural gas prices increase 8.3% annually from 2030-2040

Next Steps for the TEMOA Project

- Build in partial equilibrium capability
- Store I/O data in a SQLite relational database
- Build a US national database drawing data from NCSU TIMES model currently under development
- Solve stochastic version of more complex models using progressive hedging

Acknowledgments

- Kevin Hunter, MS student, Civil Engineering, NCSU
- Sarat Sreepathi, PhD student, Computer Science, NCSU
- Aishwarya Ravichander, MS student, Electrical Engineering, NCSU
- Daniel Mahinthakumar, junior, Apex High

- Jean-Paul Watson and Bill Hart, Sandia National Laboratory

This work would made possible through the generous support of the National Science Foundation. CAREER: *Modeling for Insights with an Open Source Energy Economy Optimization Model*. Award #1055622

National Science Foundation
WHERE DISCOVERIES BEGIN

Questions?

Pyomo versus AMPL

Algebraic Formulation: $$\sum_{t \in tech} \text{production}_{t,seg} \geq \text{dmd}_{seg}, \forall seg \in \text{segments}$$

AMPL Formulation:

```
s.t. elc_demand{seg in segments}:
sum{t in tech[seg]} production[t] >= dmd[seg];
```

Pyomo Formulation:

```
 model.elc_demand = Constraint( model.segments, rule=elc_demand )

def elc_demand (seg, model):
        "Constraint: Electricity production >= demand each segment"
        constraint_val = sum(
                model.production[t]
                for t in model.tech[ seg ]
        )
        return ( constraint_val >= model.dmd[ seg ] )
```

Use comment blocks to dynamically generate model documentation (via Sphinx). Can embed LaTeX formatting in comments.