SEI Stockholm Environment Institute

SEI Asia Centre

Training on Low Emissions Analysis Platform

Day 3: 21 October 2021

Jason Veysey
Deputy Director, Energy Modeling Program
Stockholm Environment Institute
jason.veysey@sei.org

Charlotte Wagner
Scientist, Energy Modeling Program
Stockholm Environment Institute
charlotte.wagner@sei.org

Sida
SWEDISH INTERNATIONAL
DEVELOPMENT COOPERATION AGENCY

Workshop registration

Please register your attendance daily

Participants need to register for at least 3 days

to be eligible for an attendance certificate

Registration link day 3

https://tinyurl.com/SEIAsiaLEAPtraining-3408

Password: Day03$

Workshop connection information

Web meetings

https://tinyurl.com/SEIAsiaLEAPtraining

Zoom meeting ID: 872 2041 5222

Zoom passcode: 353649

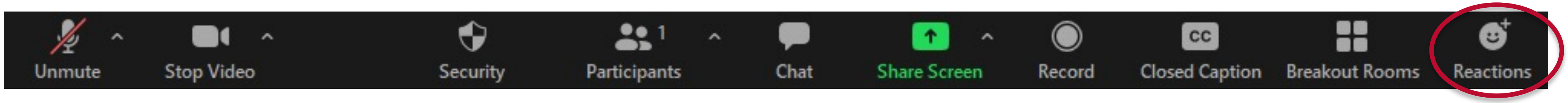
Shared files

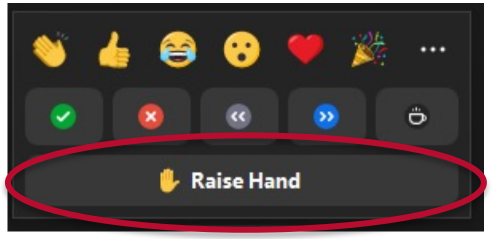
https://tinyurl.com/SEIAsiaLEAPMaterials

Password: seiasia1021

Zoom etiquette

- Please:
  - **Enter your name in Zoom** so meeting hosts can identify you in participant lists
  - **Mute yourself when not speaking**
  - Use your camera if possible
  - If you have a question, **raise your hand in Zoom**



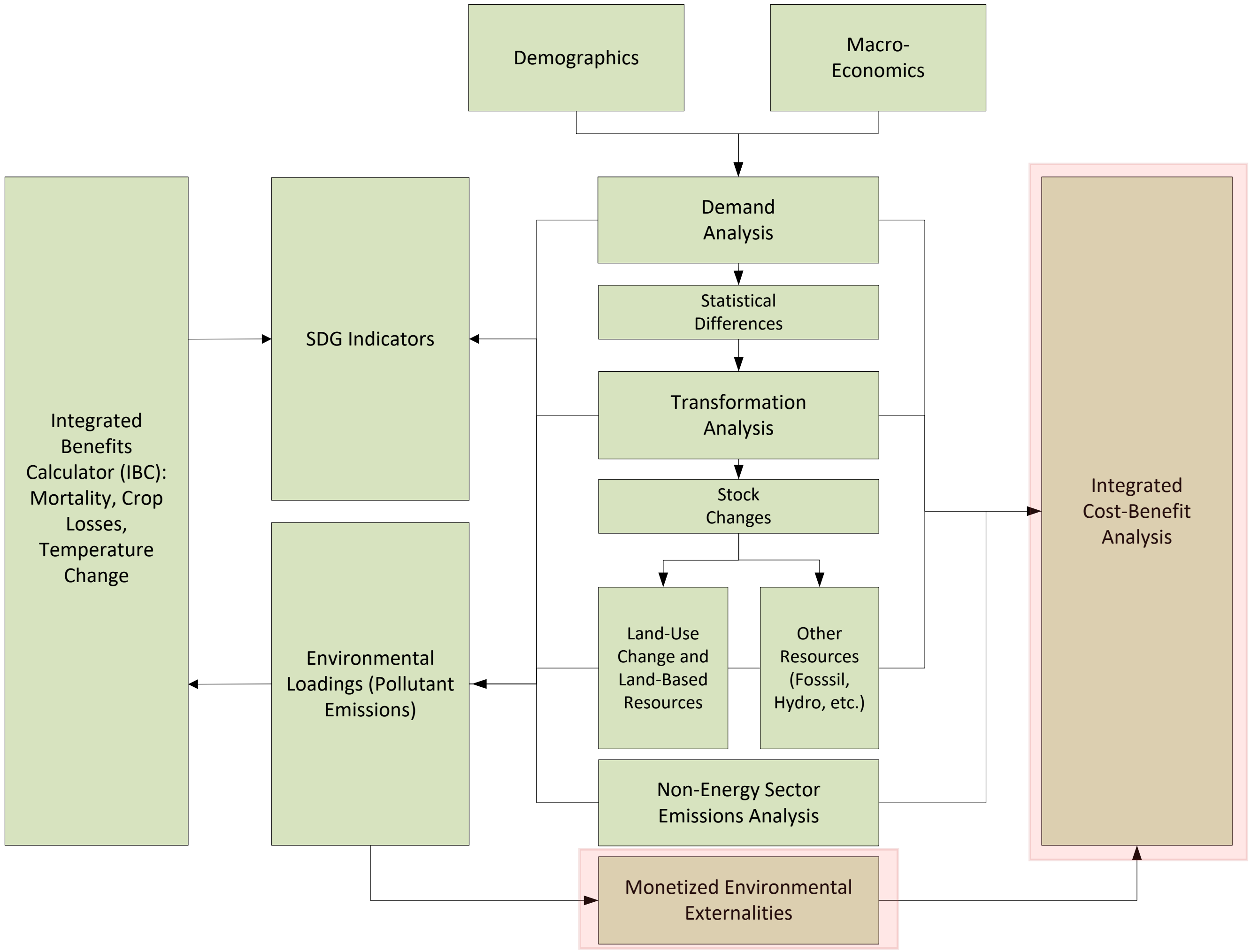


Workshop overview

- **Day 1:** Introduction to LEAP and energy demand modeling
- **Day 2:** Energy supply and emissions modeling
- **Day 3:** Cost-benefit analysis and optimization modeling
- **Day 4:** Linking LEAP and WEAP and other advanced topics

Cost-benefit analysis with LEAP

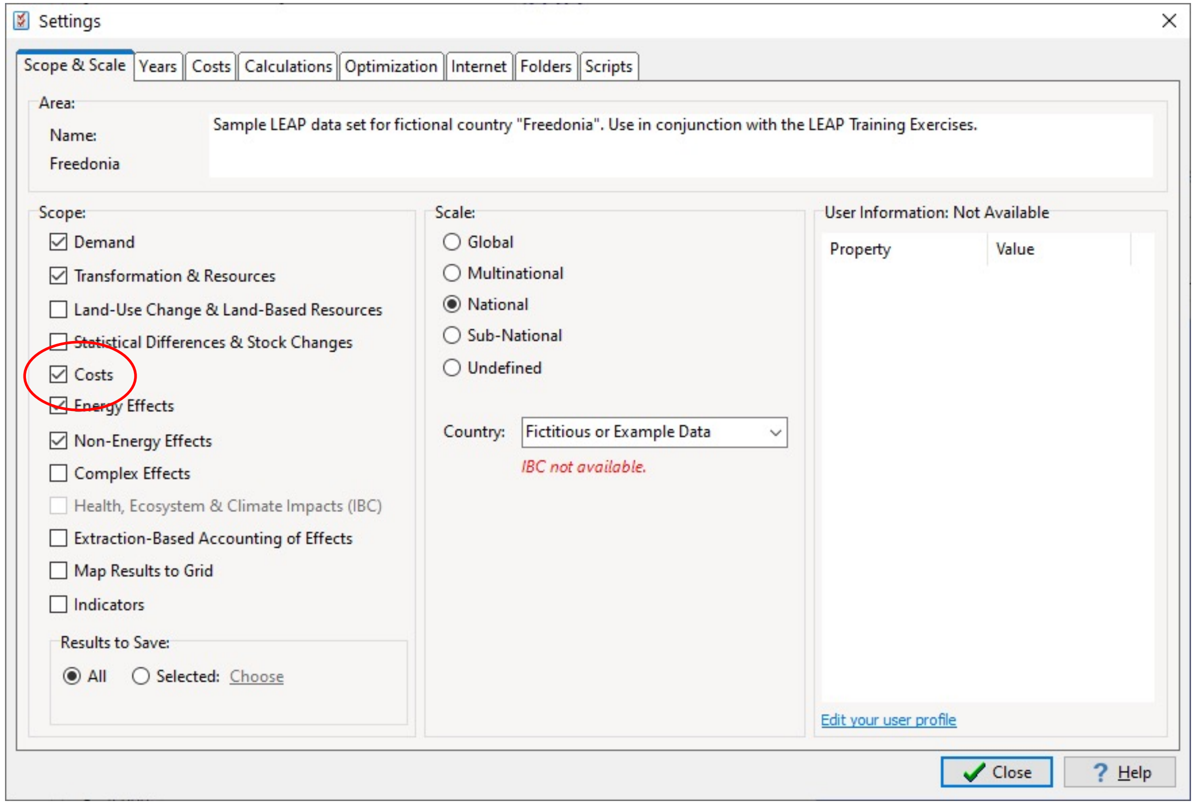
Structure of a representative LEAP analysis

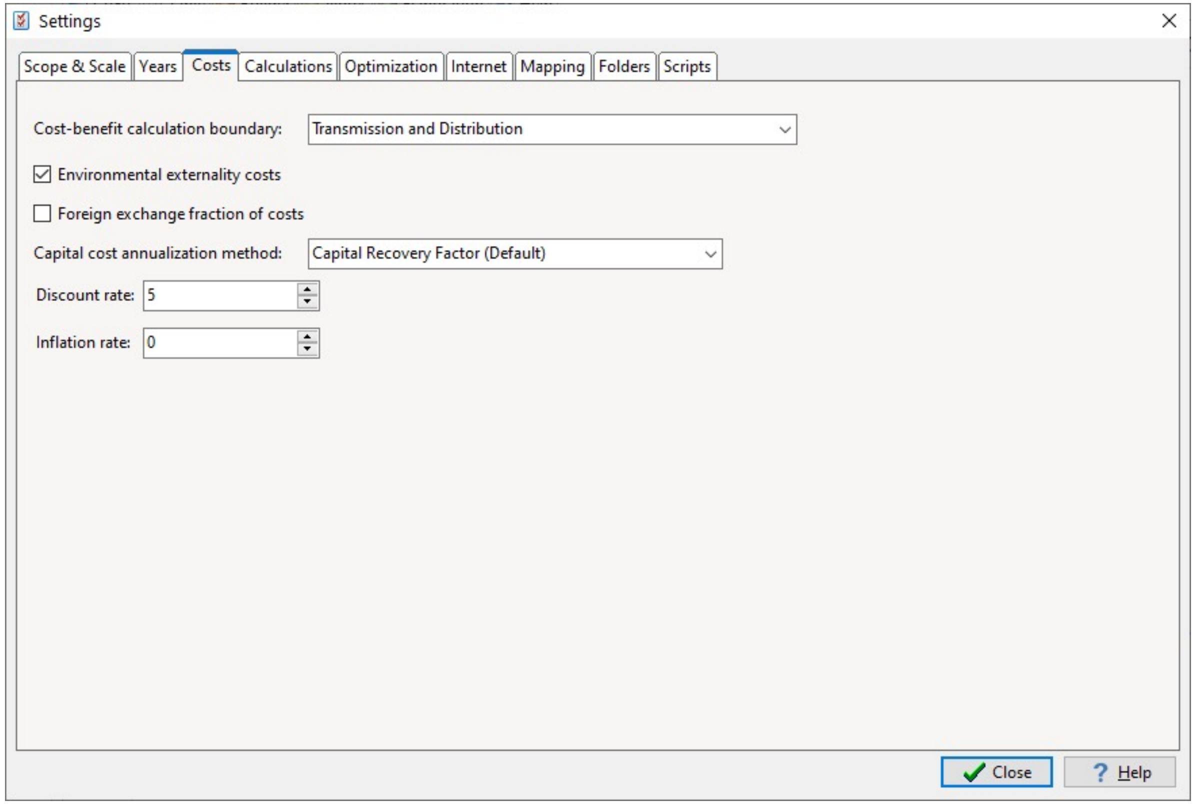


Cost-benefit analysis overview

- **Costs** can be included in **energy and non-energy** modeling in LEAP
- Typically, **real social costs** are used
- Both **undiscounted** and **discounted** costs can be reported
- LEAP avoids double counting energy costs by drawing a consistent **boundary around costing analysis**

Enabling cost modeling

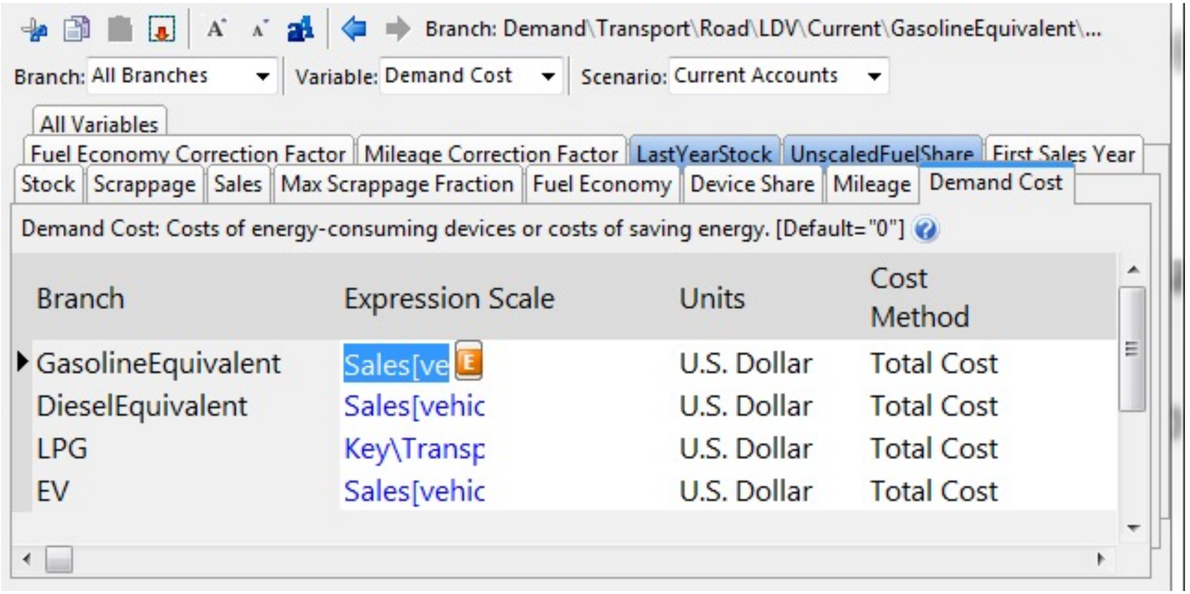




Energy demand costs

Several costing methods supported

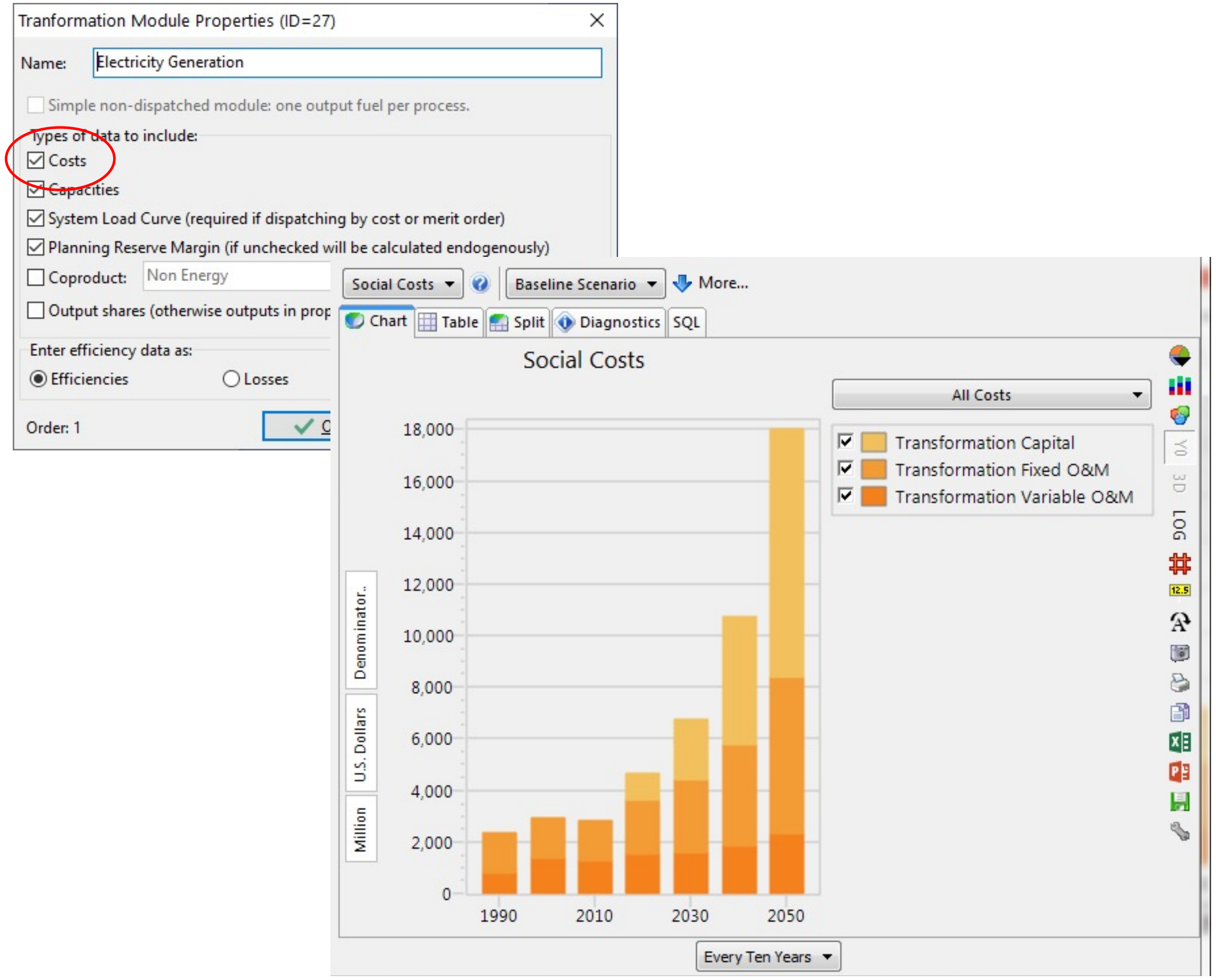
- *Costs per activity*
  - E.g., cost per vehicle-km traveled
- *Total annual costs*
  - E.g., total annual costs of a driver education program
- *Cost of saved energy relative to a reference scenario*
  - E.g., incremental costs per unit energy saved due to efficiency retrofits of long-haul trucks



Demand costs exclude fuel costs (these are calculated separately on supply side of model)

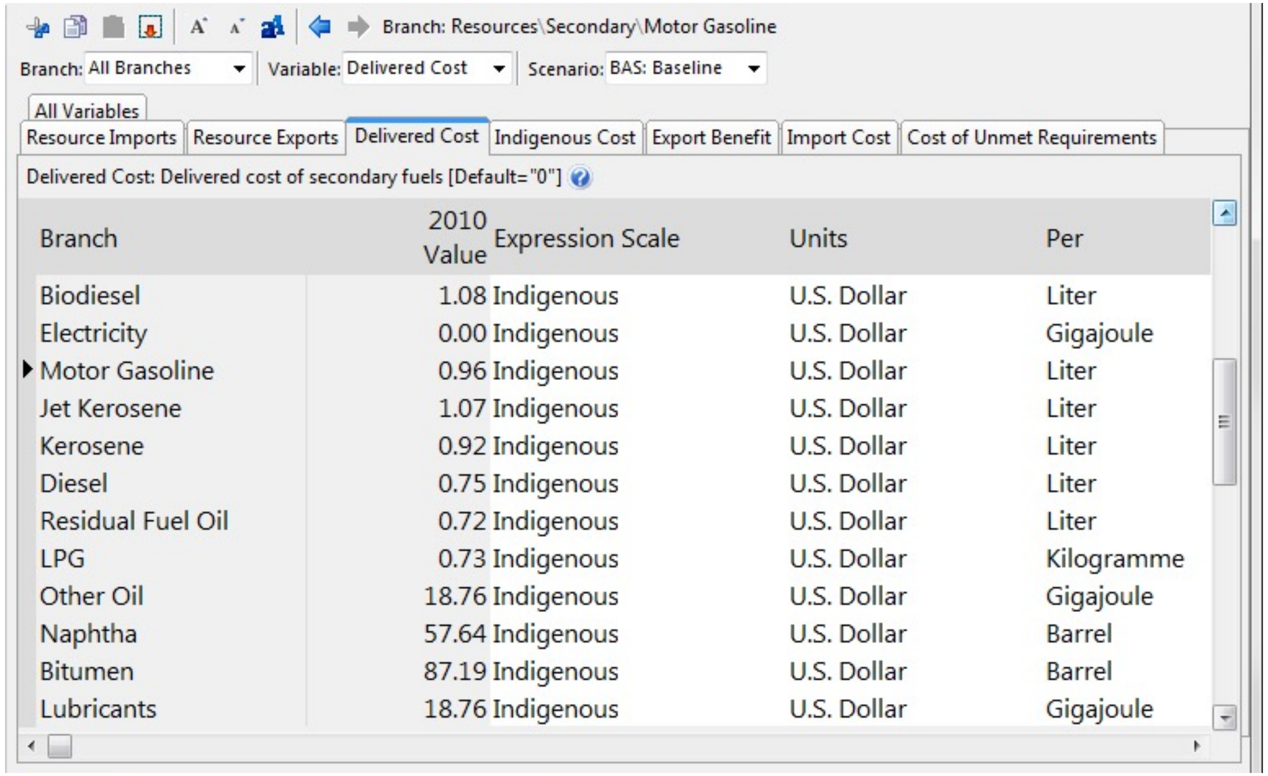
Energy transformation costs

- Considered for transformation modules inside costing boundary and for which costing is activated
- Can include:
  - Capital costs (amortized using process-specific interest rates)
  - Fixed and variable O&M costs
  - Decommissioning costs
  - Other industry (module) costs
- Fuel input costs calculated separately based on costs of other modules and resource costs



Resource costs

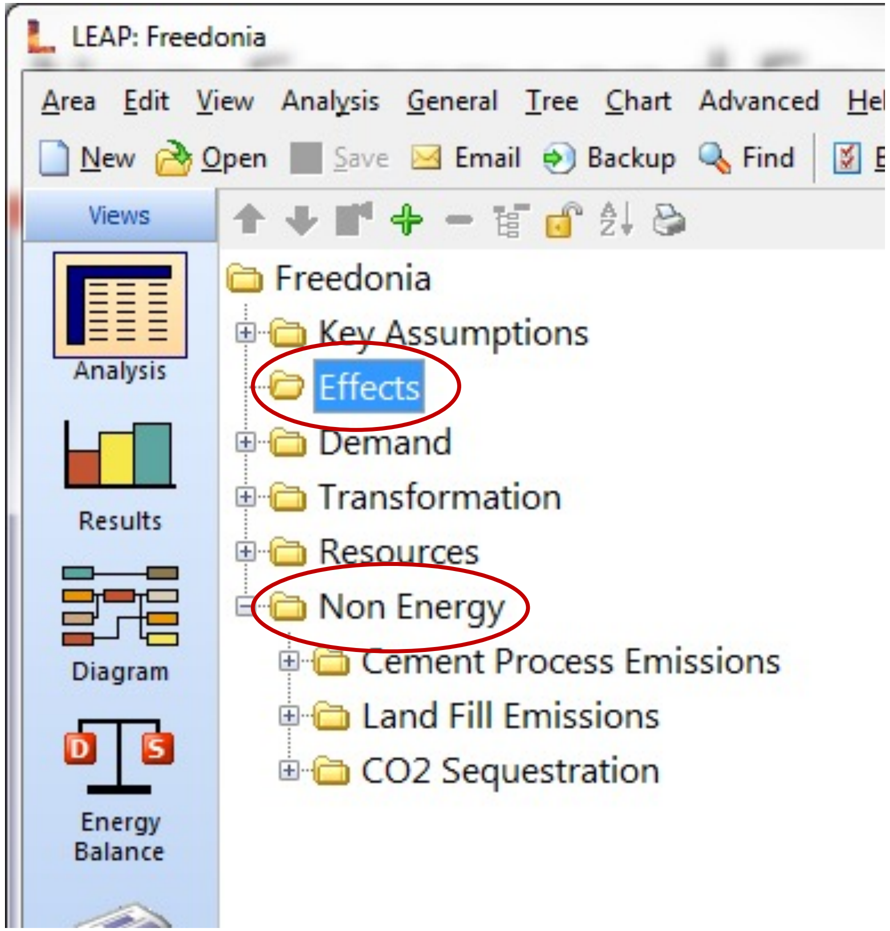
- Import costs and export benefits
- Depletion/extraction costs (indigenous costs)
- Social costs of unmet requirements for fuels (e.g., costs of brown-outs due to inadequate electricity supply)



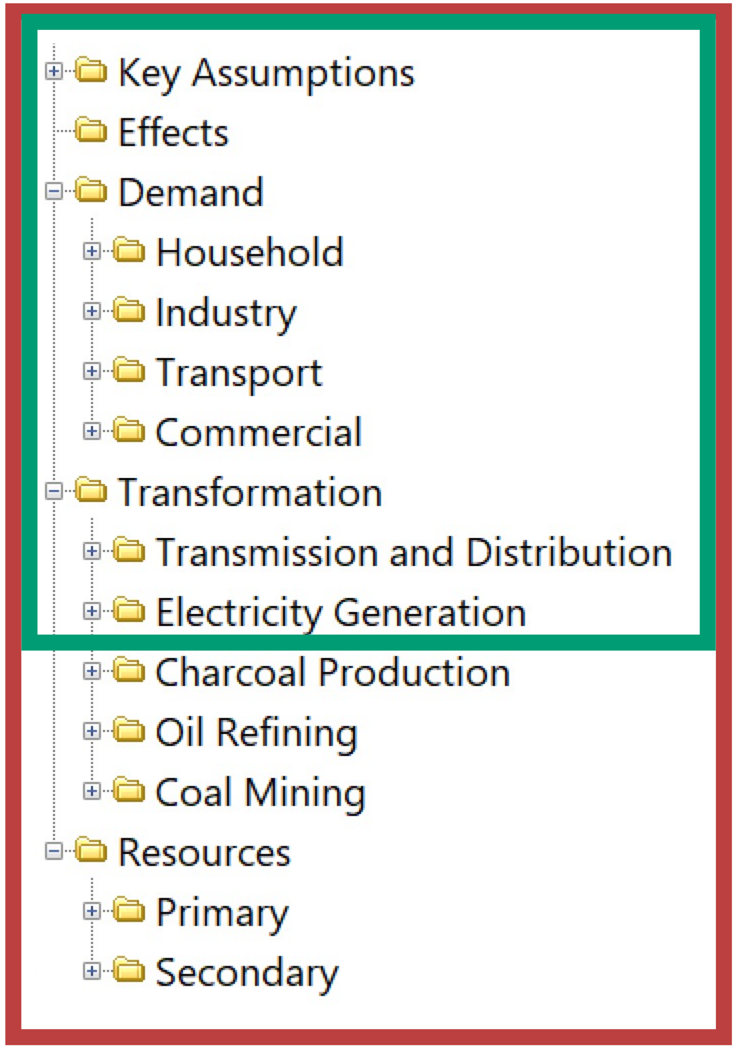
Branch	2010 Value	Expression Scale	Units	Per
Biodiesel	1.08	Indigenous	U.S. Dollar	Liter
Electricity	0.00	Indigenous	U.S. Dollar	Gigajoule
Motor Gasoline	0.96	Indigenous	U.S. Dollar	Liter
Jet Kerosene	1.07	Indigenous	U.S. Dollar	Liter
Kerosene	0.92	Indigenous	U.S. Dollar	Liter
Diesel	0.75	Indigenous	U.S. Dollar	Liter
Residual Fuel Oil	0.72	Indigenous	U.S. Dollar	Liter
LPG	0.73	Indigenous	U.S. Dollar	Kilogramme
Other Oil	18.76	Indigenous	U.S. Dollar	Gigajoule
Naphtha	57.64	Indigenous	U.S. Dollar	Barrel
Bitumen	87.19	Indigenous	U.S. Dollar	Barrel
Lubricants	18.76	Indigenous	U.S. Dollar	Gigajoule

Non-energy and pollutant costs

- Non-energy – defined for non-energy emissions
  - Total costs of annual emissions of a pollutant from a source
- Environmental externalities – defined for selected pollutants
  - Costs per unit of emissions



Costing boundary



Red box: ***complete energy system***

- Energy demand costs
- Bottom-up costing for all energy transformation modules
- Resource import, export, and extraction costs

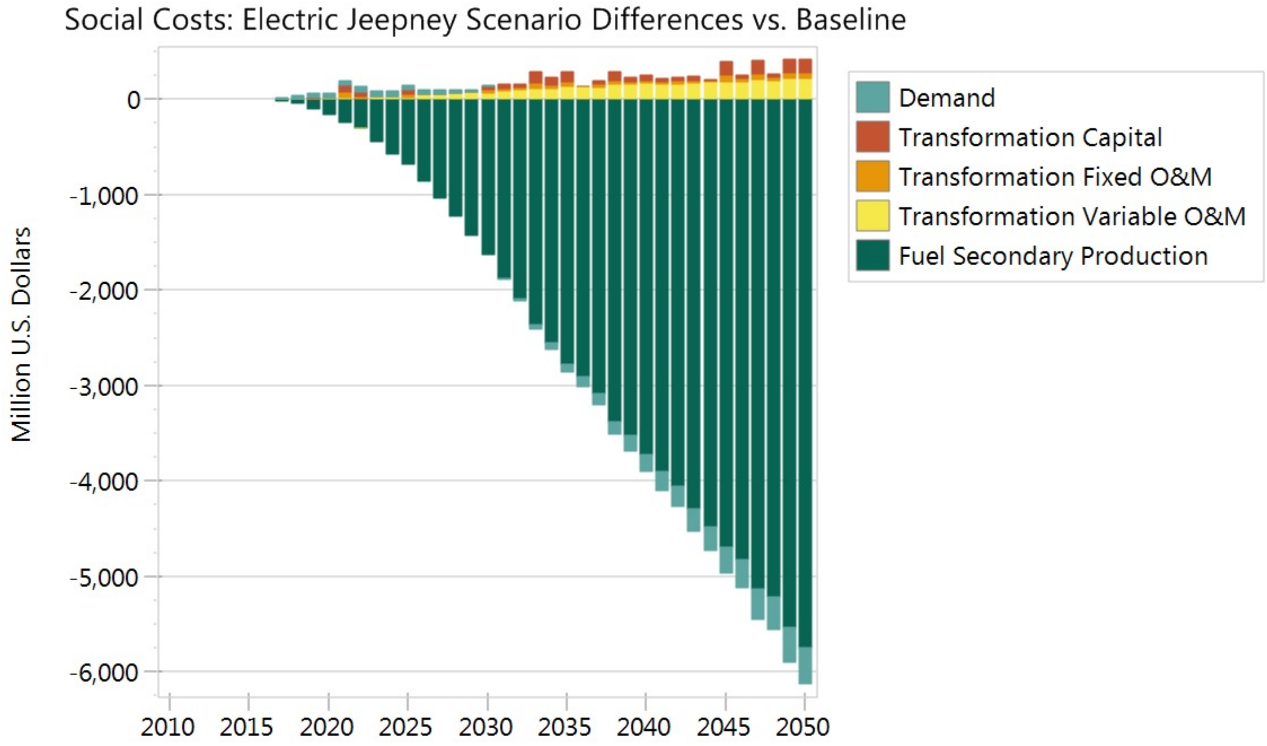
Green box: ***subset of energy system***

- Energy demand costs
- Bottom-up costing for transformation modules within boundary
- Bottom-up production costs in other modules and resource import, export, and extraction costs *not* used => instead, **a "delivered cost" is used for fuels produced outside boundary**

Non-energy and pollutant costs not affected by boundary

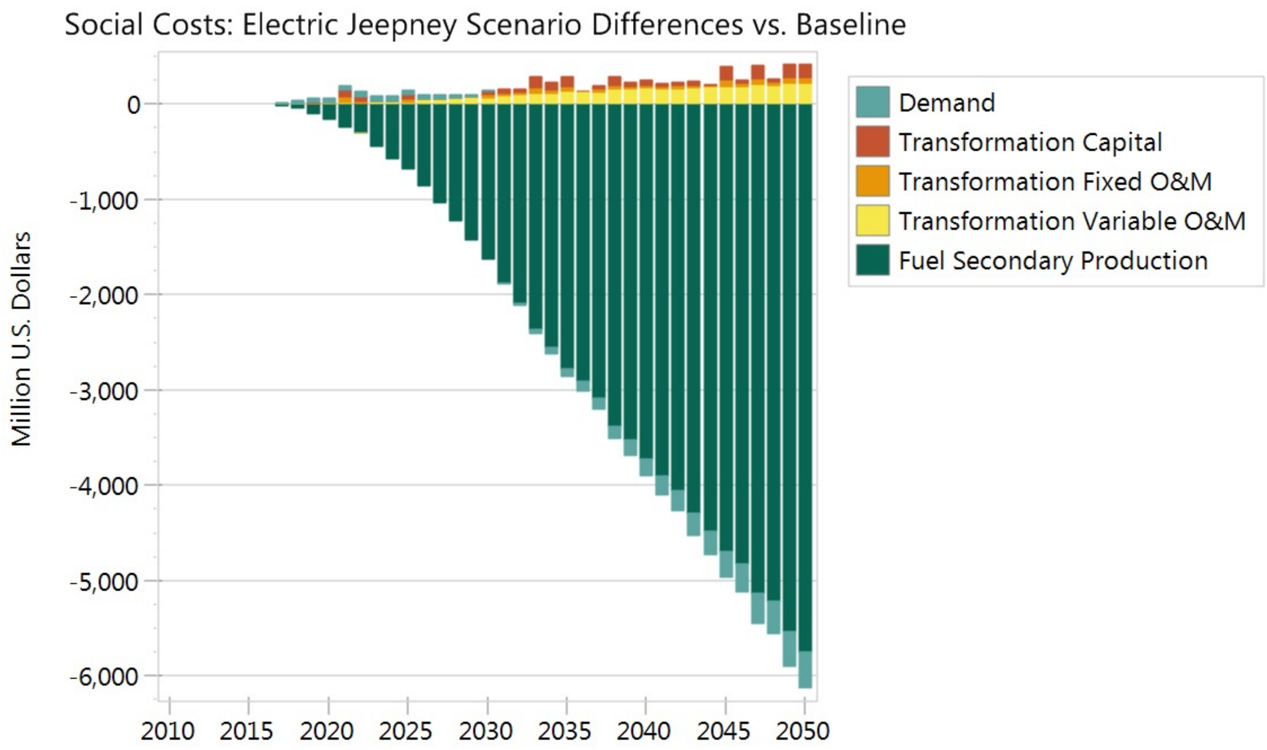
Cost reporting in Results view

- Real or discounted costs
- Annual or cumulative costs
- All or selected cost types/categories (e.g., demand, fuel)
- Incremental costs of one scenario compared to another
- Currency conversions



Cost reporting in Results view

- Real or **discounted** costs
- Annual or **cumulative** costs
- All or selected cost types/categories (e.g., demand, fuel)
- **Incremental** costs of one scenario compared to another
- Currency conversions



Optimization modeling with LEAP and NEMO

How to model energy systems?

- Investment
- Operations
- Impacts
- Limits
- Vulnerabilities

A common approach: least-cost optimization

Meet energy requirements while minimizing discounted energy system costs

- Capital
- Operation & maintenance (O&M)
- Fuel imports and exports
- Salvage values
- Environmental externalities

Real costs, typically quantified from a social perspective

Basic rationale

- **Efficiency**
  - *Other things being equal, it makes sense to meet requirements in the least costly way*
- **Planner's perspective**
  - *It's what a planner would do*
- **Market dynamics**
  - *It's a reasonable approximation of how (some) energy markets operate*
- **Technical specificity**
  - *It accounts for important physical features of energy technologies*

An idealization, but a potentially useful one

Mathematical formulation

$$minimize: \sum \begin{array}{c} discounted\ energy \\ system\ costs \end{array}$$

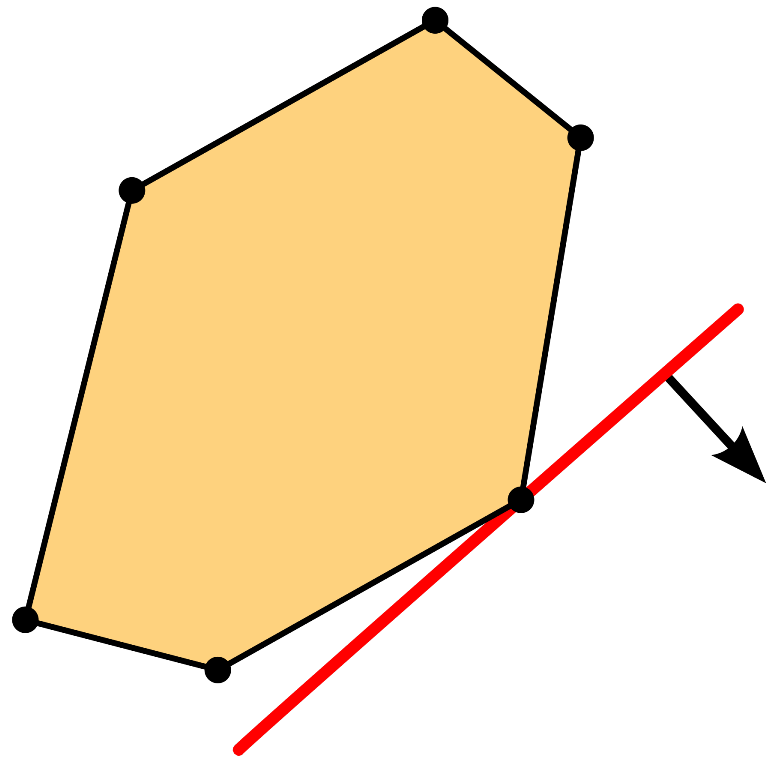
$$subject\ to: \ g_i(\boldsymbol{x}) \leq 0 \quad i = 1, \ldots, m$$

$$and: \ h_j(\boldsymbol{x}) = 0 \quad j = 1, \ldots, n$$

$$and: \ \boldsymbol{x}^L \leq \boldsymbol{x} \leq \boldsymbol{x}^U$$

- Sum of discounted costs is objective function
- x is a vector of model variables (e.g., capacity built; values determined by model)
- $\mathrm{g(x)}$ and $\mathrm{h(x)}$ are constraints (e.g., operating limits, accounting identities, demand requirements)
- $\mathrm{x^L}$ and $\mathrm{x^U}$ are lower and upper bounds for variables
- Modeler can specify other (exogenous) values in constraints and objective function – these are parameters (e.g., power plant efficiencies)

Graphical interpretation



en.wikipedia.org/wiki/Linear_programming

- A simple linear program with two variables
- The set of feasible solutions is the yellow polygon
- Each line on the polygon is a particular constraint
- The linear cost function is represented by the red line and the arrow: the arrow indicates the direction of optimization (cost minimization)

Full vs. partial energy system optimization

- Some tools allow for full energy system optimization
  - Starting with useful energy requirements (e.g., for transport or heating), they identify least cost configuration of entire energy system subject to constraints
  - Examples include NEMO, TIMES/MARKAL, MESSAGE, HOMER, and OSeMOSYS
- Others support partial energy system optimization
  - Only selected sectors can be modeled – typically, electricity production
  - Examples include LEAP, WASP, and PLEXOS
- In practice, most optimization analyses focus on energy supply

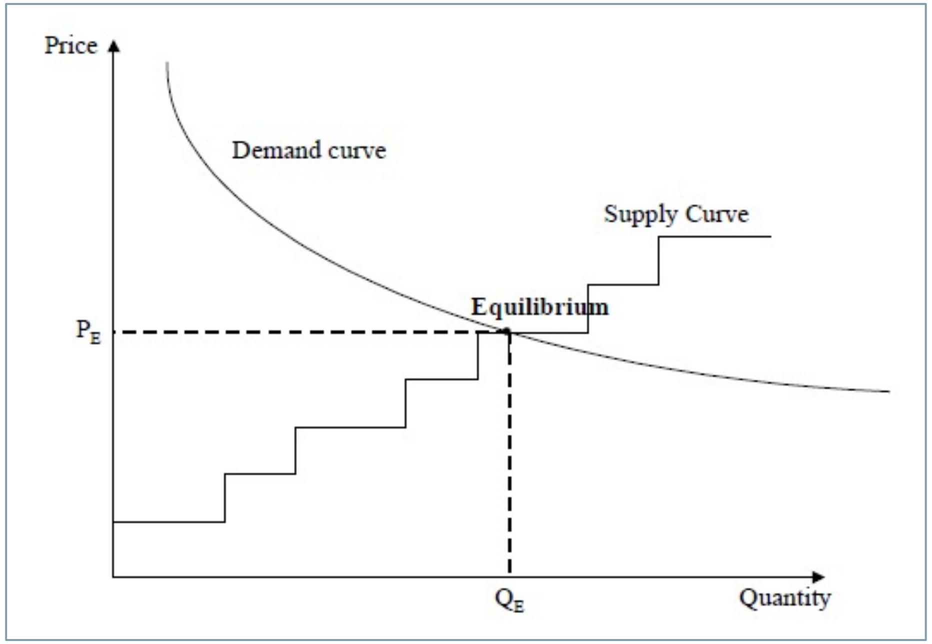
Fixed demand vs. partial equilibrium

Fixed demand

- Demand does not depend on supply
- Demand determined first, then supply found through optimization

Partial equilibrium

- Demand subject to price elasticity, price depends on supply
- Supply-demand equilibrium found through optimization
- Motivated by neoclassical economic hypotheses:
  - Competitive markets
  - Profit-maximizing firms
  - Utility-maximizing consumers

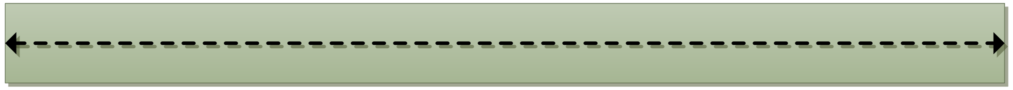


- Most energy system models consider demand to be fixed

Perfect vs. limited foresight

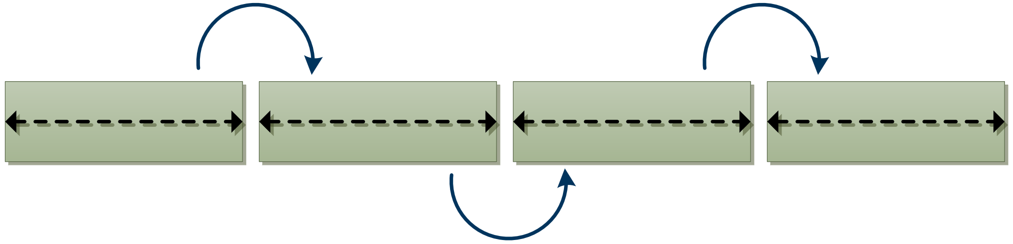
Perfect foresight

- Costs for all years in planning horizon minimized simultaneously (global optimum found)



Limited foresight

- Costs for subsets of years minimized
- Results from first interval provide starting conditions for second, and so on



Multi-objective optimization

Pareto-optimum: can't improve one objective without worsening another

- **Minimize costs *and***
  - Minimize environmental impacts
  - Maximize energy independence

  …

- Various approaches
  - Weight objectives
  - Optimize primary objective but with limits on allowed values for others
  - Find Pareto-optimum solutions and choose the one that minimizes the maximum deviation from single-objective optima ("min-max")

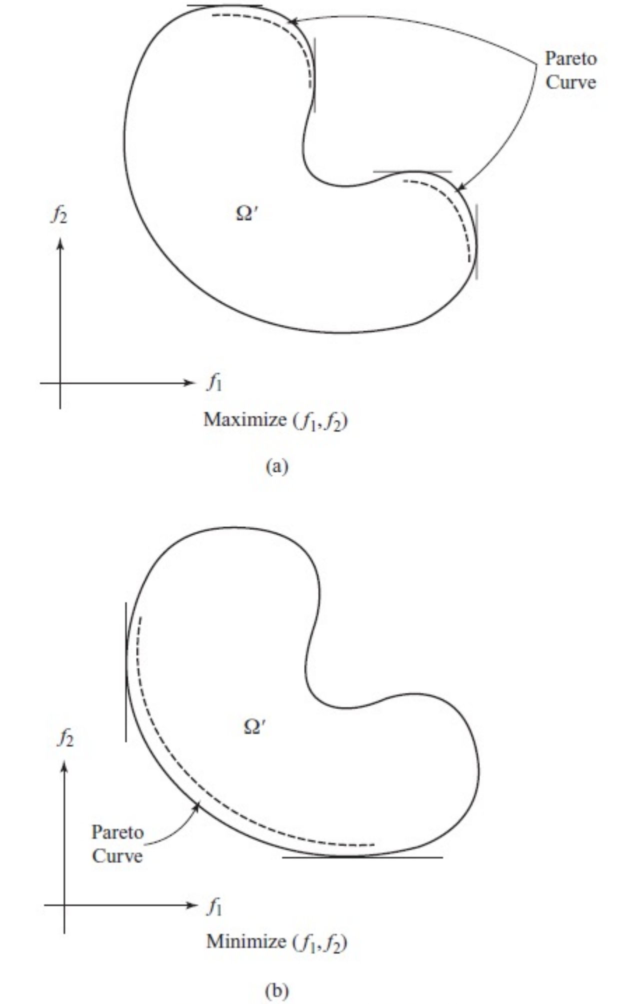
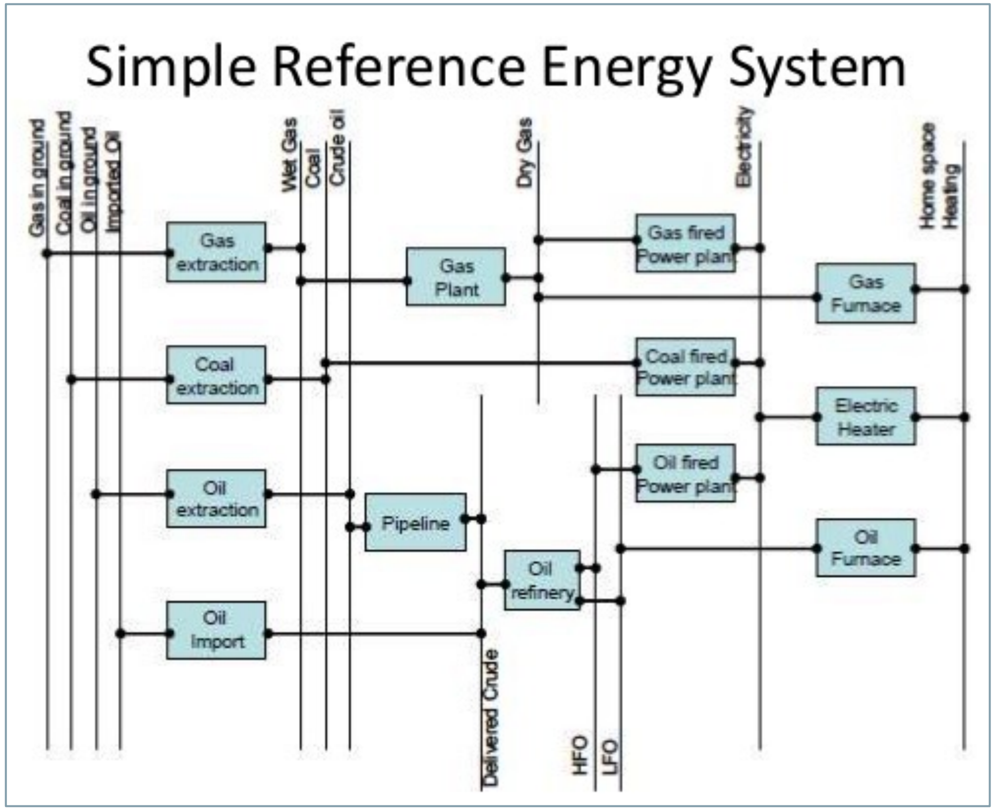


Figure E8.1. Pareto curves.

Reference energy system

- A common way of conceptualizing structure of energy optimization models
- A network description of an energy system showing connections between energy producers and consumers
  - How does energy flow from primary resources to consumers?
  - What provides energy to what, and with what transformations?
- Lends itself to bottom-up, technologically explicit modeling
- Provides a framework for defining technical and cost characteristics of energy production – a basis for optimization



Computation

Special-purpose software tools are used to specify models as a linear or non-linear programming problem (depending on linearity of objective and constraints)

- GAMS
- MATLAB
- Julia
- Or even Excel for simple models

Most tools work in concert with separate **"solvers," which are used to find the solution to a problem** (if there is one!). Some solvers are open source and free, such as:

- GLPK
- COIN-OR (Cbc/Clp)

Others are commercially available but are faster and capable of handling more difficult problems. Examples include:

- CPLEX
- Gurobi
- Mosek
- FICO Xpress

LEAP and NEMO work with all of the solvers above

Practical considerations

Data intensiveness

- Optimization requires considerably more data than some other approaches to energy modeling (e.g., trends analysis, professional judgement)
  - Particularly costs: capital, O&M, etc.
  - Current and projected values needed
- Requirements can be lessened with a more aggregate model structure, but this risks eliding important system dynamics

Practical considerations

Sensitivity and plausibility

- Many optimization models are quite sensitive to their parameters (exogenous inputs)
- Their initial tendency is to produce a corner solution – e.g., building only one technology
- This compels modeler to add constraints to attain a plausible result
- The content of these iteratively determined constraints is critically important – in many ways, it decides the outcome!

Practical considerations

Complexity and performance

- Optimization modeling is computationally intense
- There is a significant trade-off between model specificity / resolution / complexity and run time
  - Important drivers: numbers of dimensions (e.g., years, technologies) and constraints
- Key points to keep in mind:
  - What questions is the modeling trying to answer?
  - What dynamics affect these questions?
- Long-term and short-term models typically have very different resolutions

NEMO: Next Energy Modeling system for Optimization

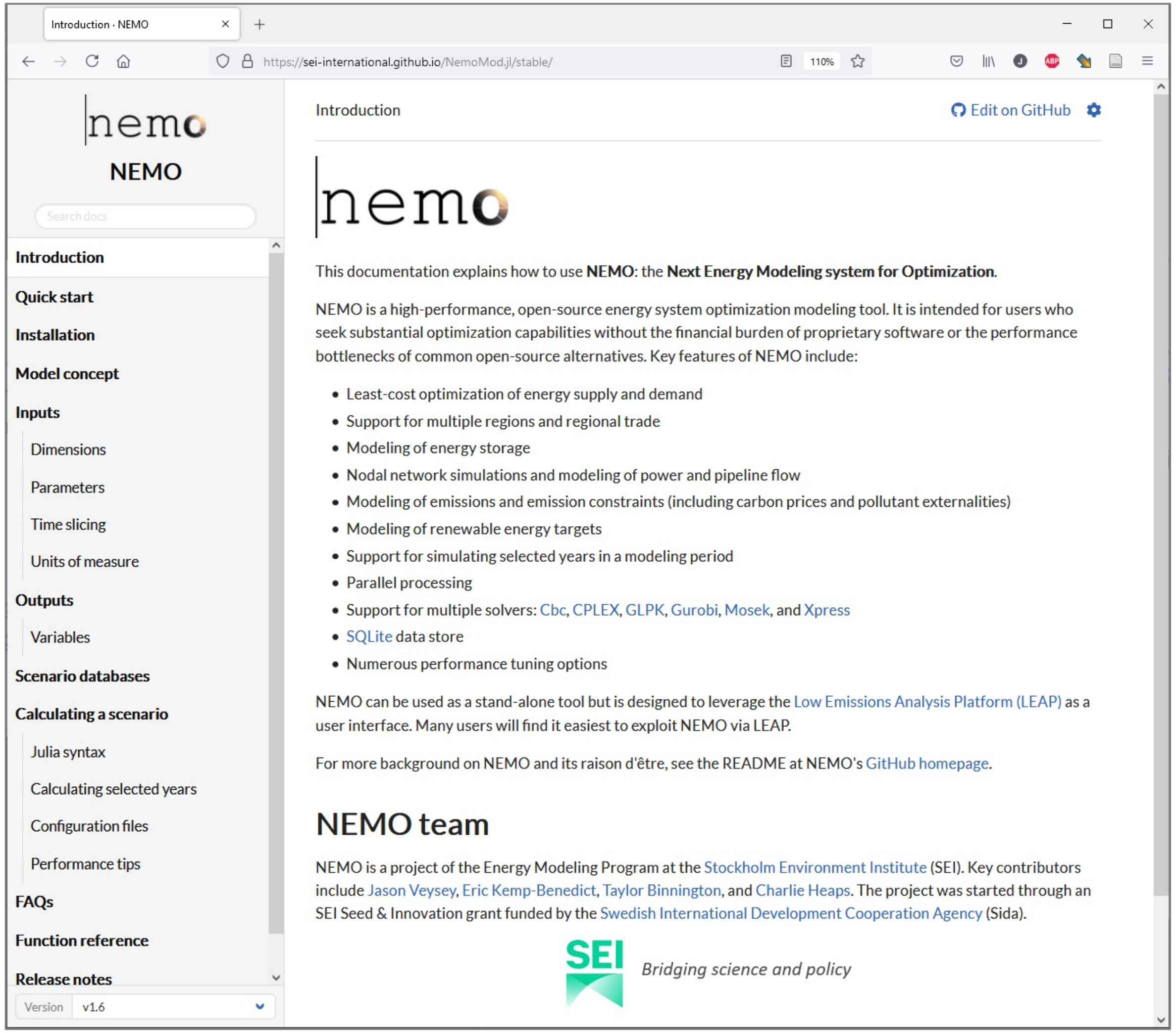
- High performance, open source energy system optimization tool
- Integrates with LEAP as a graphical user interface
- Key features for decarbonization and electricity system analyses
  - Energy storage
  - Nodal networks, power and pipeline flow
  - Emission and renewable energy targets
  - Carbon/pollutant pricing
  - Regional modeling and energy trade
- Parallel processing
- Support for simulating selected years in a modeling period
- Compatible with multiple solvers; numerous performance tuning options

nemo

https://leap.sei.org/nemo

Code and documentation

- NEMO is programmed in Julia, a high performance language for mathematical computing
- Code available at: https://github.com/sei-international/NemoMod.jl
- Full documentation online at: https://sei-international.github.io/NemoMod.jl/stable/

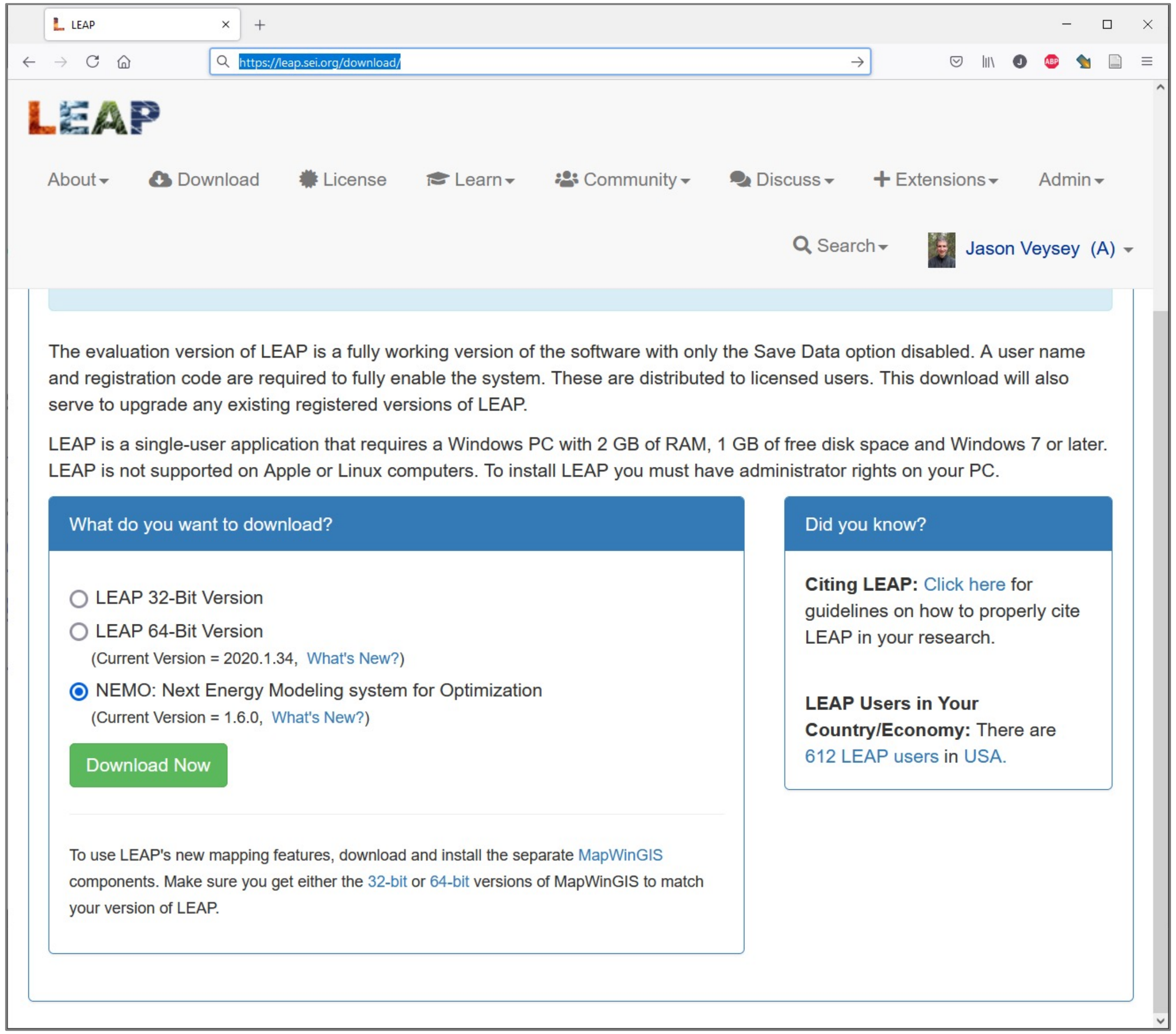


Prerequisites

- NEMO can run on any system compatible with Julia: Windows, macOS, Linux, FreeBSD
- To use NEMO with LEAP, a Windows computer is necessary
- Hardware requirements depend on model size; for most models, following specifications are advisable:
  - Multi-core processor
  - At least 8 GB RAM
  - At least 500 MB free disk space
- Administrator privileges facilitate installation

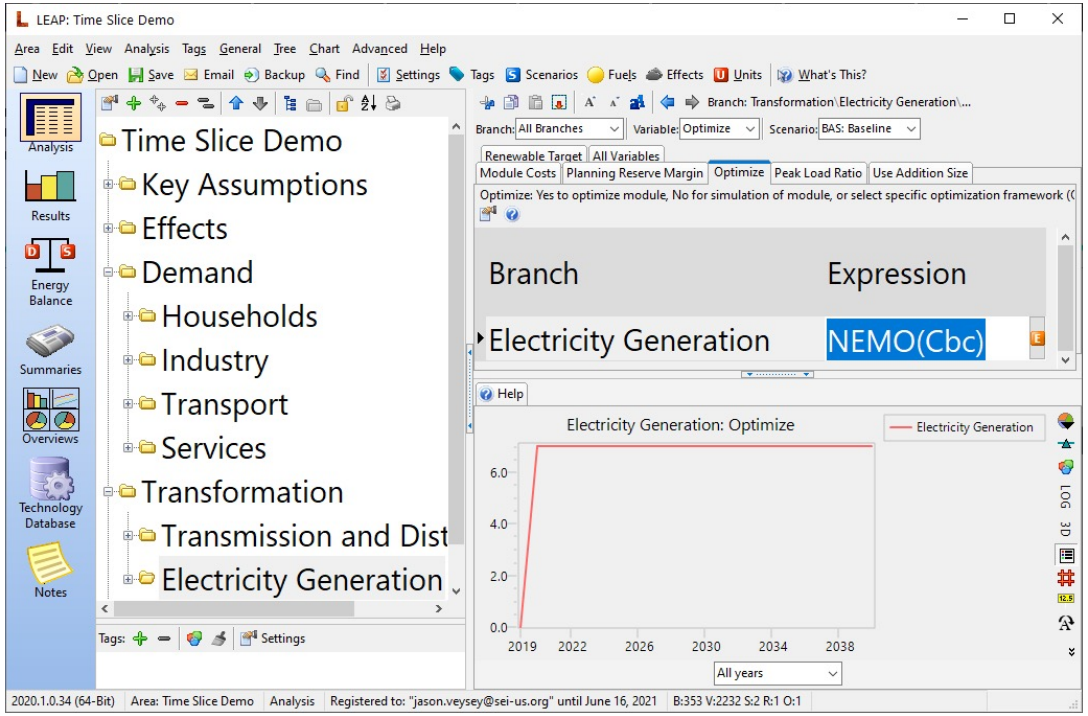
Installation

- For best performance with LEAP, install NEMO with installer program on LEAP website (https://leap.sei.org/download/)
- Can also install from source code – directions at https://sei-international.github.io/NemoMod.jl/stable/installation/



Enabling optimization in LEAP

- At present, LEAP can optimize one transformation module (supply sector) in a model
- Module-level Optimize variable activates optimization with NEMO and selects solver
  - Optimize only available in projection scenarios
- Costs and capacities must be enabled in optimized module, and process costs (including fuel costs) and performance parameters must be populated
- Optimized results are calculated and displayed when Results View is selected



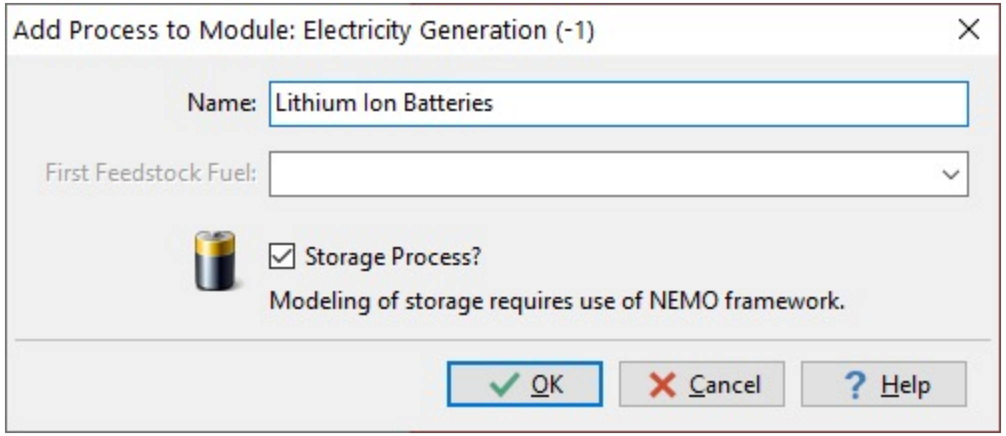
Additional variables in optimized modules

- LEAP provides a number of additional variables in Analysis View for optimized modules
- These allow specification of various constraints on optimization, e.g.:
  - Renewable energy production targets
  - Minimum and maximum capacity and capacity additions
  - Discrete addition sizes for new capacity
  - Operating rules for storage

Carefully review variables and documentation (help)
when calibrating an optimization model

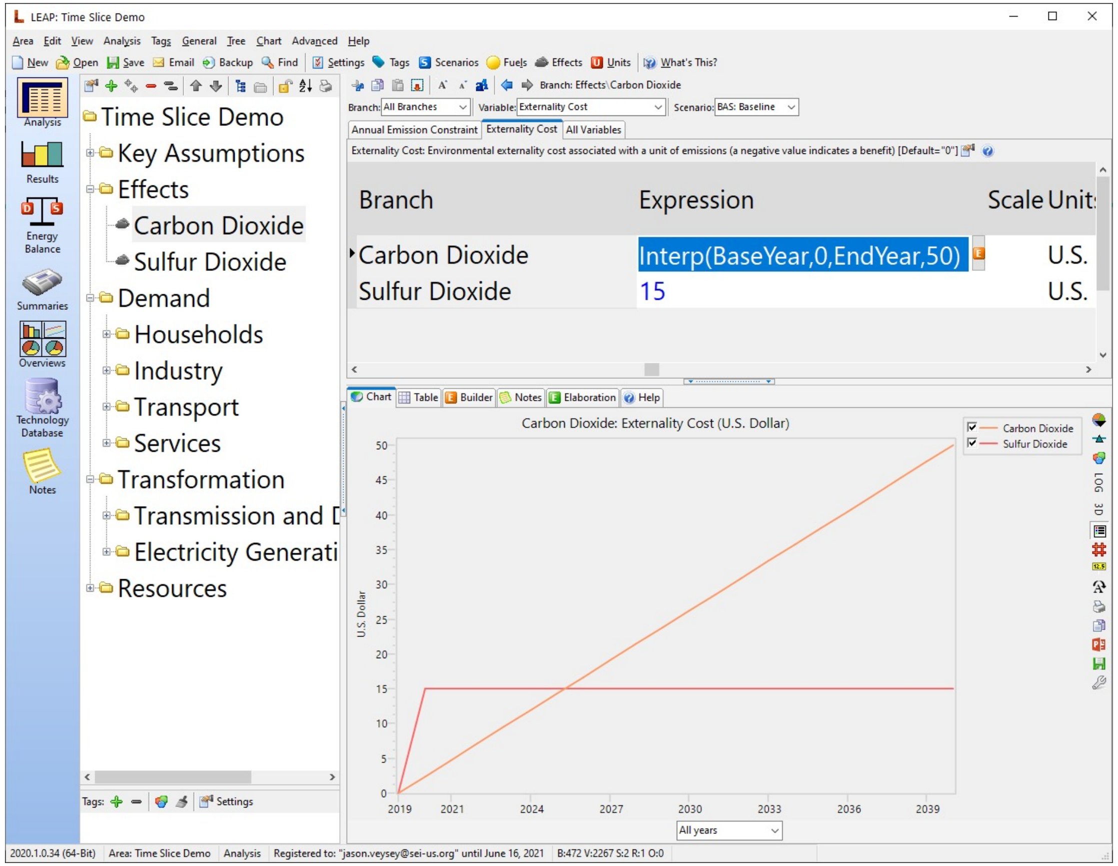
Modeling energy storage

- When used with NEMO, LEAP can simulate energy storage
- In an optimized module, processes can be identified as storage when they are created
- In addition to normal process variables, LEAP displays a number of storage-specific variables used to control simulation in NEMO
  - Starting Charge
  - Minimum Charge
  - Full Load Hours
  - Energy carryover rules



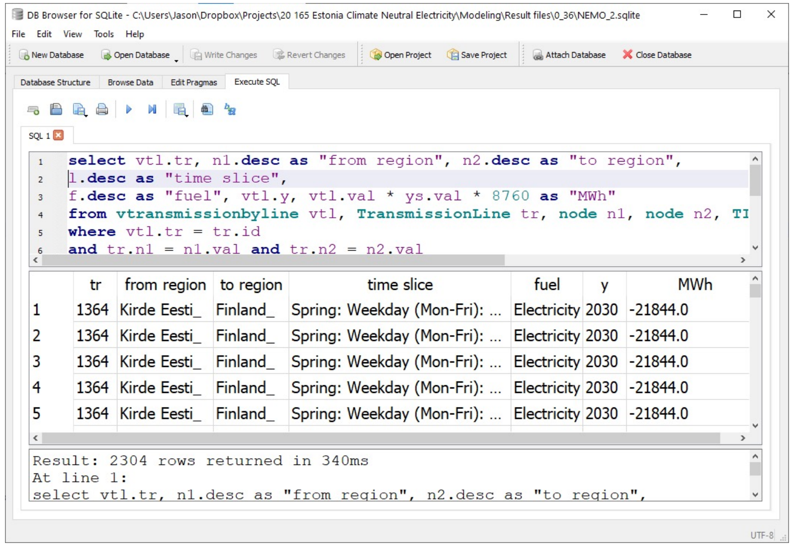
Emission costs and constraints

- Emission costs and limits can be included in LEAP-NEMO optimization
- Activation in Settings
  - Emission costs
    - Scope & Scale => Energy Effects
    - Scope & Scale => Costs
    - Costs => Environmental externality costs
  - Emission limits
    - Scope & Scale => Energy Effects
    - Optimization => Enable Emissions Constraints
- Costs and limits to use in a scenario are then added under Effects branch



Additional NEMO results

- LEAP's Results View displays a small set of NEMO outputs – capacity expansion and dispatch
- Other outputs can be accessed in NEMO scenario databases, which LEAP creates when calculating optimized scenarios
- NEMO scenario databases are in SQLite format
- Scenario databases can be saved in LEAP areas repository when saving a model: Settings => Optimization => Keep Intermediate Results
- Useful tool for working with scenario databases: DB Browser for SQLite (https://sqlitebrowser.org/)
- Information on scenario database structure and available outputs in NEMO documentation:
    - https://sei-international.github.io/NemoMod.jl/stable/scenario_db/
    - https://sei-international.github.io/NemoMod.jl/stable/variables/



NEMO configuration files

- When using NEMO with LEAP, users can modify NEMO's run-time settings for a model by adding a configuration file named "nemo.cfg" to the model's folder in the LEAP areas repository
- Configuration files can be used to specify solver parameters, tell NEMO to save additional output variables, run custom Julia scripts, and more
- Settings in a configuration file overwrite run-time defaults LEAP provides to NEMO
- NEMO documentation contains a full list of options
  - https://sei-international.github.io/NemoMod.jl/stable/configuration_file/

nemo.cfg - Notepad

File Edit Format View Help

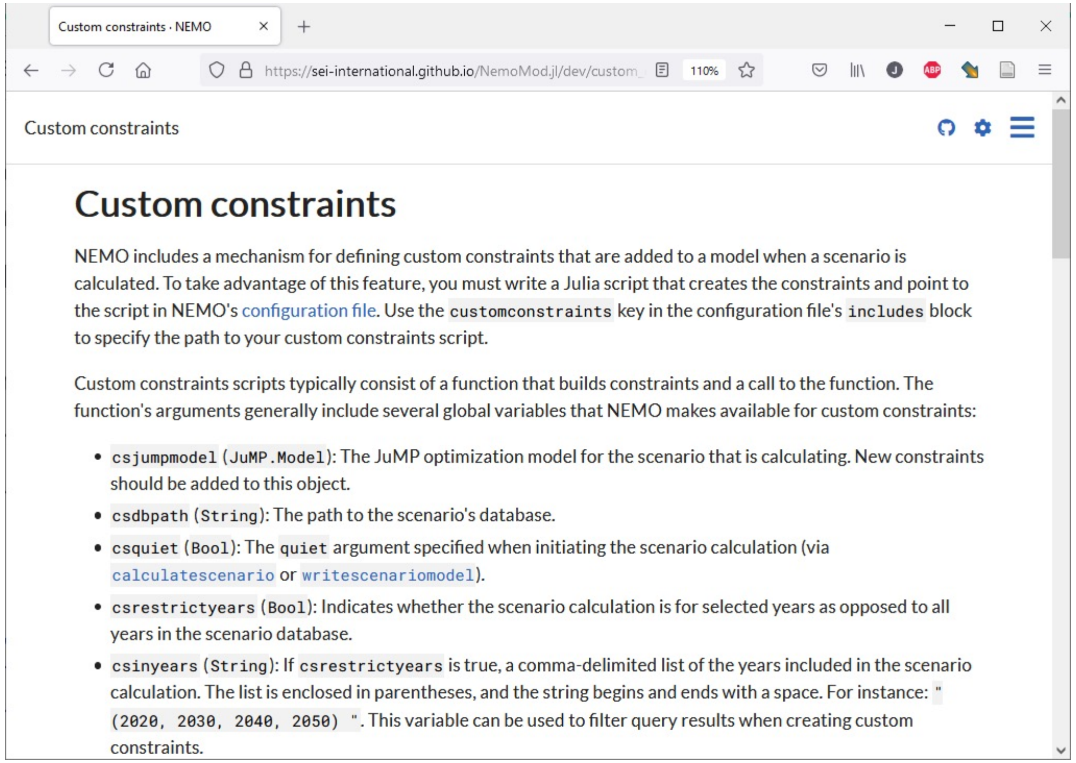
```
; Configuration file for NEMO

; Currently supported sections and keys:
;
; [calculatescenarioargs]
; calcyears=Comma-delimited list of years to include in scenario calculation
; varstosave=Comma-delimited list of model output variables to save
; numprocs=Number of Julia processes to use for parallelized operations (a positive integer or 0 for half the number of logical processors on the executing machine)
; targetprocs=List of target process numbers
; restrictvars=true or false
; reportzeros=true or false
; continuoustransmission=true or false
; forcemip=true or false
; quiet=true or false
;
; [solver]
; parameters=Comma-delimited list of solver parameters and values in the form parameter1=value1,parameter2=value2
;
; [includes]
; beforescenariocalc=path
; customconstraints=path
;
; Notes
; • NEMO searches for this file (nemo.ini or nemo.cfg) in the Julia working directory.
;
; • Values for keys in [calculatescenarioargs] are string representations of arguments to pass to calculatescenario().
;              See the documentation for calculatescenario() for more information.
; • calcyears overrides calcyears specified on command line.
; • varstosave adds to varstosave specified on command line.
; • numprocs overrides numprocs specified on command line.
; • targetprocs adds to targetprocs specified on command line.
; • restrictvars overrides restrictvars specified on command line.
; • reportzeros overrides reportzeros specified on command line.
; • continuoustransmission overrides continuoustransmission specified on command line.
; • forcemip overrides forcemip specified on command line.
; • quiet overrides quiet specified on command line.
;
; • NEMO infers types of values specified in [solver] parameters using Julia's tryparse function. Values are tested to see if they parse to Bool, Int, and Float64 types. Values that do not parse to any of these types are passed to the solver as strings.
;
; • beforescenariocalc defines a path to a Julia file to be included before each
;              scenario calculation in NEMO (i.e., before each execution of calculatescenario()).
;              It is optional.
; • customconstraints defines a path to a Julia file to be included when building
;              constraints during a scenario calculation in NEMO. It is optional.
; • All paths should be defined relative to the current working directory.
[calculatescenarioargs]
continuoustransmission=true
varstosave=vtransmissionannual,vtransmissionbylineannual,vtransmissionexists,vtransmissionbuilt,vtransmissionbyline,vtotaltechnol
```

Custom NEMO constraints

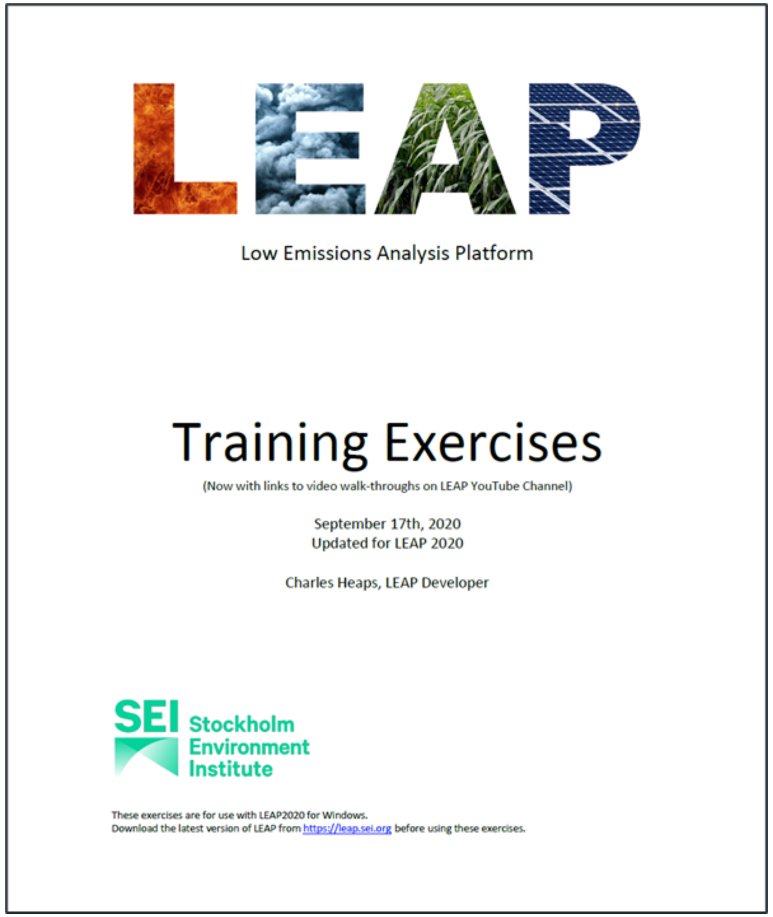
- Custom constraints can be added to a LEAP-NEMO model by writing a Julia script that defines the constraints
- Script is referenced in a NEMO configuration file and loaded when NEMO calculates a scenario
- Custom constraints can be used for a wide range of purposes:
  - Share resource limits among multiple processes
  - Link capacity additions or retirements across processes
  - Create supplemental constraints for ensuring system reliability

  …
- Detailed instructions at: https://sei-international.github.io/NemoMod.jl/dev/custom_constraints/



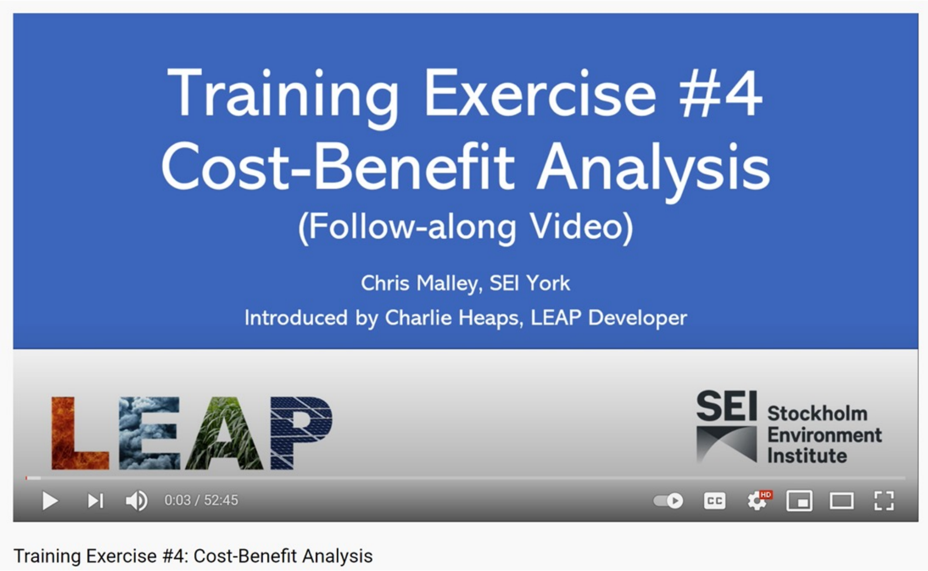
Exercise 3: Cost-benefit analysis

Freedonia



exercise_1_2_3 - LEAPTrainingExerciseEnglish2020.pdf

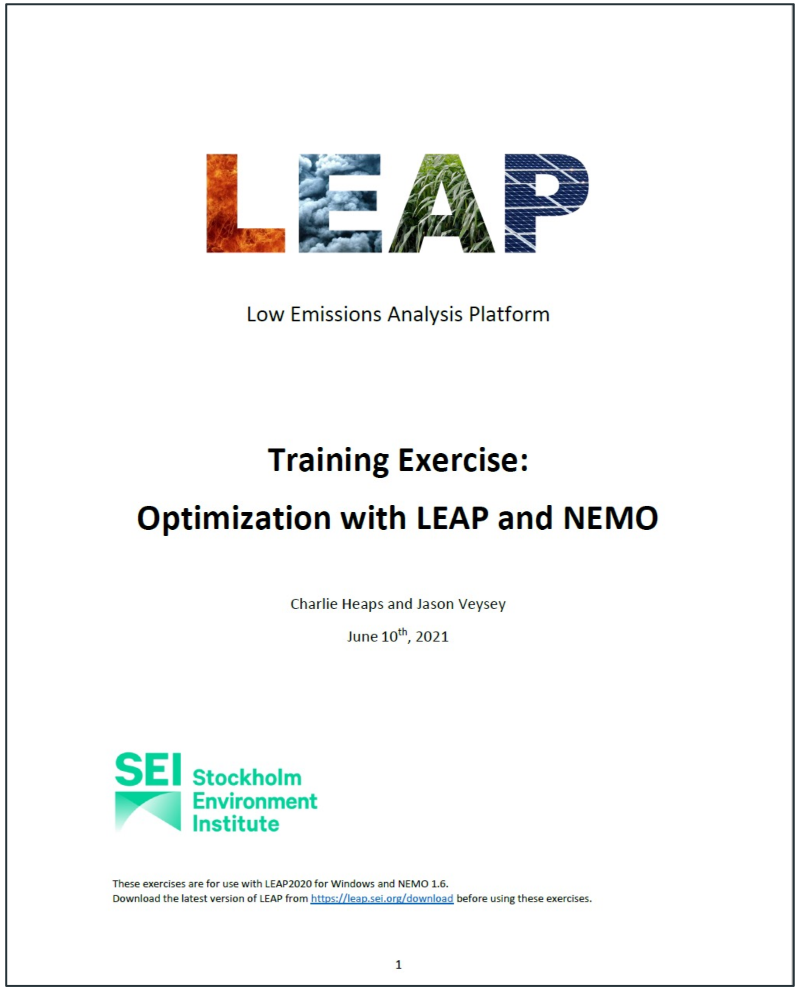
Workshop Exercise 3: Chapter 4 in Training Exercises document



https://youtu.be/au99KzpHmio

Exercise 4: Optimization modeling

Optimization Exercise



LEAP

Low Emissions Analysis Platform

Training Exercise:

Optimization with LEAP and NEMO

Charlie Heaps and Jason Veysey

June 10th, 2021

SEI Stockholm Environment Institute

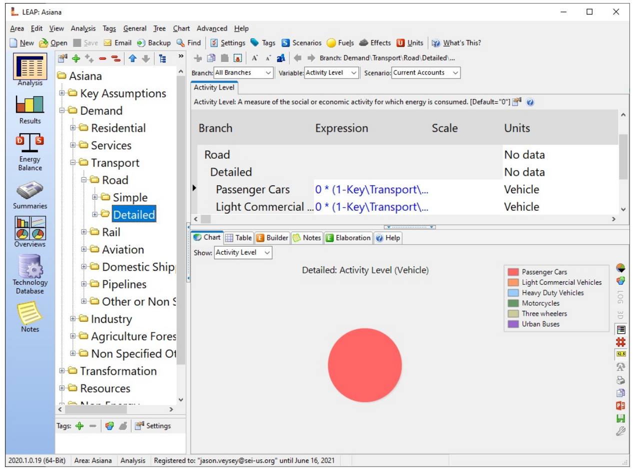
These exercises are for use with LEAP2020 for Windows and NEMO 1.6.
Download the latest version of LEAP from https://leap.sei.org/download before using these exercises.

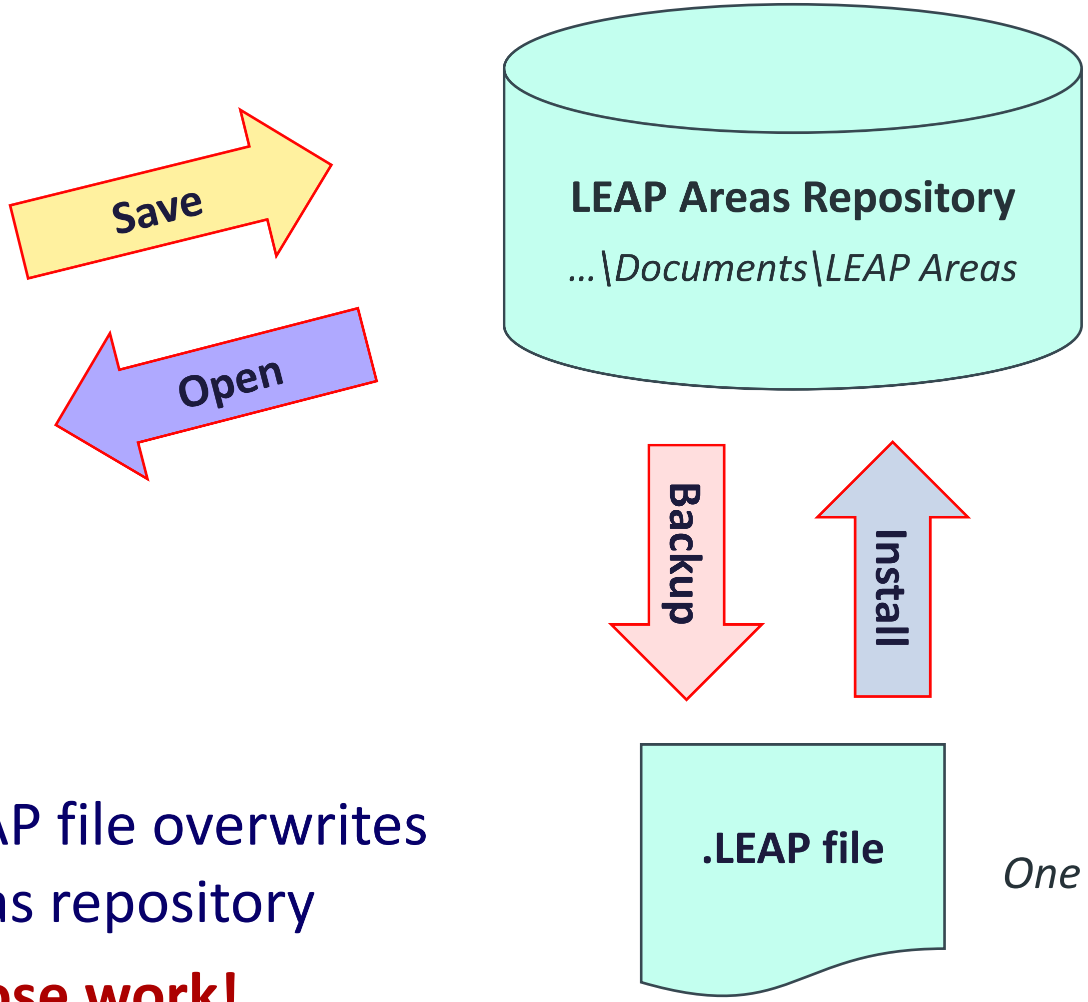
1

exercise_4_optimization.pdf

- Optimization Exercise area installed with LEAP
- Supporting Excel file – Yearly Shape Data.xlsx – in shared materials folder

Saving and sharing models





One folder with multiple files per area

One zipped file per area

Installing an area from a .LEAP file overwrites what's in local LEAP areas repository

Be careful, you can lose work!