IDADES
Institute for the Design of Advanced Energy Systems

Flexible Environments for Generation and Transmission Expansion Planning (GTEP) Analysis

Kyle Skolfield, John Siirola

Sandia National Laboratories

Stakeholder Meeting 9/19/2024

NATIONAL ENERGY TECHNOLOGY LABORATORY
BERKELEY LAB
Sandia National Laboratories
Carnegie Mellon
West Virginia University
UNIVERSITY OF NOTRE DAME
Georgia Tech

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

U.S. DEPARTMENT OF ENERGY

GTEP is Large; It Contains Multitudes

- **Expansion planning** optimizes the investment and operational decisions of the bulk power system (generation and transmission) over a long-term, multi-decade planning horizon

- Deployed to guide investment portfolios while considering evolving grid (e.g., retirements and decarb), dynamic technology, and economic conditions

- Problem scale rapidly explodes into intractability based on network size and discrete decision selection

Costs

- ❑ Which investments?
- ❑ Which operations?
- ❑ What objective?
- ❑ Which hazards?
- ❑ What (if any) recourse actions?

Time Horizon

- ❑ Decades?
- ❑ Years?
- ❑ Weeks?
- ❑ Days?
- ❑ Hours?
- ❑ Minutes?

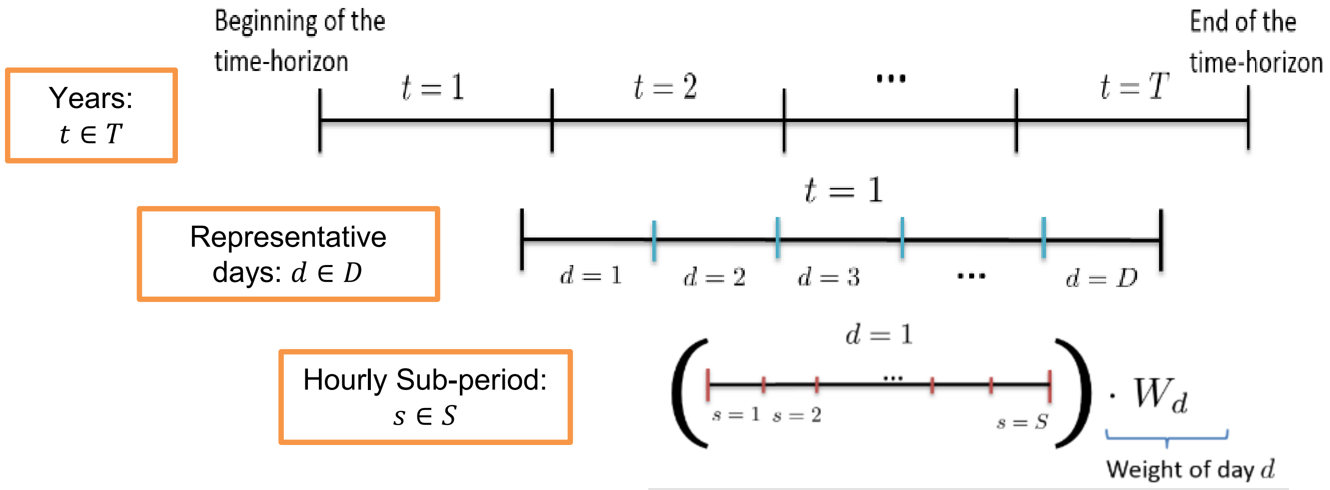
Uncertainties

- ❑ Renewable forecasts?
- ❑ Load forecasts?
- ❑ Resource forecasts?
- ❑ Climate?
- ❑ Weather?
- ❑ Technologies?

The set of GTEP-related questions is already enormous and growing rapidly

IDAES
Institute for the Design of Advanced Energy Systems

State-of-the-art (GTEP is hard)



Simplifying Assumptions:

- Reduced-network model
- Power flow approximations
- Scenario reduction
- Single-source uncertainty

Investments

- Either resilience or capacity, not both
- Either generation or transmission, rarely both
- Single timescale decisions

$\vee$

Objective

- Almost always pure **single** objective
- Rarely two competing objectives based on a tolerance trade-off parameter

$\vee$

Uncertainties

- At most one, often zero
- Typically modeled exogenously and abstractly
- Rarely geo-spatially located

Existing GTEP models are bespoke; i.e., highly individualized and tailored to specific questions or cases

Sample Models in Active Development

ReEDS

Features	
✓ Well-established ✓ Large scale functionality (CONUS) ✓ High level aggregation	❖ Relaxed model (LP) ❖ Limited disaggregation capability ❖ Rigid assumptions

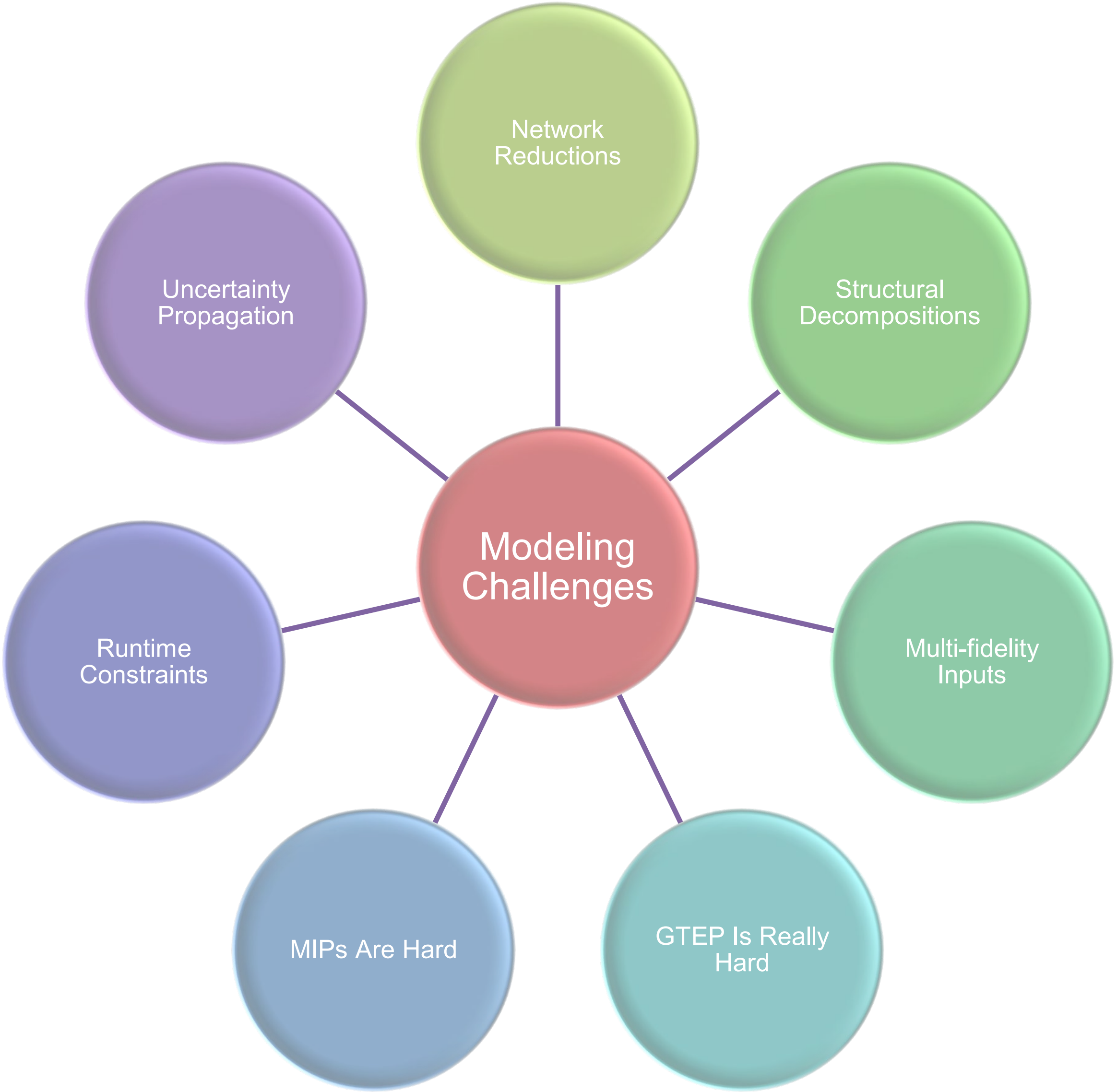
GenX.jl

Features	
✓ Flexible scale ✓ Relatively high configurability ❑ Entirely in Julia	❖ Deterministic and/or myopic multi-stage optimization only ❖ Representative time periods repeatedly duplicated

These tools are more flexible and can answer more questions than previous GTEP attempts, but are still *just models*

IDAES Institute for the Design of Advanced Energy Systems

We Don't Want *A* Model



We need to identify the "right" model from a superset AND the "right" way to solve it

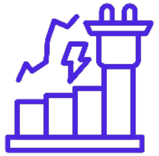
How do we do that when designing, building, testing, and evaluating *new* models is so expensive?

IDAES
Institute for the Design of Advanced Energy Systems

What is IDAES GTEP if Not a Model?

Electrification

- ❑ Increased unpredictability
- ❑ Impact of inverter-based resources
  - ❑ Reliability
  - ❑ Valuation
  - ❑ Inertia

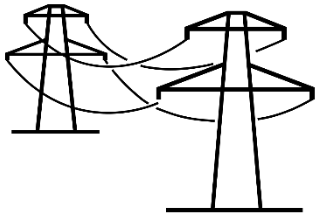


Regulation

- ❑ Decarbonization
- ❑ Renewable minima
- ❑ Technology restrictions
- ❑ FERC 1920

Modeling Choices

- ❑ Appropriate network reductions
- ❑ Improved temporal multi-fidelity
- ❑ Case study independence
- ❑ Flexible investment options
- ❑ Common & verified flow models (e.g., EGRET)
- ❑ Consistent Verification & Validation (e.g., PRESCIENT)



We need a *superstructure* to explore multiple GTEP questions without creating new models

IDAES®
Institute for the Design of Advanced Energy Systems

We Want A ***Superstructure***

Superstructure (metamodel design)

- What combination of input fidelities is necessary?
- How do we decompose the problem?
- How much can we simplify the network?

Model Design

- Uncertainties
- Regulations
- Investments

Algorithmic Decisions

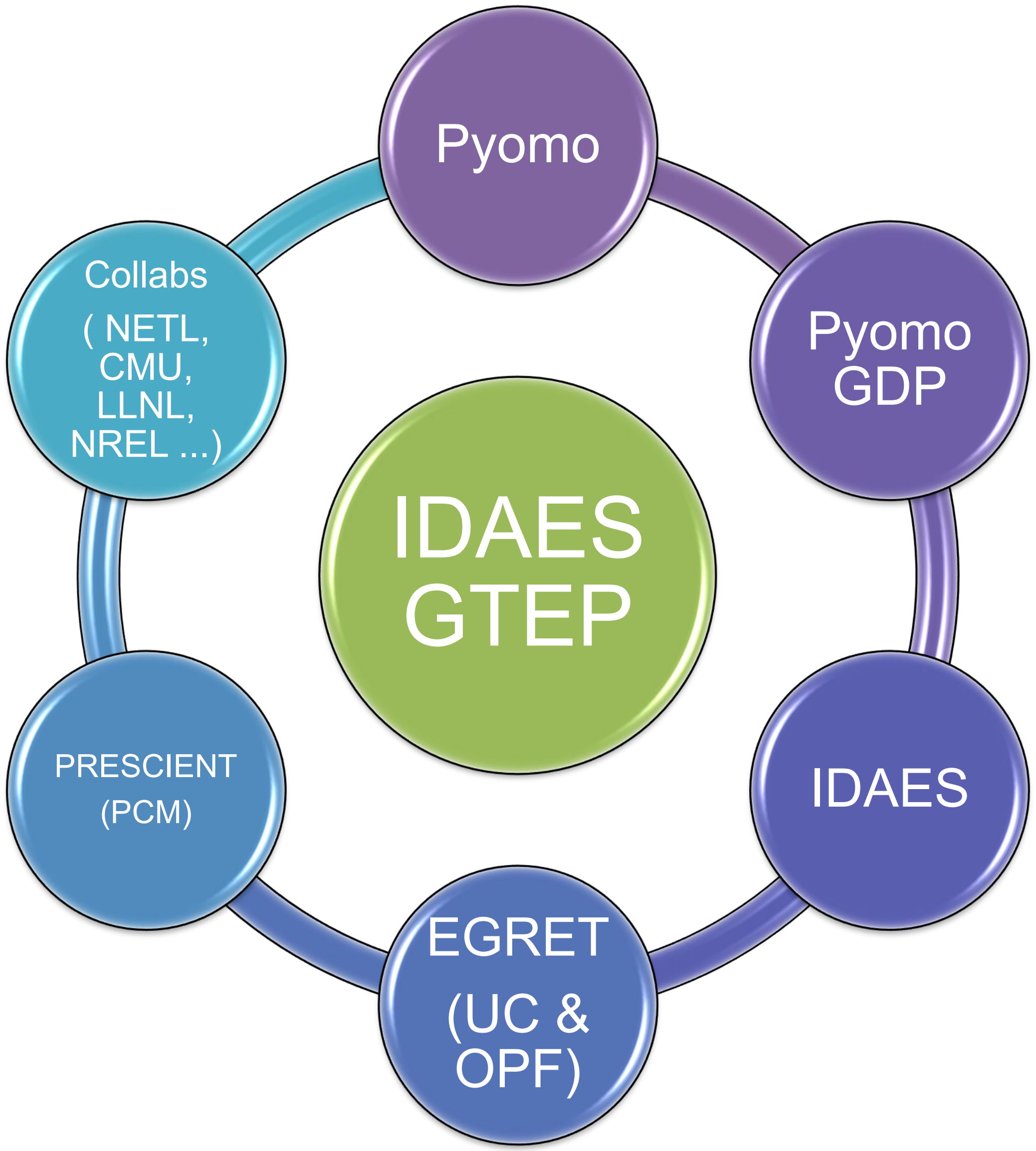
- Multi-fidelities
- Decomposition
- Simplified Network

IDEAS GTEP Superstructure is Built on an Ecosystem

IDAES GTEP builds on and interacts with established, trusted systems

PYOMO

PYOMO GDP



IDAES Institute for the Design of Advanced Energy Systems

How Do We Build a Superstructure?

Extensive Use of *Generalized* Disjunctive Programming

- Unit Commitment
- Investment
- Storage

Make Case Study and Model Independent

- Generation decisions (technologies, locations, sizes, etc.)
- Transmission decisions
- Storage decisions
- Regional requirements
- Time periods

Select from a Menu of Relevant Assumptions

Investment Periods
- Decadal
- Annual
- Monthly
- Other

Representative Periods
- Daily
- 72-hourly
- Weekly
- Other

Power Flow Model
- Transport
- DCPF
- ACPF
- LPAC

Commitment Periods
- Hourly
- Sub-hourly
- Other

Dispatch Periods
- Hourly
- 15-minutely
- 5-minutely
- Other

GDP as a modeling framework enables easy selection of "menu items" to model – in isolation or jointly

GDP Defined

There is no such thing as a *BINARY* variable for yes/no decisions.

Variable Types:

- Real
- Integer

- ***Boolean***

Enables Logical Constraints:

$$Y_{ik} \Rightarrow \begin{cases} M_{ik}x + N_{ik}z \leq e_{ik} \\ r_{ik}(x,z) \leq 0 \end{cases}$$

Problems naturally described by GDP:

- Yes/no decision sets
- No-overlap
- Equipment selection
- State alternatives
- “Choose exactly *n* of”

GDP formulation admits a common logical structure for any Boolean decision variables

IDAES Institute for the Design of Advanced Energy Systems

PYOMO GDP

Example of GTEP as GDP

$$Y_{gT}\left[\begin{bmatrix} U_{g\tau}^{ON} \\ \underline{p}_g \le p_{gt} \le \overline{p}_g \end{bmatrix} \vee \begin{bmatrix} U_{g\tau}^{SU} \\ 0 \le p_{gt} \le \underline{p}_g \end{bmatrix} \vee \begin{bmatrix} U_{g\tau}^{SD} \\ 0 \le p_{gt} \le \underline{p}_g \end{bmatrix}\right]$$

Disjuncts

Disjunctions

GDP nests and isolates menu items, handling them consistently and independently

IDAES Institute for the Design of Advanced Energy Systems

GDP Gives Significant Advantages

- **Flexibility**
  - Abstracts modeling decisions from specific case studies or implementations
- **Modularity**
  - Enables easy inclusion/exclusion of modeling subcomponents (e.g., storage, transmission investment, power flow approximation, etc.)
- **Comprehensibility**
  - Results in cleaner, easier to understand codebase

Sample Menu Items

Decisions	Flow Models	Temporal Scale
Generation	DCOPF	Identical Length
Transmission	Copperplate	Any Number
Storage		Non-uniform Length
Switching	ACOPF	
Reconductoring		

Any case study in a standard format (e.g., RTS-GMLC) can be modeled with any set of assumptions *without* modifying data

Clean Math ⇔ Clean Code

```
@b.Disjunct(m.thermalGenerators)
    def genOn(disj, generator):
        # operating limits
        b = disj.parent_block()

        # Minimum operating Limits
        @disj.Constraint(b.dispatchPeriods)
        def operating_limit_min(d, dispatchPeriod):
            return (
                m.thermalMin[generator]
                <= b.dispatchPeriod[dispatchPeriod].thermalGeneration[generator]
            )

        # Maximum operating limits
        @disj.Constraint(b.dispatchPeriods)
        def operating_limit_max(d, dispatchPeriod):
            return (
                b.dispatchPeriod[dispatchPeriod].thermalGeneration[generator]
                + b.dispatchPeriod[dispatchPeriod].spinningReserve[generator]
                <= m.thermalCapacity[generator]
            )
```

$$\begin{bmatrix} U_{g\tau}^{ON} \\ \underline{p}_g \leq p_{gt} \leq \overline{p}_g \end{bmatrix} \vee$$

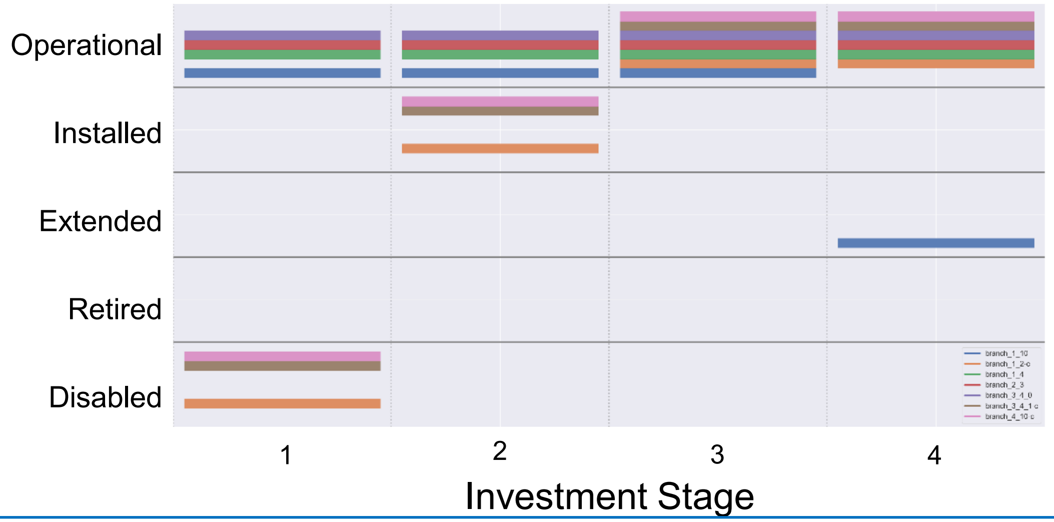
```
@b.Disjunction(m.thermalGenerators)
    def genStatus(disj, generator):
        return [
            disj.genOn[generator],
            disj.genStartup[generator],
            disj.genShutdown[generator],
            disj.genOff[generator],
        ]
```

GDP permits direct translation between abstract math formulations and concrete code implementations

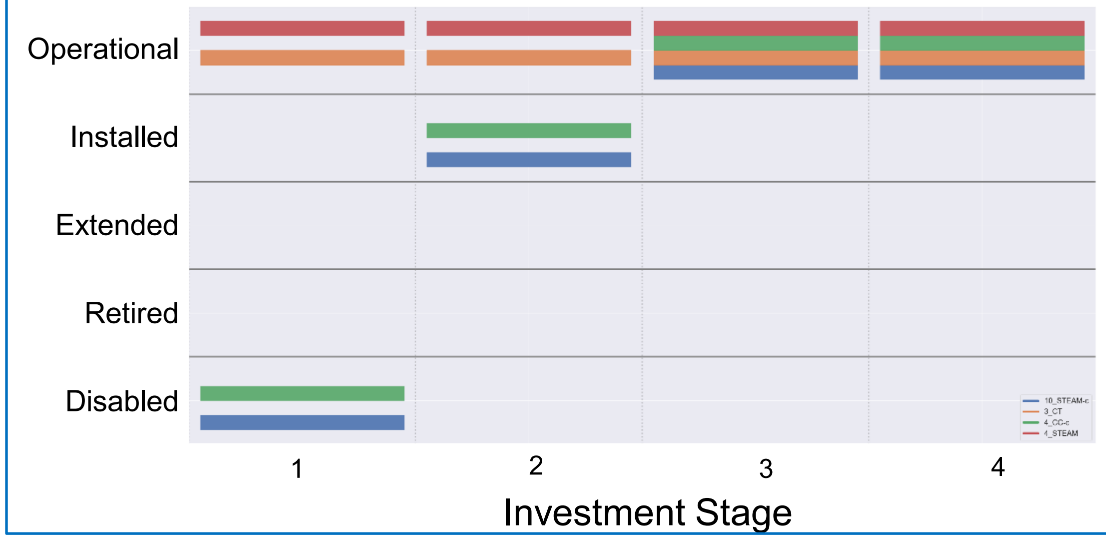
The whole IDEAS GTEP architecture can be built on trusted, unit-tested components (e.g., EGRET, PRESCIENT, etc.) in a plug-and-play fashion

Generalized Viz

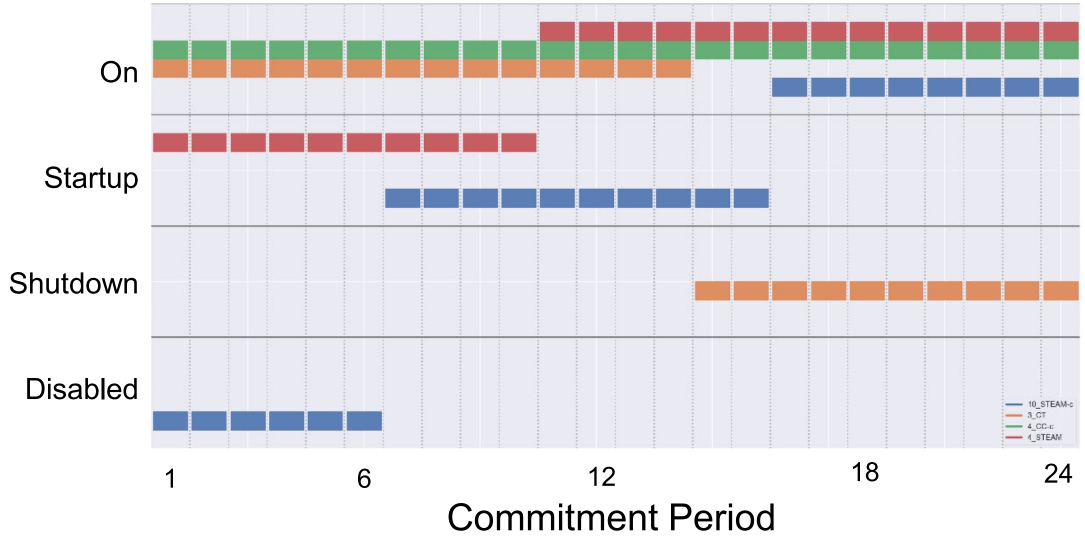
TRANSMISSION INVESTMENT STATUS



GENERATION INVESTMENT STATUS



UNIT COMMITMENT STATUS



Consistent architecture means visualization & analysis are straightforward for any questions about sets of Boolean decision variables

What IDAES GTEP Is ***NOT***

Simulation Tool

Policy Predictor

Stability Analysis

Market Model

What IDAES GTEP *IS*

Model Selection Optimization

Infrastructure Planning Optimization

Plant-Grid Interaction Optimization

Reliability / Resilience Optimization

Capabilities

Present and Accessible

- Planning Decisions
  - Generation
  - Transmission
- Flow Models
  - Copperplate
  - DCOPF
- Temporal Scale
  - Hourly UC
  - Hourly ED
  - Daily Rep. Periods
  - Yearly GTEP
- Test Cases
  - 5 Bus
  - 11 Bus
  - 31 Bus

Present and Guide-able

- Planning Decisions
  - Storage
- Temporal Scale
  - Arbitrary and consistent
- Test Cases
  - 133 Bus
- Visualization Tools
- Validation and Verification via Prescient

Near Future (<6 months)

- Planning Decisions
  - Undergrounding
  - Reconductoring
  - Reliability
- Algorithmic
  - Stochastic
  - Decomposition
- Temporal Scale
  - Arbitrary and inconsistent
- Test Cases
  - 2000/7000 Bus

Mid Future (<2 years)

- Planning Decisions
  - Resilience
- Algorithmic
  - AOS
  - Automated (Dis-) Aggregation
- Flow Models
  - ACOPF
- Test Cases
  - 10,000 Bus Geolocated [including line paths] CA

Where do I find IDAES GTEP?

REPO

https://github.com/IDAES/idaes-gtep

DOCS

https://idaes-gtep.readthedocs.io/en/latest/

QUESTIONS/CONTACT

jkskolf@sandia.gov