GLOBAL PST CONSORTIUM

LF ENERGY

Open-Source Power Systems Analysis Packages: Cross-package Coordination for System Planning

Special Interest Group on Open-source Power System Planning Tools
June 22, 2022

Zoom Housekeeping

- This webinar is **being recorded** and will be shared with attendees.
- You will be **automatically muted** upon joining and throughout the webinar.
- Please use the **chat feature** to add comments and share input.
- Please use the **Q&A function** to ask questions and ensure your question is answered. You can find this function in your toolbar.
- If you have **technical issues**, please use the chat feature to message Emily Klos.
- You can adjust your audio through the **audio settings.** If you are having issues, you can also dial-in and listen by phone. Dial-in information can be found in your registration email.
- There is a **short survey** that will launch at the end of the webinar. Your survey responses help us tailor training content, improve your webinar experience, and gain support to host future trainings.

G-PST and LF Energy Overview

Juha Kiviluoma

G-PST Pillar 5 Lead, **VTT Finland**

Benoît Jeanson

Enterprise Architect, Open-Source Program Office of RTE (French TSO), **RTE France**

Contributors

Roman Bolgaryn
Researcher
Fraunhofer IEE

Shuli Goodman
Executive Director
LF Energy

Nicolas Omont
Vice President of Operations
Artelys

Hannele Holttinen
G-PST Pillar 5 Lead
Recognis Consultnig

Clayton Barrows
Senior Engineer
NREL

Maximilian Parzen
Director / PhD Candidate
PyPSA meets Earth Initiative / University of Edinburgh

Dheepak Krishnamurthy
Distribution Systems Researcher
NREL

Benoît Jeanson
Enterprise Architect, Open-Source Program Office of RTE (French TSO)
RTE France

Juha Kiviluoma
G-PST Pillar 5 Lead
VTT, Finland

GLOBAL PST CONSORTIUM

nationalgridESO

NREL Transforming ENERGY

IEEE Advancing Technology for Humanity

VTT

AEMO AUSTRALIAN ENERGY MARKET OPERATOR

CSIRO

EPRI ELECTRIC POWER RESEARCH INSTITUTE

EIRGRID GROUP

Imperial College London

Fraunhofer CINES

DTU

olade Organización Latinoamericana de Energía

ercot

ESIG ENERGY SYSTEMS INTEGRATION GROUP

California ISO

CSIR Touching lives through innovation

ENERGINET

ASEAN CENTRE FOR ENERGY ACE

Global Power System Transformation Consortium advances action in 5 key areas

1. System Operator Research & Peer Learning

Perform cutting edge applied research to create novel system operator solutions and globally disseminate and infuse new insights through peer learning

2. System Operator Technical Assistance

Provide implementation support to scale established best practice engineering and operational solutions

3. Foundational Workforce Development

Build the inclusive and diverse workforce of tomorrow through enhanced university curriculum and technical upskilling for utility and system operator staff

4. Localized Technology Adoption Support

Adapt modern power system technologies to individual country contexts through testing programs and standards development activities

5. Open Data and Tools

Support rigorous planning, operational analysis and enhanced real-time system monitoring through open data and tools

CORE TEAM – All Core Team members contribute to all activity pillars

REGIONAL LEADS – Coordinate regional peer learning networks and country-level TA delivery efforts for Africa, Asia, and Latin America and the Caribbean

INTERIM SECRETARIAT – Work program coordination, partnerships and support, outreach, etc.

LF ENERGY

Providing a 21st century plan of action to decarbonization through open source, open frameworks, reference architectures, and a support ecosystem of complementary projects.

LF Energy Projects (+1)

LF ENERGY FlexMeasures

LF ENERGY OpenSTEF

LF ENERGY FLEDGE POWER

LF ENERGY EVerest

LF ENERGY CARBON DATA SPECIFICATION CONSORTIUM

LF ENERGY SOGNO

LF ENERGY OpenGEH

LF ENERGY SHAPESHIFTER

LF ENERGY GRID EXCHANGE FABRIC

LF ENERGY OPERATORFABRIC

LF ENERGY CoMPAS

LF ENERGY openLEADR

LF ENERGY Hyphae

LF ENERGY GRID CAPACITY MAP

LF ENERGY POWSYBL

LF ENERGY SEAPATH

LF ENERGY OPENEEMETER

LF Energy Members

Strategic

- Alliander
- Google
- Microsoft
- Rte — Le réseau de transport d'électricité

General

- AVEVA
- BTP
- Cleartrace
- CloudBees.
- electricityMap
- ENERGINET
- FLEXIDAO
- FUTUREWEI Technologies
- GE
- HITACHI Inspire the Next
- indra
- M-RETS
- MISSION:DATA empowering energy savings
- PIONIX
- RECURVE
- Savoir-faire LINUX
- REPLY SENSE
- Statnett
- TenneT
- UTILITYAPI
- VATTENFALL
- WattCarbon
- ZEDEDA

Associate

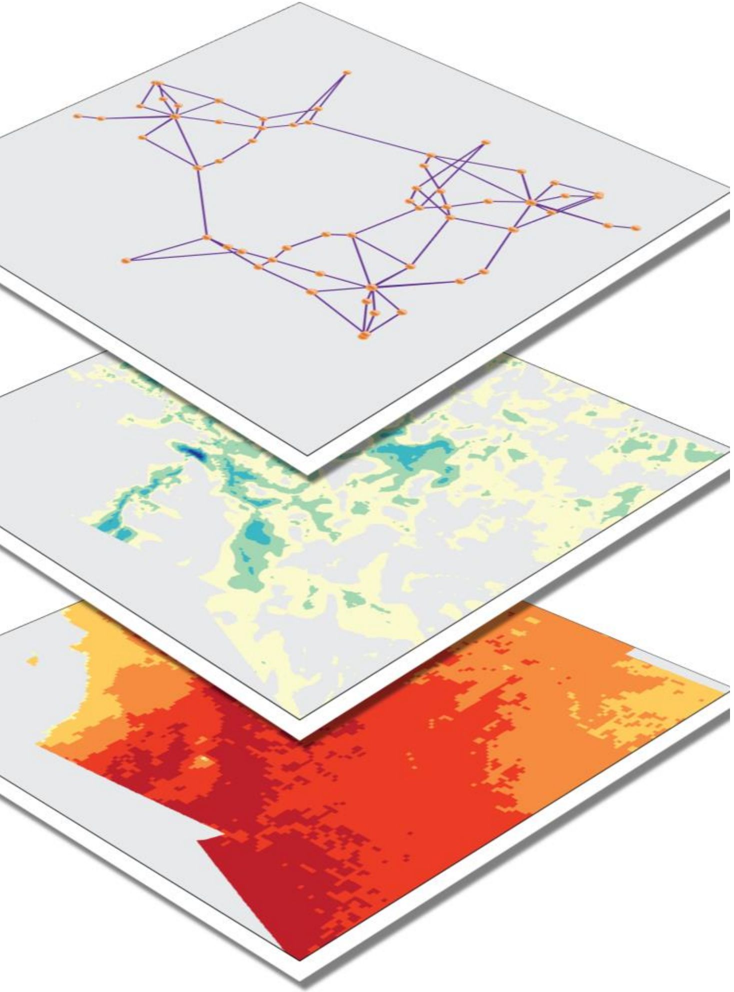
- Carbon Co-op
- TU Delft
- ElaadNL
- EPRI
- ENACCESS
- Energy Foundation
- EnergyAi.Berlin
- entsoe
- EPFL
- Fraunhofer IEE
- FREEDM Systems Center
- FAU
- Iowa State University
- KTH
- Hochschule
- Monash University
- NREL
- OpenEarth Foundation
- openADR Alliance
- OpenUK
- Pecan Street
- Project Haystack
- RWTH Aachen University
- Stanford
- The Alan Turing Institute
- the Energy Coalition
- Universität Kassel
- Vanderbilt University
- Washington State University

Agenda

- Introduction
- Workflow and Tool Demonstrations
- Outcomes and Lessons

Contributors

- G-PST
  - Hannele Holttinen – Recognis
  - Juha Kiviluoma – VTT
- PyPSA
  - Max Parzen – University of Edinburgh
- pandapower
  - Roman Bolgaryn – Fraunhofer IEE
  - Sadia Ferdous Snigdha – Fraunhofer IEE & TU Ilmenau
- PowSybl
  - Benoit Jeanson – RTE
  - Nicolas Omont – Artelys
- PowerSimulations.jl
  - Clayton Barrows – NREL
  - Dheepak Krishnamurthy – NREL



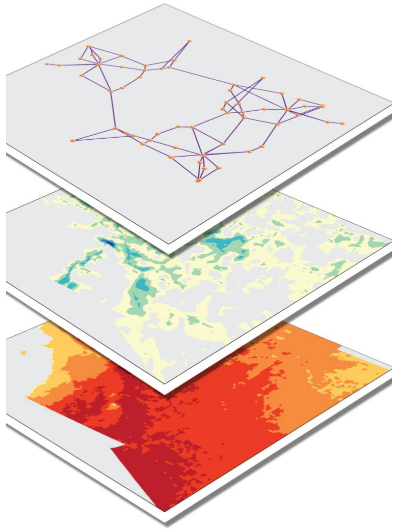
Data

- RTS-GMLC (github.com/gridmod/rts-gmlc)
  - 73 buses, 120 branches, and 115 generators

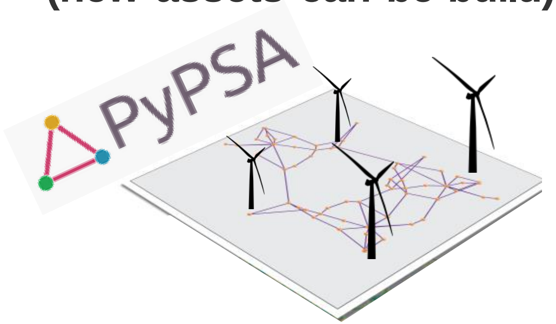
Problem and Workflow Specification

- Event 2: System expansion planning and analysis

Existing RTS-GMLC



New **Capacity expansion (new assets can be build)**



- Perform invest and dispatch *co-optimization*
- Add investment costs
- Add constraints (i.e.CO2)
- Add load scenario

Operational analysis (PyPSA results valid?)

pandapower

LF ENERGY POWSYBL

PowerSystems.jl
PowerSimulations.jl

Value Proposition

- Demonstration of open-source power systems modeling capabilities
- Coordination:
  - Utilize the strengths of multiple tools to perform a diverse set of analysis
  - Standardize data specifications, enhance/build tool interoperability

Workflow Step Demonstrations

Maximilian Parzen, Co-steering the PyPSA meets Earth initiative

PyPSA

PyPSA

Official website:

https://pypsa.org/

GitHub repository:

https://github.com/PyPSA

Documentation:

https://pypsa.readthedocs.io/en/latest/

What is PyPSA?

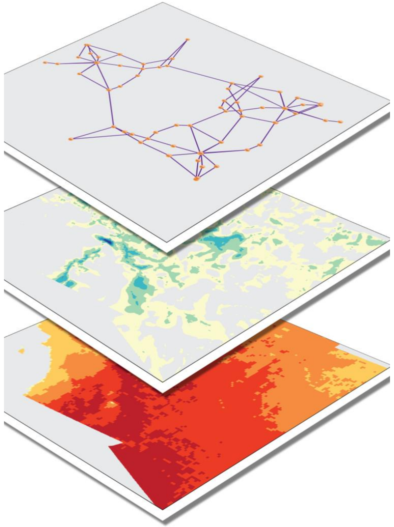
Purpose:

- A tool that can do both **economic** and **grid analysis**
- Developed for **large-scale optimization** and
- Studies in **high spatial resolution**

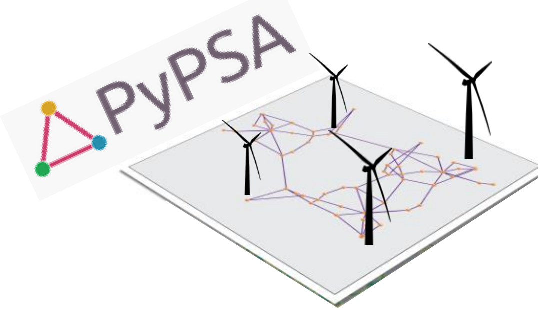
					Grid Analysis			Economic Analysis							
	Software	Version	Citation	Free Software	Power Flow	Continuation Power Flow	Dynamic Analysis	Transport Model	Linear OPF	SCLOPF	Nonlinear OPF	Multi-Period Optimisation	Unit Commitment	Investment Optimisation	Other Energy Sectors
Power system tools	MATPOWER	6.0	[6]	✓	✓	✓		✓	✓		✓				
	NEPLAN	5.5.8	[2]		✓		✓	✓	✓	✓	✓				✓
	pandapower	1.4.0	[9]	✓	✓			✓	✓		✓				
	PowerFactory	2017	[1]		✓		✓		✓	✓	✓				
	PowerWorld	19	[3]		✓		✓	✓	✓	✓	✓				
	PSAT	2.1.10	[7]	✓	✓	✓	✓		✓		✓	✓	✓		
	PSS/E	33.10	[4]		✓		✓		✓	✓	✓				
	PSS/SINCAL	13.5	[5]		✓		✓				✓				✓
	PYPOWER	5.1.2	[8]	✓	✓			✓	✓		✓				
	PyPSA	0.11.0		✓	✓			✓	✓	✓		✓	✓	✓	✓
Energy system tools	calliope	0.5.2	[11]	✓				✓				✓		✓	✓
	minpower	4.3.10	[12]	✓				✓	✓			✓	✓		
	MOST	6.0	[13]	✓	✓	✓		✓	✓	✓	✓	✓	✓		
	oemof	0.1.4	[14]	✓				✓				✓	✓	✓	✓
	OSeMOSYS	2017	[15]	✓				✓				✓		✓	✓
	PLEXOS	7.400	[16]					✓	✓	✓		✓	✓	✓	✓
	PowerGAMA	1.1	[17]	✓				✓	✓			✓			
	PRIMES	2017	[18]					✓	✓			✓	✓	✓	✓
	TIMES	2017	[19]					✓	✓			✓	✓	✓	✓
	urbs	0.7	[20]	✓				✓				✓	✓	✓	✓

Planning exercise

Existing RTS-GMLC



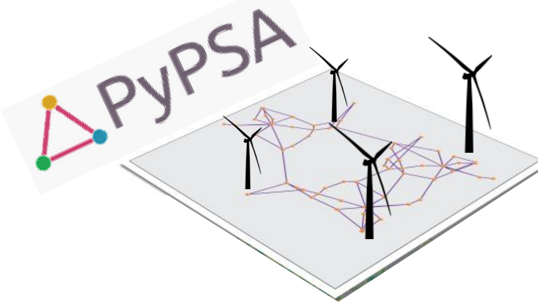
New **Capacity expansion (new assets can be built)**



- Add data for 1 year!
- Add investment costs
- Add constraints (i.e.CO2)
- Add load scenario

Planning exercise

New **Capacity expansion (new assets can be built)**

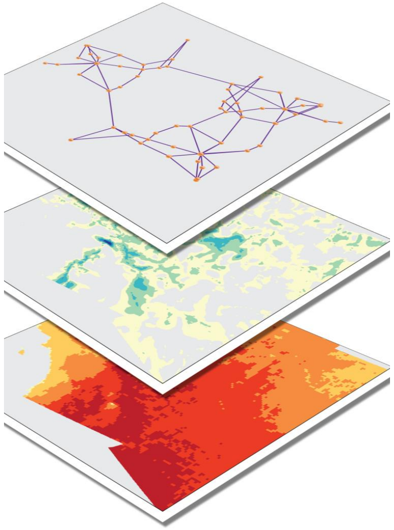


5 SCENARIOS:

```
d = {
    "scenario": [
    "RTS_GMLC_base", # no expansion but opf
    "RTS_GMLC_base+line_expansion",  # line expansion and opf
    "RTS_GMLS_base+gen_expansion",  # generation expansion and constraints, a
    "RTS_GMLS_base+gen_and_line_expansion",  # generation expansion and const
    "RTS_GMLS_1p5xload+0emission+gen_and_line_expansion",  # generation expa
    ],
```

Planning exercise

Existing RTS-GMLC



Pandapower import of Matpower

```
_sets_path_to_root("power-flow-exercise")
net=load_rts_grid()
# and convert to pypsa
network=convert_to_pypsa(net)
n = network
```

Required some renaming:

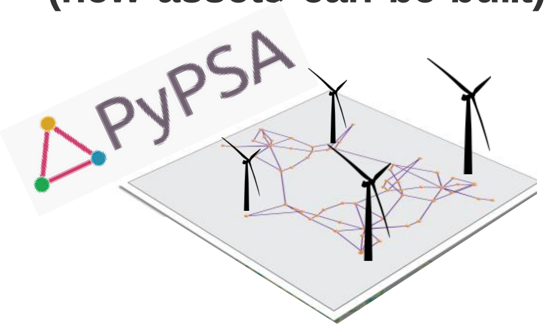
prepare pandapower network for pypsa

```
net.gen.loc[:, "fuel"] = net.gen["fuel"].replace({
        "Oil": "oil",
        "Coal": "coal",
        "Nuclear": "nuclear",
        "Hydro": "hydro",
    })

ccgt_condition = (net.gen["fuel"]=="NG") & (net.gen["type"]=="CC")
ocgt_condition = (net.gen["fuel"]=="NG") & (net.gen["type"]=="CT")
sync_condition = (net.gen["fuel"]=="Sync_Cond")
net.gen.loc[ccgt_condition, "fuel"] = net.gen.loc[ccgt_condition, "fuel"].replace({"NG": "CCGT",})
net.gen.loc[ocgt_condition, "fuel"] = net.gen.loc[ocgt_condition, "fuel"].replace({"NG": "OCGT",})
net.gen = net.gen.drop(net.gen[sync_condition].index)  # remove sync_cond
```

```
net.sgen.loc[:, "fuel"] = net.sgen["fuel"].replace({
        "Oil": "oil",
        "Coal": "coal",
        "Nuclear": "nuclear",
        "Hydro": "hydro",
        "Solar": "solar",
        "Wind": "onwind",
    })

ccgt_condition = (net.sgen["fuel"]=="NG") & (net.sgen["type"]=="CC")
ocgt_condition = (net.sgen["fuel"]=="NG") & (net.sgen["type"]=="CT")
storage_condition = (net.sgen["fuel"]=="Storage")

net.sgen.loc[ccgt_condition, "fuel"] = net.sgen.loc[ccgt_condition, "fuel"].replace({"NG": "CCGT",})
net.sgen.loc[ocgt_condition, "fuel"] = net.sgen.loc[ocgt_condition, "fuel"].replace({"NG": "OCGT",})
net.sgen = net.sgen.drop(net.sgen[storage_condition].index)  # remove storage
```

Planning exercise

New **Capacity expansion (new assets can be built)**



- **Add data for 1 year!**
- Add investment costs
- Add constraints (i.e. CO_2)
- Solve

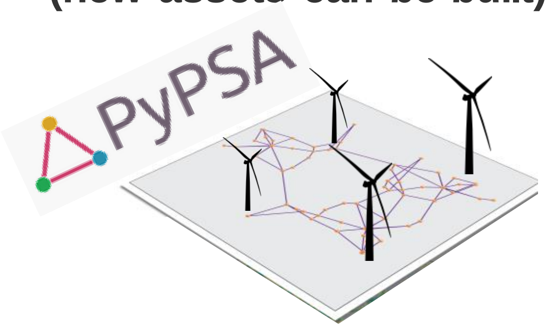
Load and prepare time-series (for all tech)

```python
load_path = os.path.join(os.getcwd(), "example-pypsa/timeseries_files/Load/bus_load.csv")
utpv_path = os.path.join(os.getcwd(), "example-pypsa/timeseries_files/PV/DAY_AHEAD_pv.csv")
rtpv_path = os.path.join(os.getcwd(), "example-pypsa/timeseries_files/RTPV/DAY_AHEAD_rtpv.csv")
wind_path = os.path.join(os.getcwd(), "example-pypsa/timeseries_files/Wind/DAY_AHEAD_wind.csv")
hydro_path = os.path.join(os.getcwd(), "example-pypsa/timeseries_files/Hydro/DAY_AHEAD_hydro.csv")


# In[16]:


utpv_series = pd.read_csv(utpv_path)
utpv_series.rename(columns={"Period": "Hour"}, errors="raise",inplace=True)
utpv_series.index = pd.to_datetime(utpv_series[['Year', 'Month', 'Day','Hour']])
utpv_series = utpv_series.drop(columns=['Year', 'Month', 'Day','Hour'])
utpv_series_pu = utpv_series/utpv_series.max()
utpv_series_max_potential = utpv_series.max() * res_scale


# In[17]:


rtpv_series = pd.read_csv(rtpv_path)
rtpv_series.rename(columns={"Period": "Hour"}, errors="raise",inplace=True)
rtpv_series.index = pd.to_datetime(rtpv_series[['Year', 'Month', 'Day','Hour']])
rtpv_series = rtpv_series.drop(columns=['Year', 'Month', 'Day','Hour'])
rtpv_series_pu = rtpv_series/rtpv_series.max()
rtpv_series_max_potential = rtpv_series.max() * res_scale
```

```python
wind_series = pd.read_csv(wind_path)
wind_series.rename(columns={"Period": "Hour"}, errors="raise",inplace=True)
wind_series.index = pd.to_datetime(wind_series[['Year', 'Month', 'Day','Hour']])
wind_series = wind_series.drop(columns=['Year', 'Month', 'Day','Hour'])
wind_series_pu = wind_series/wind_series.max()
wind_series_max_potential = wind_series.max() * res_scale


# In[19]:


hydro_series = pd.read_csv(hydro_path)
hydro_series.rename(columns={"Period": "Hour"}, errors="raise",inplace=True)
hydro_series.index = pd.to_datetime(hydro_series[['Year', 'Month', 'Day','Hour']])
hydro_series = hydro_series.drop(columns=['Year', 'Month', 'Day','Hour'])
hydro_series_pu = hydro_series/hydro_series.max()
hydro_series_max_potential = hydro_series.max() * res_scale


# In[20]:


load_series = pd.read_csv(load_path)
load_series["DateTime"] = pd.to_datetime(load_series["DateTime"])
load_series.set_index("DateTime", inplace=True)
load_series = load_series * load_scale
load_series.columns = [element.upper() for element in load_series.columns]
```

Planning exercise

New **Capacity expansion (new assets can be built)**



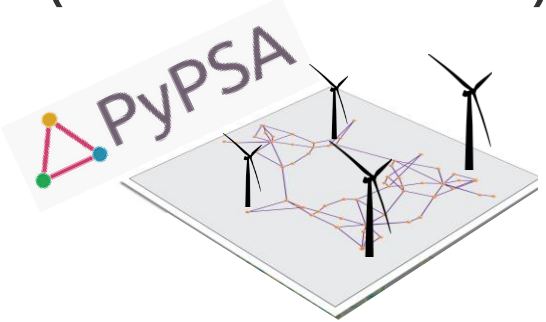
- **Add data for 1 year!**
- Add investment costs
- Add constraints (i.e. CO_2)
- Solve

Add time-series to network

```
n.madd("Generator",
    wind_series_pu.columns,
    bus=wind_series_pu.columns,
    p_nom_extendable=True,
    p_max_pu=wind_series_pu,
    p_nom_max=wind_series_max_potential)
```

Planning exercise

New **Capacity expansion (new assets can be built)**

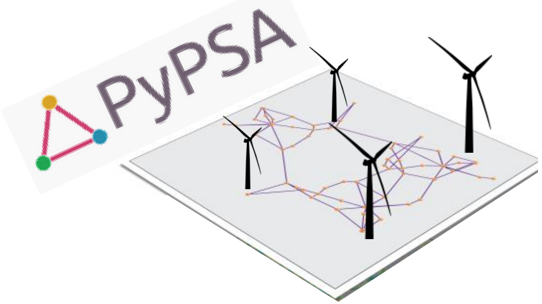


- Add data for 1 year!
- **Add investment costs**
- Add constraints (i.e.CO2)
- Solve

```
# Load costs and data modification

# In[8]:

costs = load_costs(Nyears=1., tech_costs=None, config=None, elec_config=None)

n.generators.loc[:,"capital_cost"] = n.generators["carrier"].map(costs.capital_cost)
n.generators.loc[:,"marginal_cost"] = n.generators["carrier"].map(costs.marginal_cost)
n.generators.loc[:,"lifetime"] = n.generators["carrier"].map(costs.lifetime)
n.generators.loc[:,"efficiency"] = n.generators["carrier"].map(costs.efficiency)
```

Planning exercise

New **Capacity expansion (new assets can be built)**

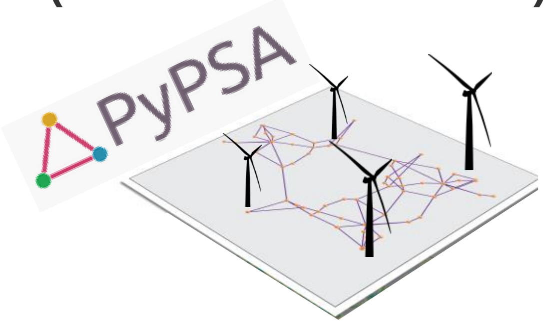


- Add data for 1 year!
- Add investment costs
- **Add constraints (i.e.CO2)**
- Solve

```python
def add_co2limit(n, co2limit, Nyears=1.):

    n.add("GlobalConstraint", "CO2Limit",
        carrier_attribute="co2_emissions", sense="<=",
        constant=co2limit * Nyears)
```

Planning exercise

New **Capacity expansion (new assets can be built)**

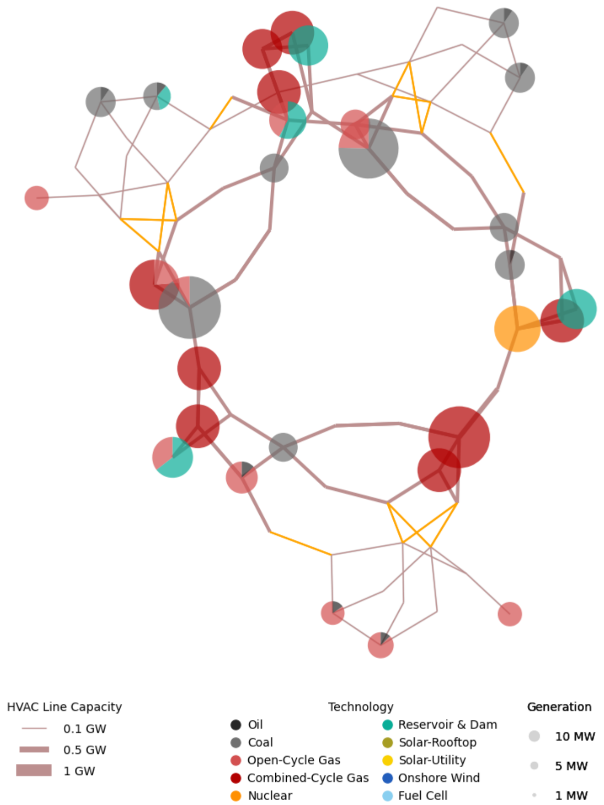


- Add data for 1 year!
- Add investment costs
- Add constraints (i.e.CO_2)
- **Solve**

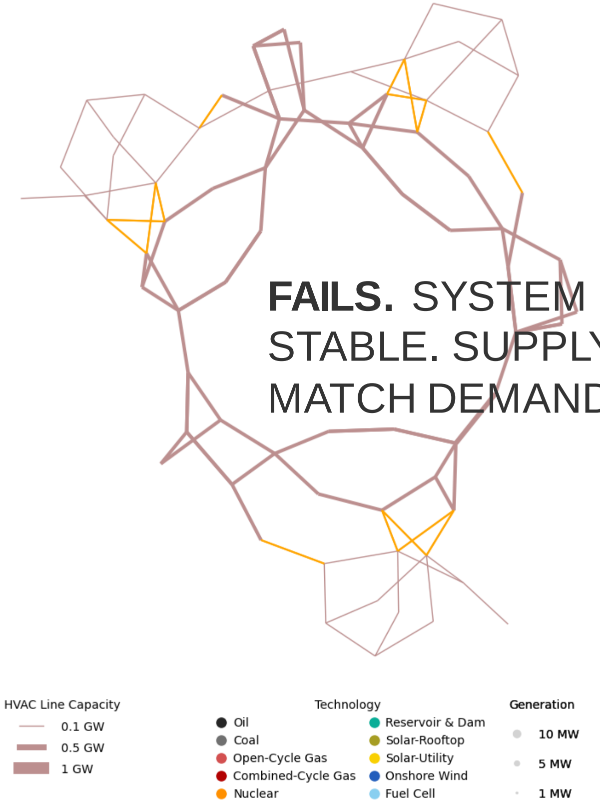
```
ilopf(n, solver_name=solver_name, solver_options=solver_options,
    track_iterations=track_iterations,
    min_iterations=min_iterations,
    max_iterations=max_iterations,
    extra_functionality=extra_functionality, **kwargs)
```

Planning exercise

ORIGINAL



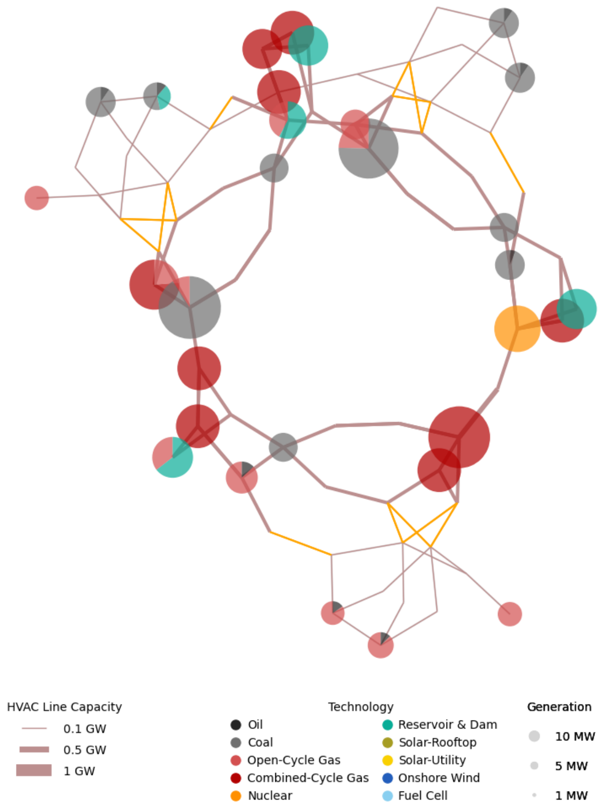
NO EXPANSION BUT LOPF



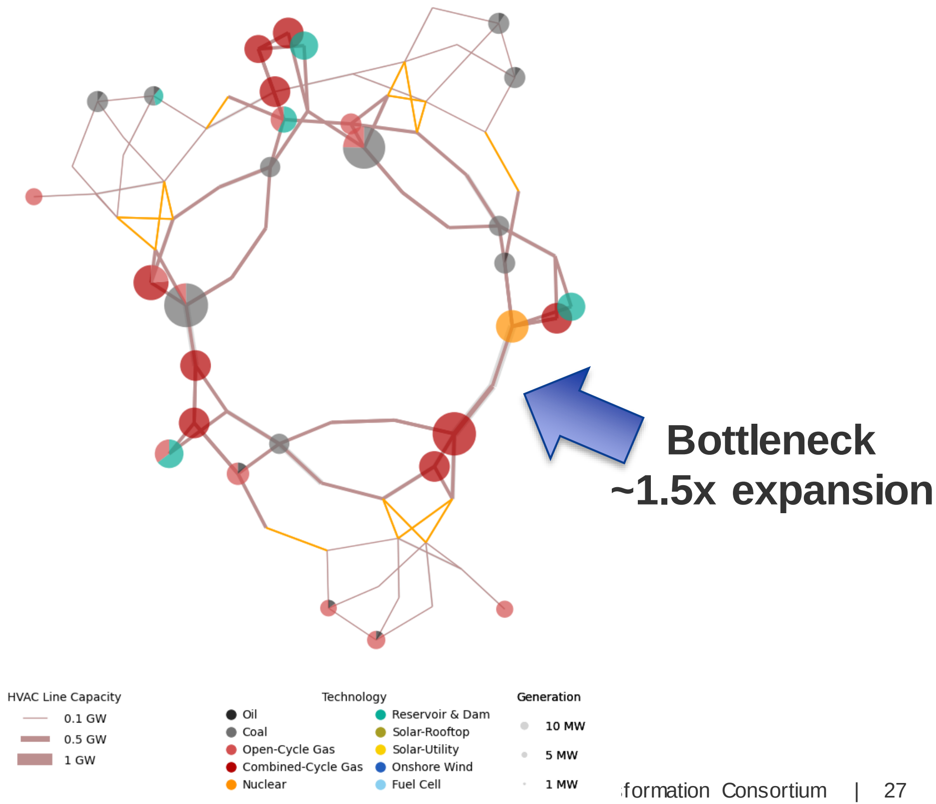
FAILS. SYSTEM NOT STABLE. SUPPLY DOES NOT MATCH DEMAND.

Planning exercise

ORIGINAL

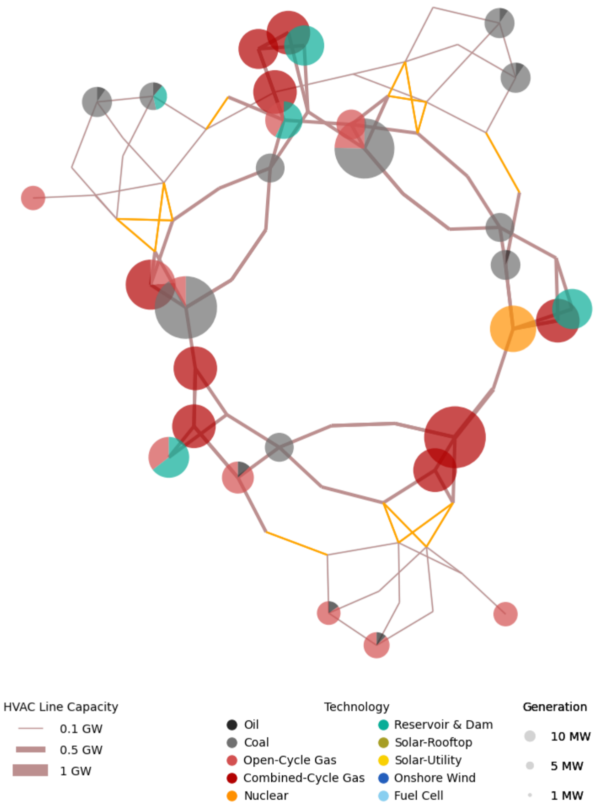


LINE EXPANSION

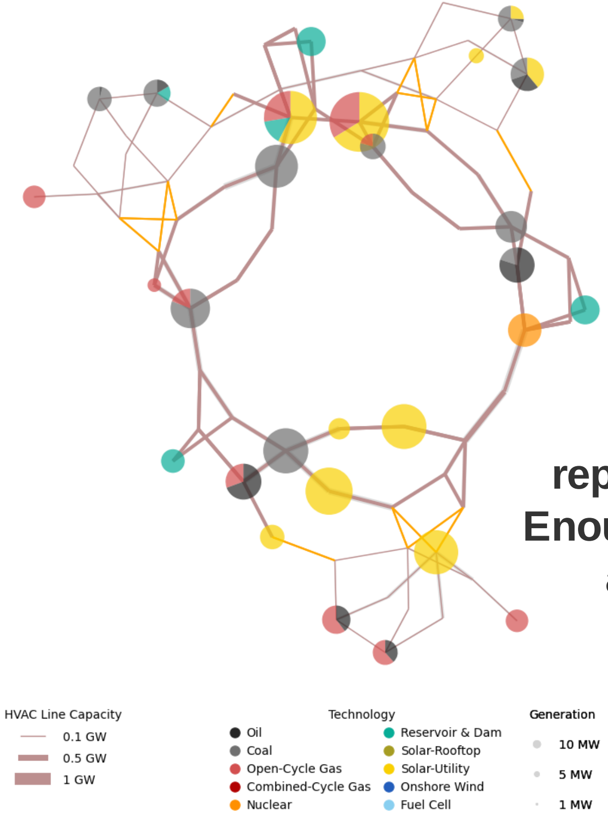


Planning exercise

ORIGINAL



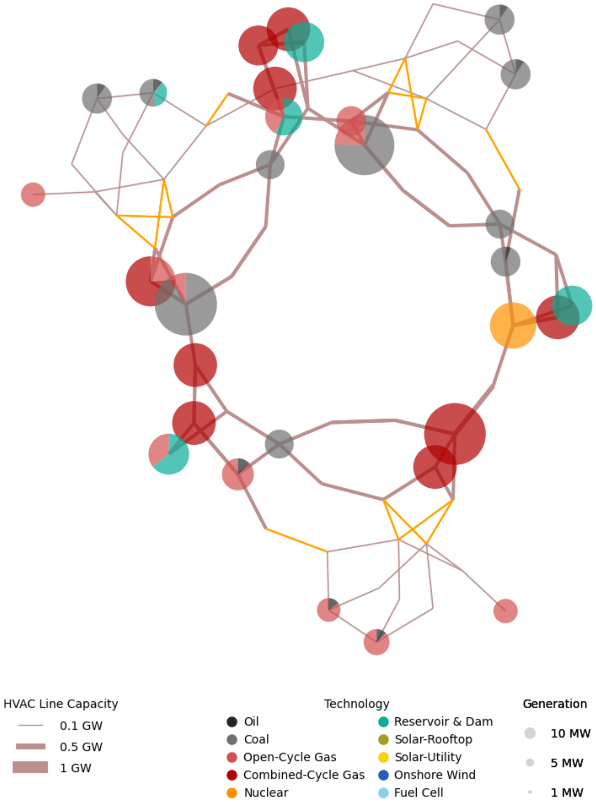
LINE + GEN. EXPANSION



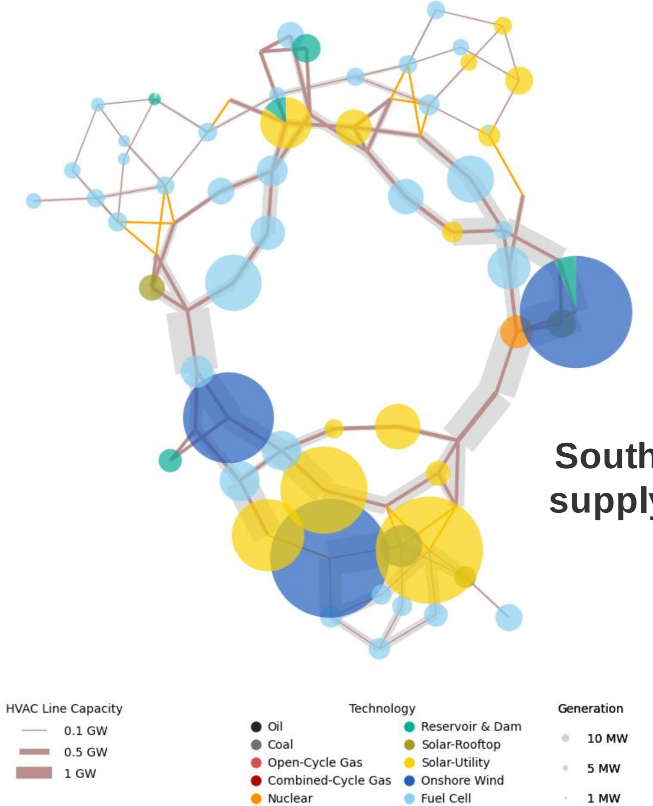
Solar PV replaces CCG. Enough flexibility available.

Planning exercise

ORIGINAL



LINE + GEN. EXPANSION + zero emission + 1.5x load



South = Main supply region

BUILDING A MODEL TAKES TIME... VALIDATING AND MAINTAINING IT DOES, TOO...

Photo by christopher lemercier https://unsplash.com/photos/12yvdCiLaVE

PyPSA is a framework. We build tools on top.

MODEL = Data+Framework

PyPSA

PyPSA meets Earth

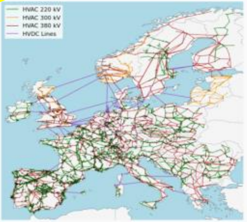
FRAMEWORK

PyPSA

A python software toolbox for simulating and optimising modern power systems.

MODEL

PyPSA-Eur



An open optimisation model of the European transmission system.

MODEL

PyPSA-Eur-Sec

A sector-coupled open optimisation model of the European energy system.

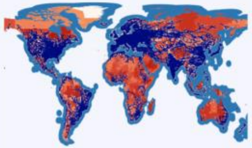
MODEL

PyPSA-Africa

An open optimization model of the African transmission system

MODEL

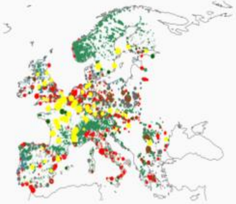
PyPSA-Earth



A highly flexible **sector-coupled** energy system model of the global energy system

DATA

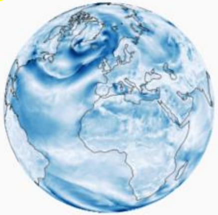
Powerplantmatching



A toolset for cleaning, standardizing and combining multiple power plant databases.

DATA

Atlite



A Lightweight Python Package for Calculating Renewable Power Potentials and Time Series

BETTER PYOMO

Linopy

Linear optimization interface for N-D labeled variables.

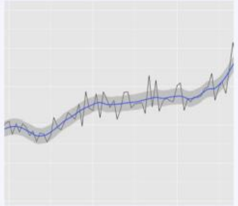
DATA

Detect-Energy

A machine learning framework to detect energy assets from satellites

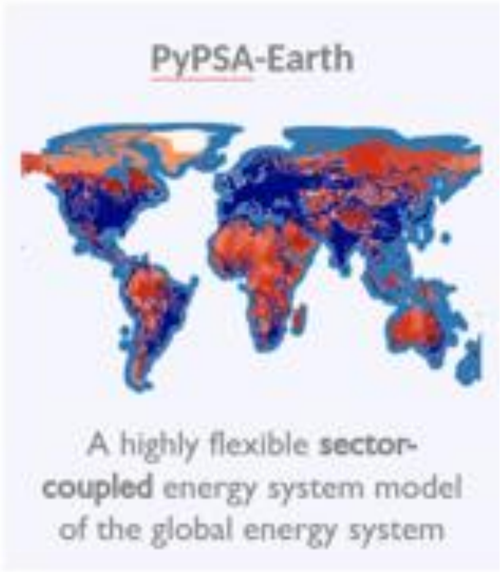
DATA

Demand-Creator



A general framework to create demand timeseries in subnational resolution

PyPSA-Earth: The Wikipedia for energy models.



- **MODEL ANY COUNTRY** OF THE EARTH WITH GLOBAL DEFAULT DATA WORKFLOW
- **REPLACE DEFAULT DATA WITH CUSTOM DATA** WORKFLOW THAT CAN BE SHARED = SUSTAINED (OR NOT)

"PyPSA meets Earth's vision is to create together the most compelling open-data and open-source planning tools to accelerate the world's sustainable energy transition."

CHALLENGE THE BLACK-BOX MODELLING STANDARD

MAXIMILIAN PARZEN

Co-steering the PyPSA meets Earth initiative

Address:
Institute of Energy Systems
University of Edinburgh
Kings Building
EH9 3JL Edinburgh, UK
+49 176 70889068

Contact:
https://pypsa-meets-africa.github.io/
max.parzen@ed.ac.uk
@maxparzen

THE UNIVERSITY of EDINBURGH

pandapower

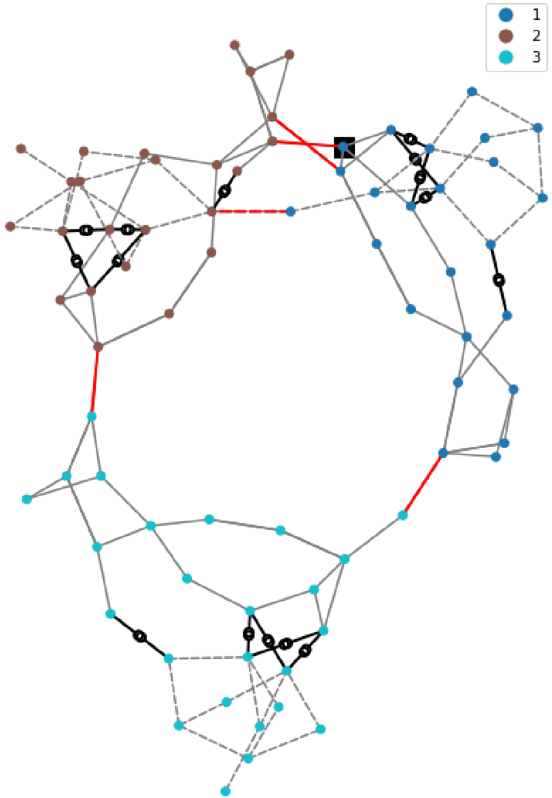
Roman Bolgaryn
Researcher, Fraunhofer IEE

pandapower

Useful links

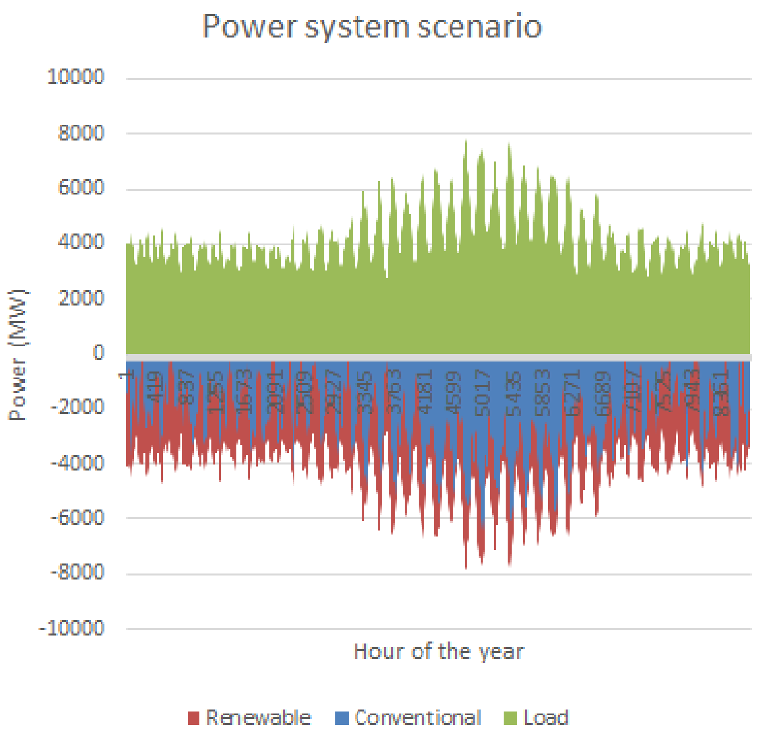
Official website:
http://www.pandapower.org/
GitHub repository:
https://github.com/e2nIEE/pandapower
Documentation:
https://pandapower.readthedocs.io

Case study for the RTS grid



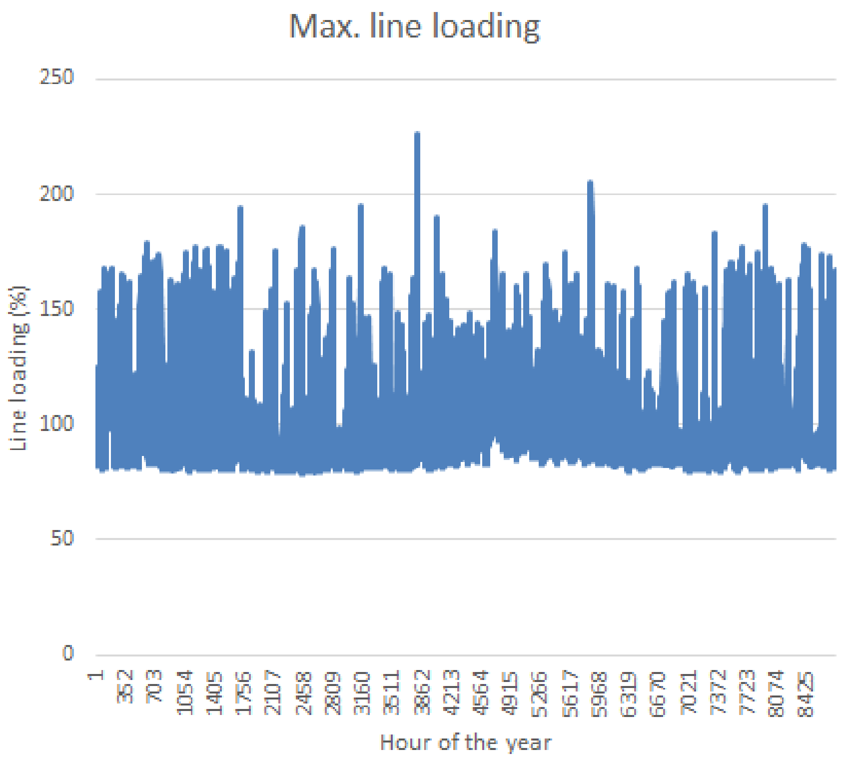
- Adjustments had to be made for the case study for AC-feasibility
  - Model conventional generation as PV node instead of PQ node
  - Use distributed slack with conventional generation
  - Add transformer tap changer controllers

Case study: scenario



- The scenario represents the load, conventional generation and renewable generation
- The conventional generation is used to cover the gap between the load and renewable generation

Case study: initial line loading



- Input data: load profiles and renewable generation profiles
- Conventional generation is considered in a simplified manner: balancing with the distributed slack approach, weighed by the installed power
- This results in line overloading during many time steps

Case study: method

$$\text{minimize} \sum (c_K^T \cdot \triangle P_K)$$

$$\text{subject to } I_{ft} - DF_{ft} \cdot \triangle P_K \leq I_{max,ft}$$

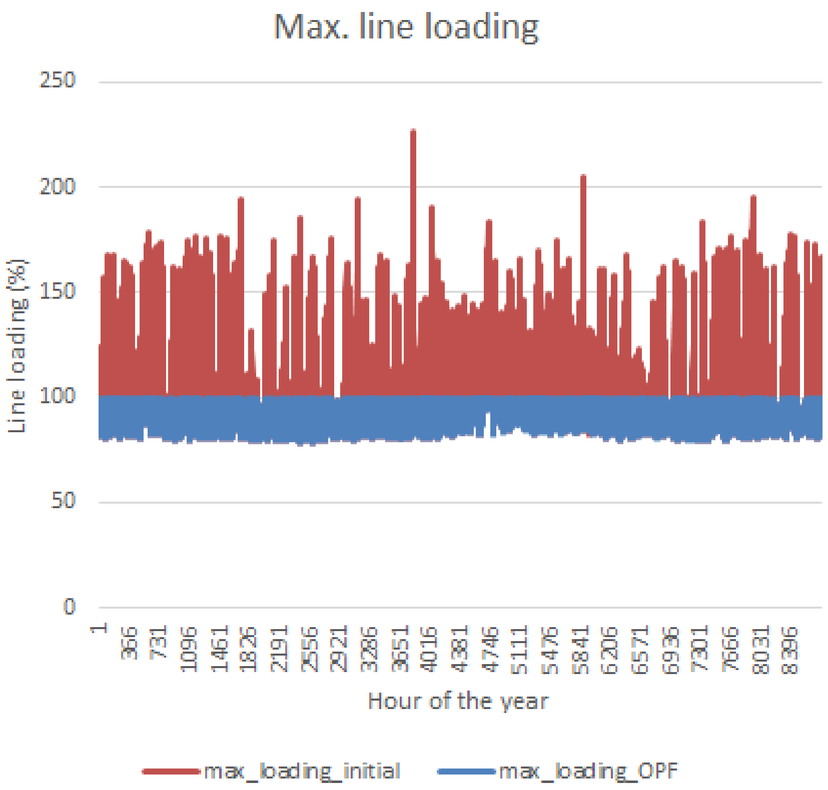
$$I_{ft} - DF_{ft} \cdot \triangle P_K \geq -I_{max,ft}$$

$$\text{(for every line (f, t))}$$

$$0 \leq \triangle P_K \leq P_{max,K}$$

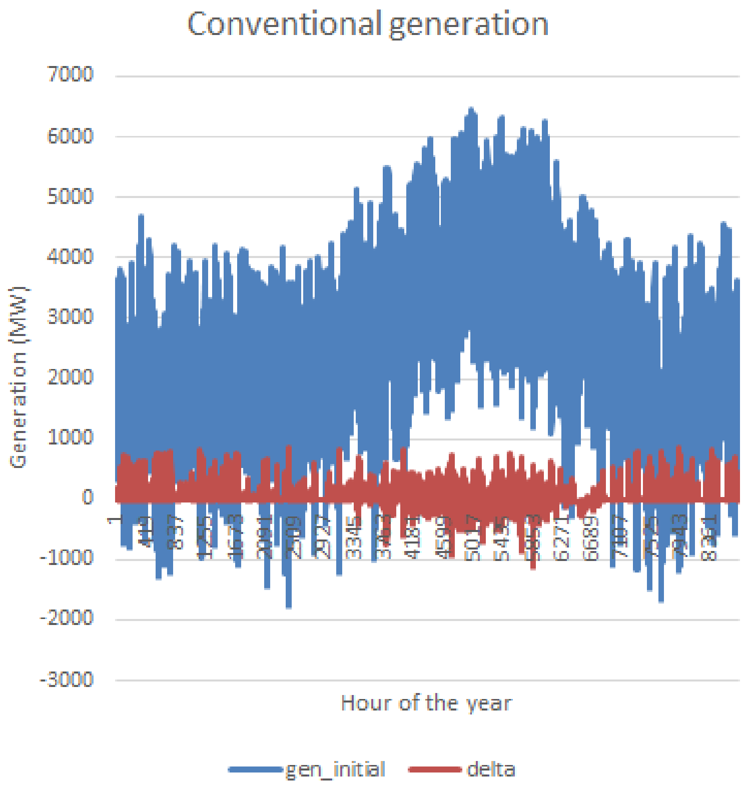
- AC-OPF with LP problem formulation
- Positive costs for conventional generation and negative costs for renewable generation
- OPF can be repeated with load shedding enabled for the time steps that fail. It was required for 1 time step in the case study

Case study: line loading



- The line loading could be maintained within a set limit for all time steps

Case study: generation



- The redispatch of conventional generation is not very high in overall (in absolute values)
- Reason: the balancing was already done with distributed slack
- The OPF is redistributing the conventional generation across the power system
- Renewable generation did not need to be redispatched

Conclusions and Further work

- Challenges in AC LP-OPF:
  - Performance
  - "Oscillations" of the solution
- Advantages:
  - AC-feasible solution
  - Possible to include outer-loop control (e.g. transformer tap changer), with an easier formulation

- Further work:
  - Security-Constrained AC LP-OPF
  - Consideration of preventive and curative measures
  - Consideration of overhead line temperature and thermal inertia

System
Planning with Powsybl – Metrix

Application to the GLMC-RTS model

Nicolas Omont
Vice President of Operations, Artelys

GLOBAL PST CONSORTIUM

LF ENERGY POWSYBL

System Planning with Powsybl – Metrix

Application to the GLMC-RTS model

Linux Foundation Energy – GMLC Webinar

Introduction to Powsybl

- **PowSyBl** (Power System Blocks, powsybl.org) is an open-source framework written in Java and dedicated to power grid analysis and simulation
  - Created in 2012 (**iTesla** EU funded collaborative R&D project)
  - Community of 70 users

Many supported formats
CIM-CGMES, UCTE, Matpower, PSS/E…

Power-flows, OPFs, security analysis, dynamic simulation

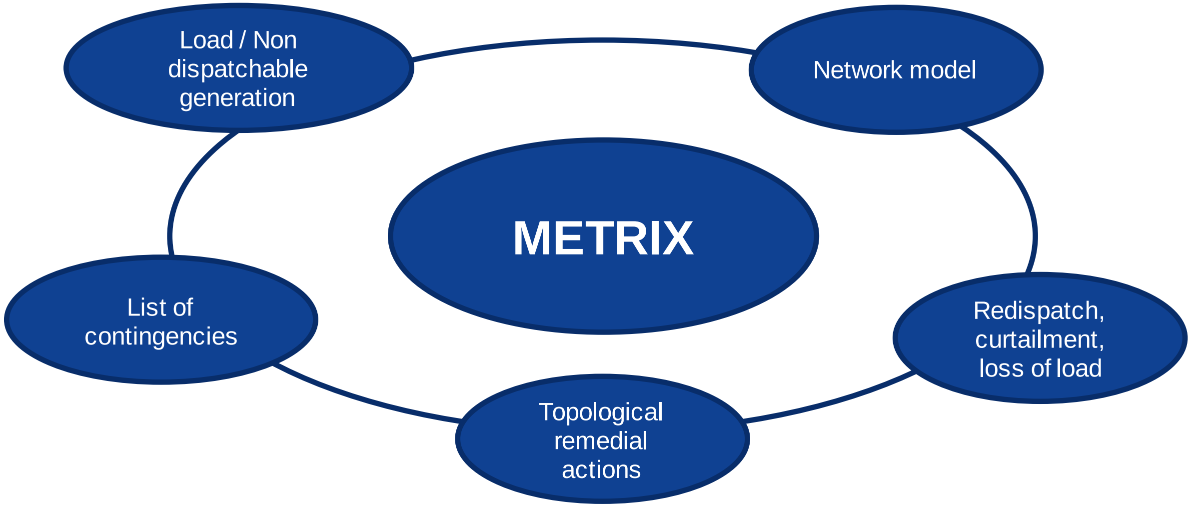
Advanced features
Analysis tools, visualization tools

Grid topology operations
Extensions, Extractions, Merges…

Friendly to non-Java developers
Python binding, fine-tuning using Groovy

Introduction to Metrix (1/2)

- An optimization model used to assess and optimize preventive and curative remedial actions to respect the network constraints on a high number of variants
  - Created in 2010 (fully open-sourced in 2021, including the linear solver)
  - Interfaced with PowSyBl



Introduction to Metrix (2/2)

- Three computation modes

DC security analysis (N, N-k)

No optimization, simple power flow

Inputs:

- Network model
- Base case topology
- Contingencies (N-k)
- Load and generation timeseries (Gen. must match demand)

Results:

- Flows at each element (N)
- Max flow violations (N, N-k)

SC-DCOPF* w/o redispatching (N, N-k)

Minimizing: **max flow violations**

Inputs:

- *Same as DC security analysis*
- Available topological remedial actions (preventive and curative)

Results:

- *Same as DC security analysis*
- Selected preventive actions
- Selected curative actions
- Remaining violations (N, N-k)

SC-DCOPF* w/ redispatching (N, N-k)

Minimizing global cost while satisfying max flow constraints

Inputs:

- *Same as DC security analysis*
- Available preventive and curative actions
  - Topological remedial actions
  - Redispatch costs

Results:

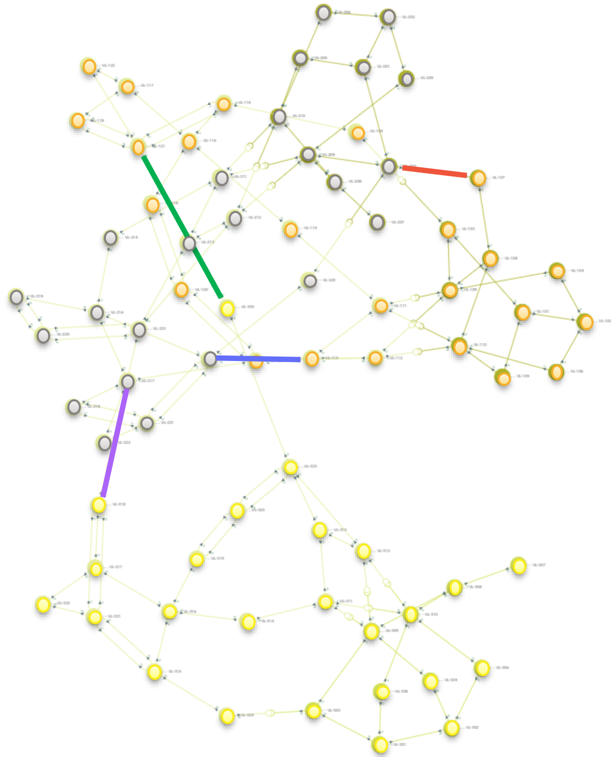
- *Same as SC-DCOPF without redispatching*
- Production and consumption adjustments (redispatch, curtailment, loss of load)

* SC-DCOPF = Security Constrained Direct Current Approximation Optimal Power Flow

A tutorial covering this functional perimeter is available online: https://github.com/powsybl/powsybl-tutorials/tree/metrix/metrix/src/main
Tutorial introduction video: https://vimeo.com/722882701
Tutorial introduction pdf: https://github.com/powsybl/powsybl-tutorials/raw/metrix/metrix/src/main/resources/PowSyBl-metrix-6-node-tutorial-presentation.pdf

Application on the RTS-GMLC model

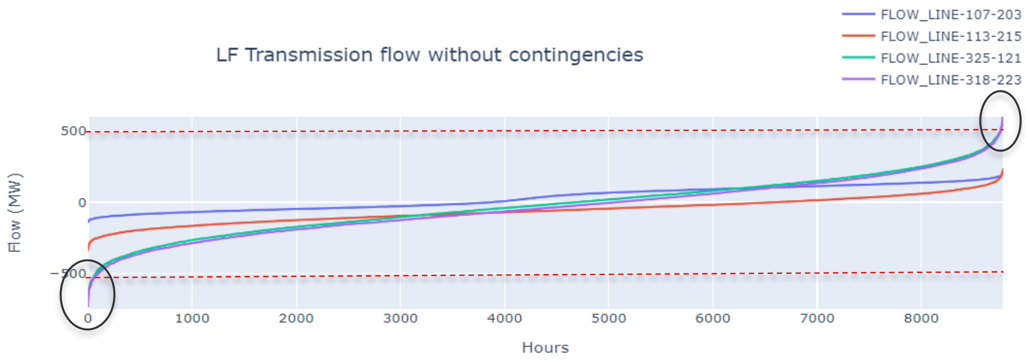
- Study case building:
  - Loading the network from the Matpower format
  - Mapping annual non-dispatchable generation and load timeseries (8784 hourly time steps) to each node, as well as dispatchable generation initial set points.
- Study computation:
  - N-K analysis: single contingencies (N-1) on all 120 elements (lines+trafos)
  - Available remedial actions: preventive redispatch
  - Evaluation of the value of an inter-zone transmission line by comparison of the cases w/ and w/o this line.
    - Not in scope: topological actions, curative remedial actions, HVDC lines.
- Analysis (KPIs)
  - Flows on the four inter-zone transmission lines (in bold on the map)
  - Localization of their threats (i.e. the contingency leading to the largest flow on a given line)
  - Assessment of redispatch and curtailment to evaluate the decrease of generation costs brought by the inter-zone transmission line.



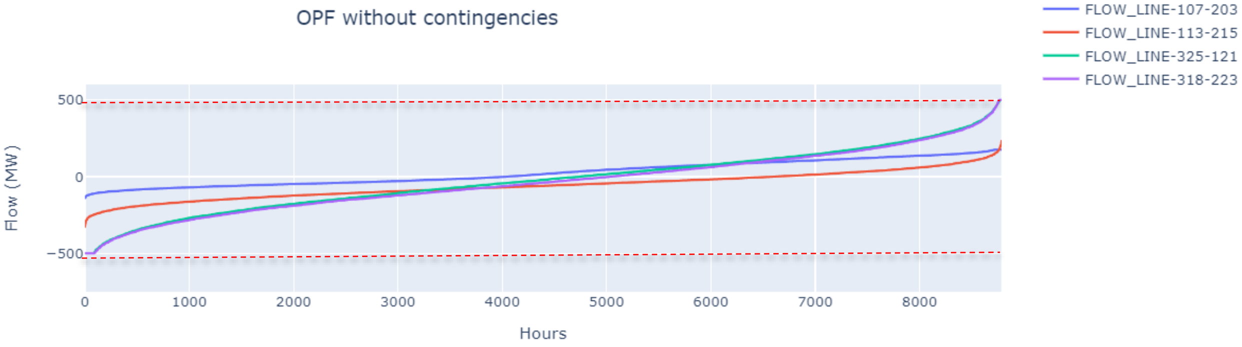
Data: https://github.com/GridMod/RTS-GMLC

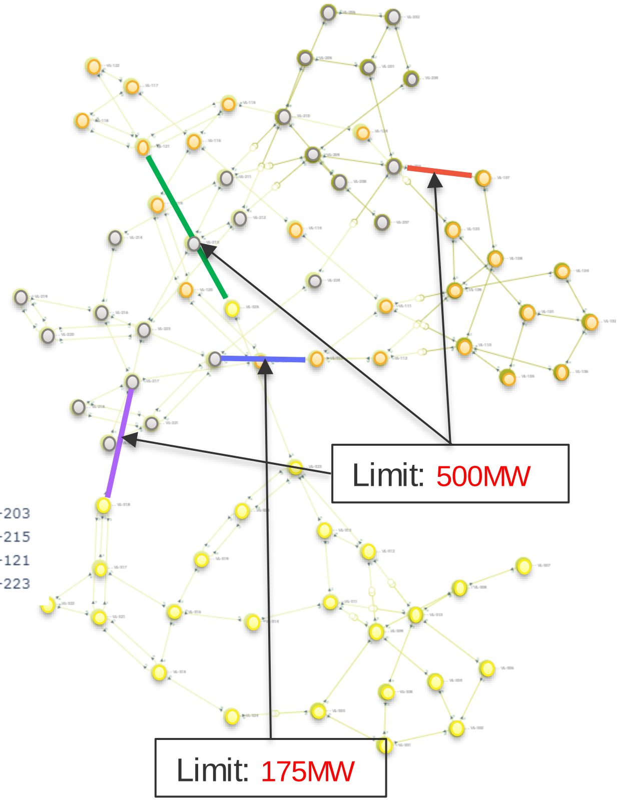
Base case **without** contingencies (N)

- Power flow (w/o redispatch)



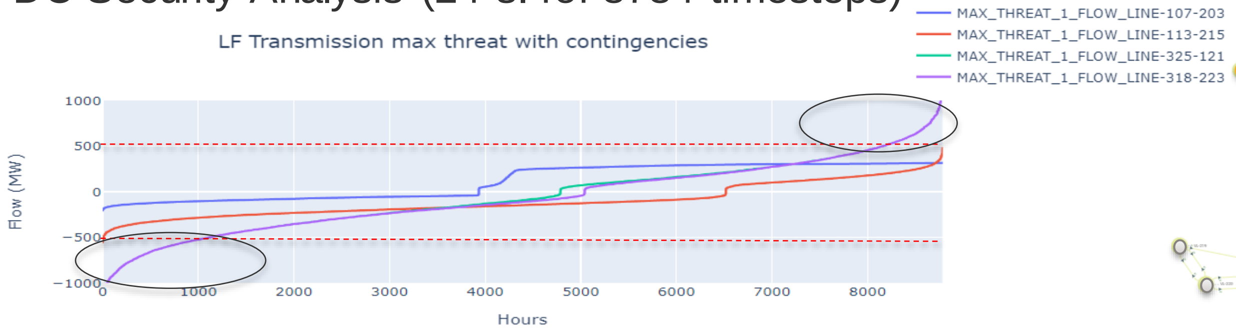
- With redispatch: no major difference (~almost no N constraints)



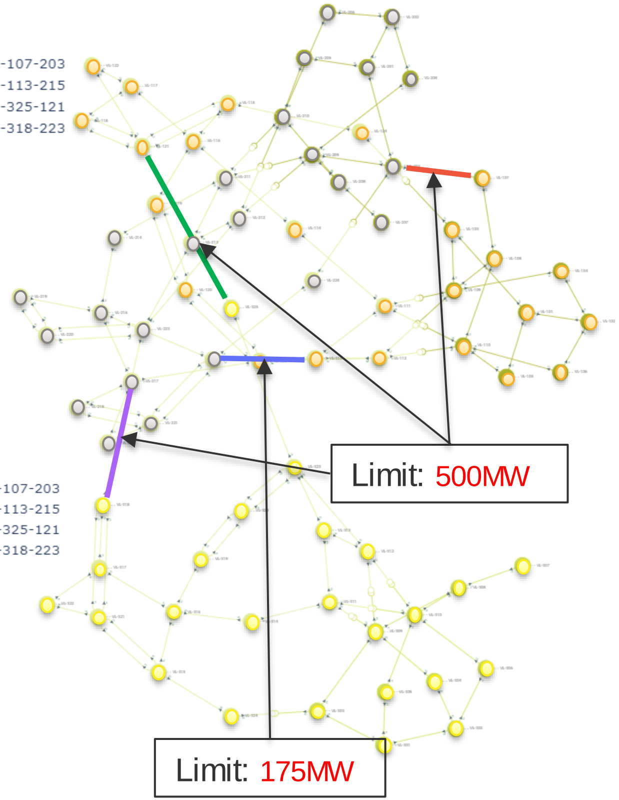


Base case **with** contingencies (N, N-1)

- DC Security Analysis (24 s. for 8784 timesteps)

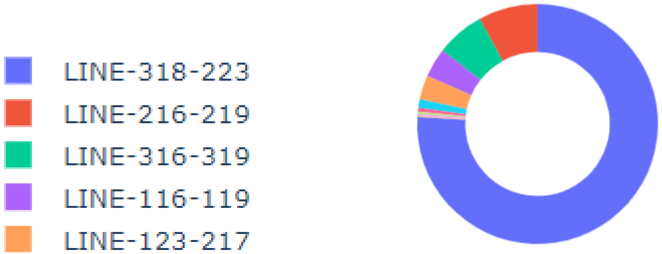


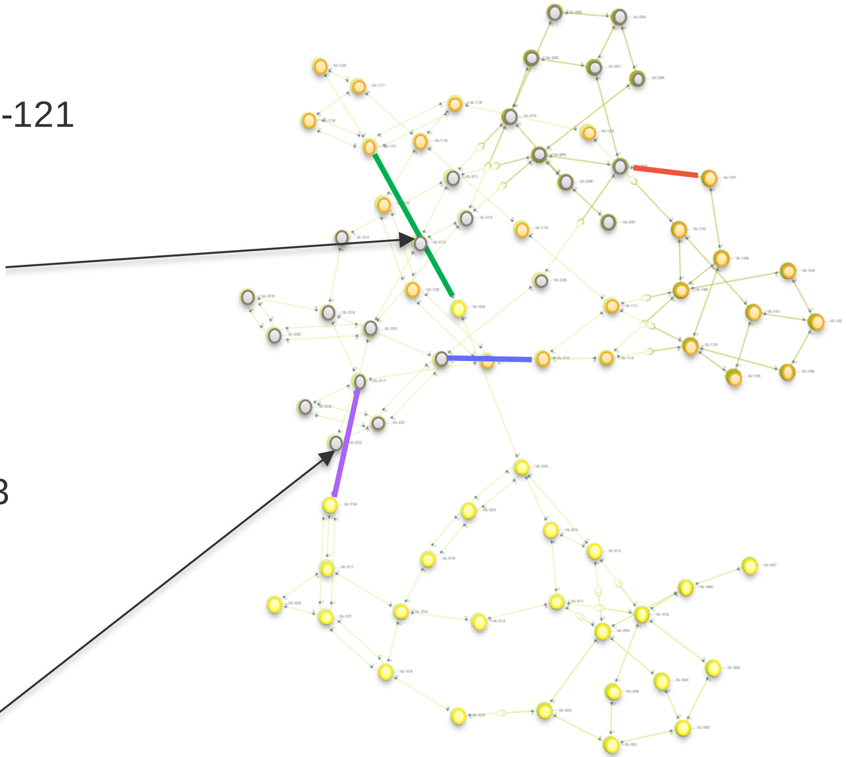
- Optimized Power Flow (49 s. for 8784 t.s.)
  - Enforcement of max flow limits is visible



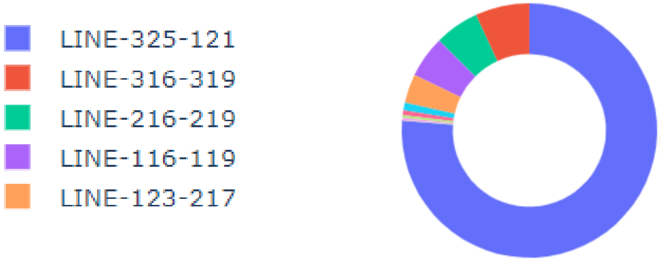
Base case **with** contingencies: threat location

- Location of the contingency causing the max flow
  - For Line 318-223: For 75% of time steps, line 325-121



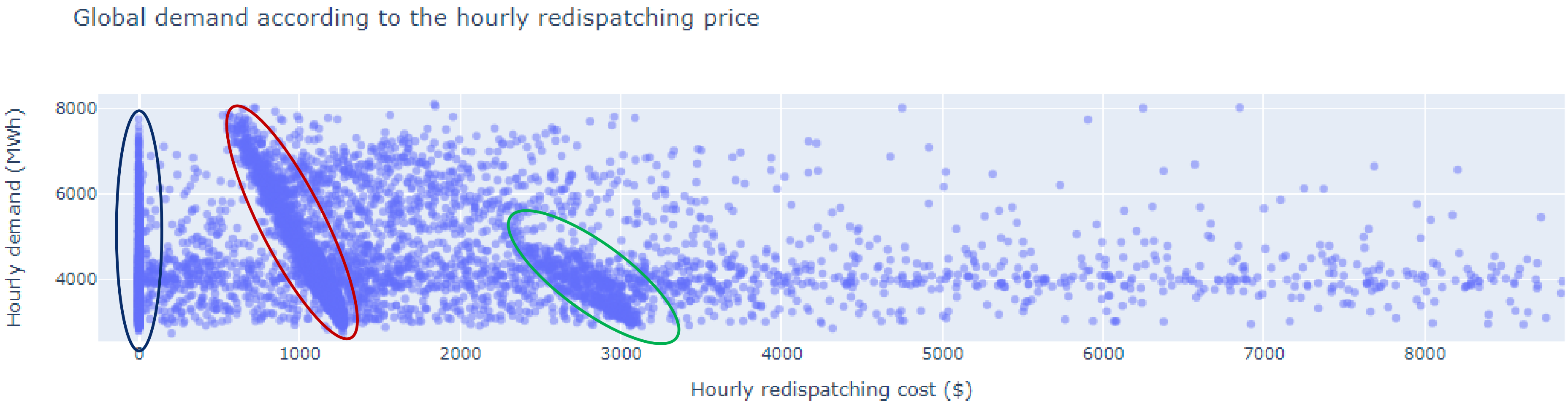


  - For Line 325-121: 80% of time steps, line 318-223



Base case with contingencies: redispatch

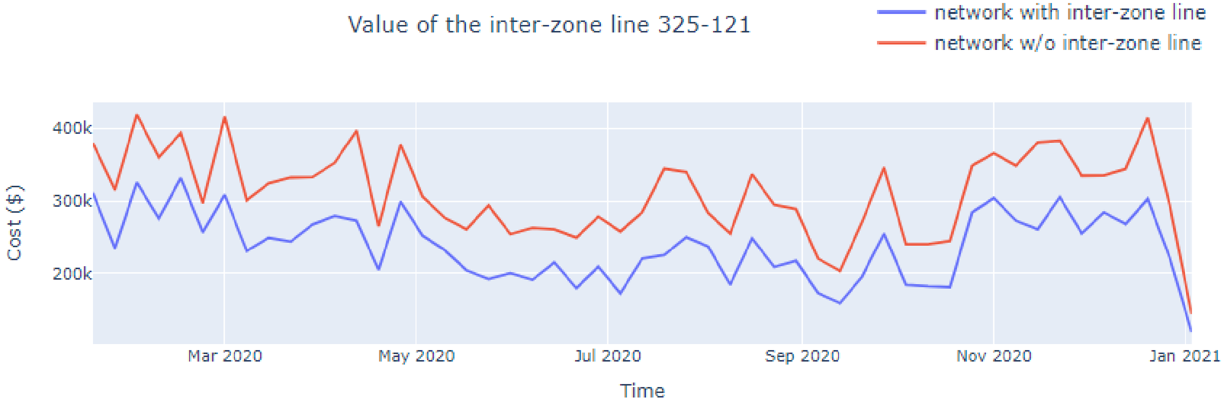
- Redispatch cost

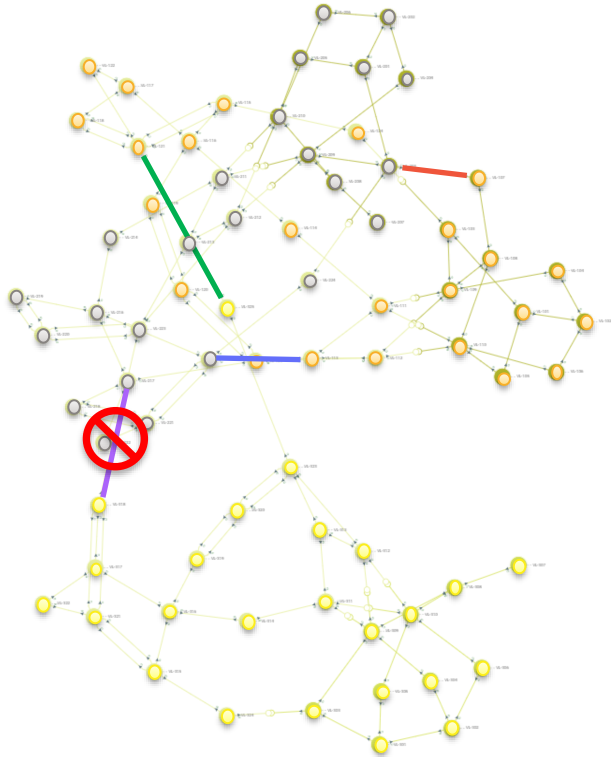


3 main regimes:
- No congestion
- Increase of ~0.1 $/MWh of additional load
- Increase of ~0.5 $/MWh of additional load

Value of a transmission line

- Weekly generation cost
  - Comparison **with and without line 325-121**, one of the two lines linking zone 1 and 3
  - Line's total savings = $3,7M/year





Conclusion on PowSyBl-Metrix

- PowSyBl Metrix is a SC-DCOPF
  - Handling both **preventive** and **curative** remedial actions
  - Including **redispatching**, **HVDC** set point adjustment and **topological** actions
  - Build for outstanding performance on sequential simulations (on independent timesteps)

- Illustration of preventive redispatching on the RTS-GMLC model
  - Maximum threat a line is subject to.
  - Cost of redispatch
  - Value of a line by comparison of redispatching cost w/ and w/o the line

- More info online:
  - Installation guide
  - A tutorial based on a 6-node grid illustrates all features:

  https://github.com/powsybl/powsybl-tutorials/tree/metrix/metrix/src/main

PowerSimulations

Clayton Barrows
Senior Engineer, NREL

PowerSystems.jl

- Data:
  - Transmission systems
  - Quasi-static, technical and economic system operational data
  - Dynamic parameters
  - Time series
  - Parsing from standard file formats (.m, .csv, .raw, dyr)
  - De/serialization (compressed storage)
  - Consistency checks
- Calculations
  - Network matrices (YBus, Adj, PTDF, LODF)
  - Power flow

```
Component
├─ Device
│  ├─ Branch
│  │  ├─ ACBranch
│  │  │  ├─ DynamicBranch
│  │  │  ├─ Line
│  │  │  ├─ MonitoredLine
│  │  │  ├─ PhaseShiftingTransformer
│  │  │  ├─ TapTransformer
│  │  │  └─ Transformer2W
│  │  └─ DCBranch
│  │     ├─ HVDCLine
│  │     └─ VSCDCLine
│  ├─ DynamicInjection
│  │  ├─ DynamicGenerator
│  │  ├─ DynamicInverter
│  │  └─ PeriodicVariableSource
│  ├─ RegulationDevice
│  └─ StaticInjection
│     ├─ ElectricLoad
│     │  ├─ ControllableLoad
│     │  │  └─ InterruptibleLoad
│     │  │     ⋮
│     │  ├─ FixedAdmittance
│     │  └─ StaticLoad
│     │     └─ PowerLoad
│     │        ⋮
│     ├─ Generator
│     │  ├─ HydroGen
│     │  │  ├─ HydroDispatch
│     │  │  │  ⋮
│     │  │  ├─ HydroEnergyReservoir
│     │  │  │  ⋮
│     │  │  └─ HydroPumpedStorage
│     │  │     ⋮
│     │  ├─ RenewableGen
│     │  │  ├─ RenewableDispatch
│     │  │  │  ⋮
│     │  │  └─ RenewableFix
│     │  │     ⋮
│     │  └─ ThermalGen
│     │     ├─ ThermalMultiStart
│     │     │  ⋮
│     │     └─ ThermalStandard
│     │        ⋮
│     ├─ Source
│     ├─ StaticInjectionSubsystem
│     │  └─ HybridSystem
│     └─ Storage
│        ├─ BatteryEMS
│        └─ GenericBattery
```

```
├─ Service
│  ├─ AGC
│  ├─ AbstractReserve
│  │  ├─ Reserve
│  │  │  ├─ ReserveDemandCurve
│  │  │  ├─ StaticReserve
│  │  │  └─ VariableReserve
│  │  └─ ReserveNonSpinning
│  │     ├─ StaticReserveNonSpinning
│  │     └─ VariableReserveNonSpinning
│  ├─ StaticReserveGroup
│  └─ Transfer
└─ Topology
   ├─ AggregationTopology
   │  ├─ Area
   │  └─ LoadZone
   ├─ Arc
   └─ Bus
```

PyPSA2PowerSystems.jl

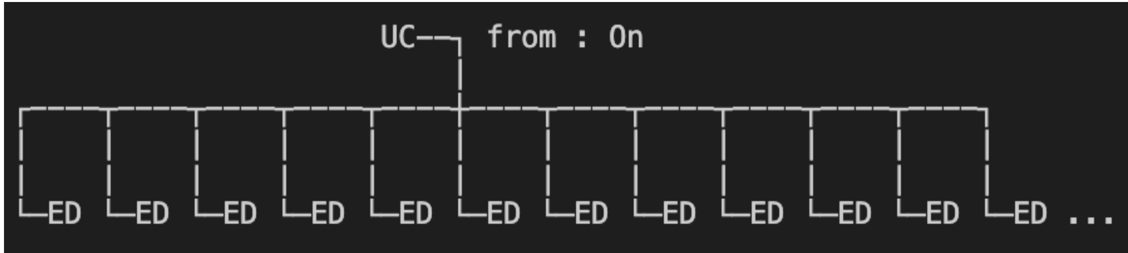
Creates a System from a PyPSA netCDF input or output file

NREL-SIIP/
PyPSA2PowerSystems.jl

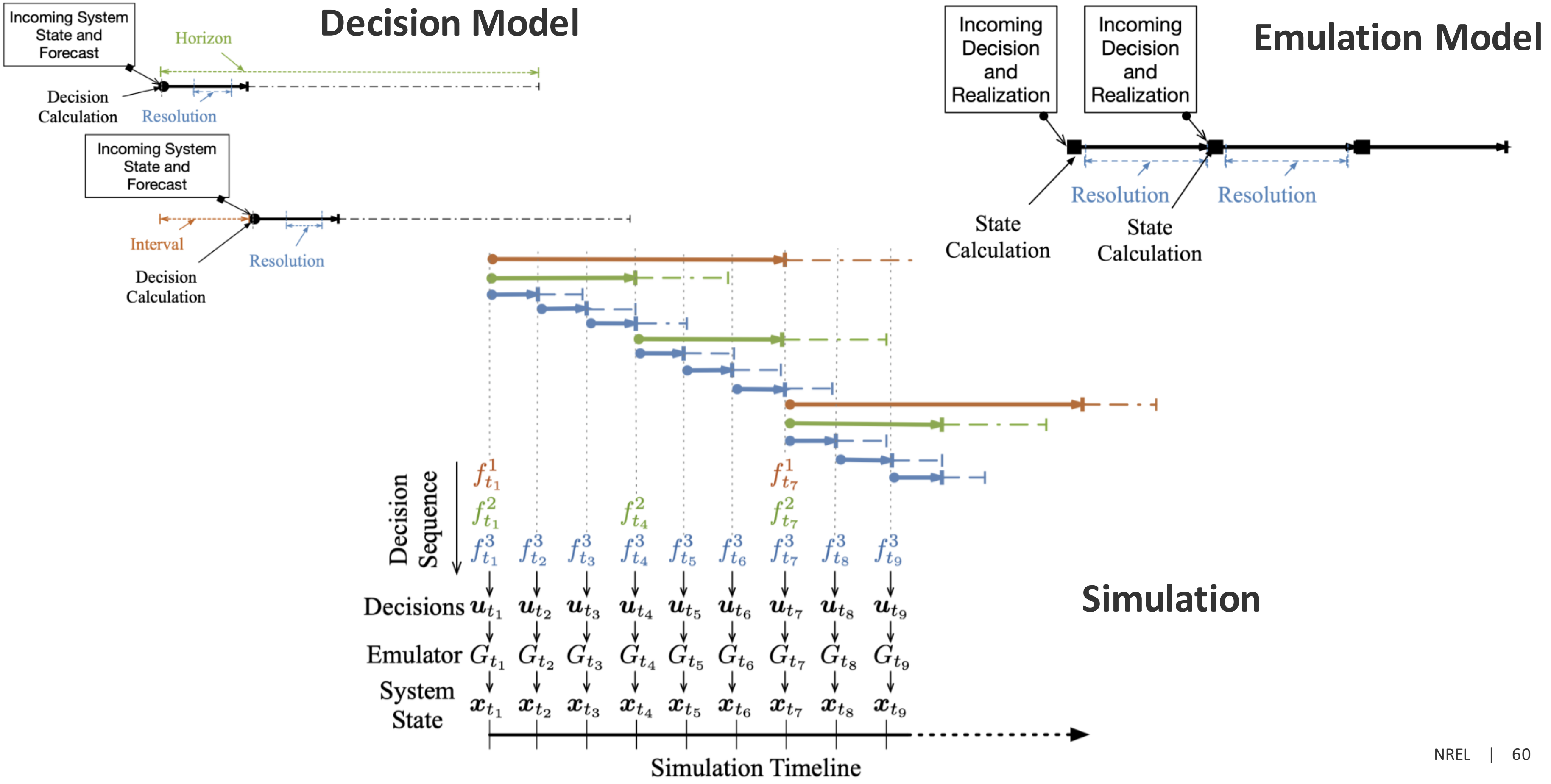
1	1	0	0
Contributor	Issue	Stars	Forks

PowerSimulations.jl

- Formulations:
  - Devices: 55 formulations
  - Networks: 52 formulations (through integration with PowerModels.jl)
  - Services (reserves): 8 formulations
- Models:
- Decision Model: UC, ED… usually multi-period forecast data
  - Emulation Model: single period realization data
- Simulations:
  - Multiple executions: 365x UC
  - Multiple models:
    - 365x (UC – 24x ED) = 365 + 8,760 = 9,125
    - 365x (DA – 24x (HA – 12x (RT – 75x AGC))) = 365 + 8,760 + 105,120 + 7,884,000 = 7,998,245
    - LT – DA, DA – RT – Emulation…
  - Co-simulation
    - Helics
    - Custom



Flexible Simulation Specifications



Exercise Simulation Specification

Simulation

Simulation Name	simulation
Build Status	EMPTY
Run Status	NOT_READY
Initial Time	Unset Initial Time
Steps	364

Decision Models

Model Name	Model Type	Status	Output Directory
UC	GenericOpProblem	EMPTY	nothing

No Emulator Model Specified

Simulation Sequence

Simulation Step Interval	24 hours
Number of Problems	1

Simulation Problems

Model Name	Horizon	Interval	Executions Per Step
UC	48	1440 minutes	1

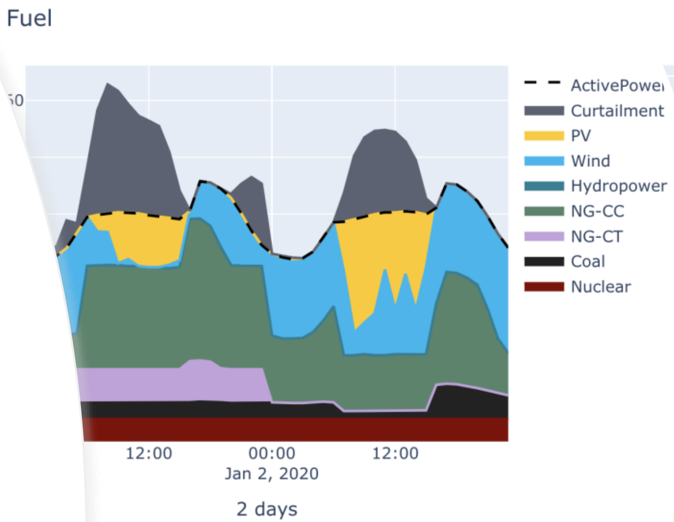
Info:

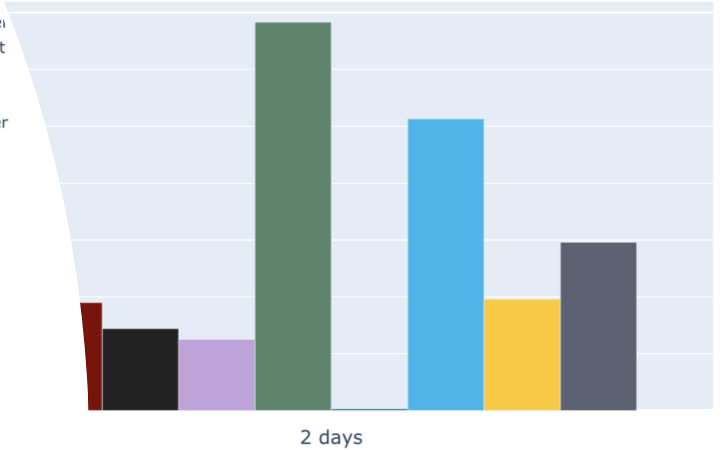
		Time			Allocations		
Tot / % measured:		10.1s / 100.0%			0.96GiB / 100.0%		
Section	ncalls	time	%tot	avg	alloc	%tot	avg
Build Simulation	1	10.1s	100.0%	10.1s	0.96GiB	100.0%	0.96GiB
Build Problems	1	8.01s	79.1%	8.01s	859MiB	87.8%	859MiB
Problem UC	1	7.87s	77.7%	7.87s	756MiB	77.2%	756MiB
MonitoredLine	2	3.36s	33.2%	1.68s	229MiB	23.4%	115MiB
ThermalStandard	2	2.00s	19.8%	1.00s	302MiB	30.9%	151MiB
Line	2	1.96s	19.4%	982ms	157MiB	16.0%	78.5MiB
HydroDispatch	2	204ms	2.0%	102ms	25.0MiB	2.6%	12.5MiB
RenewableDis...	2	172ms	1.7%	86.1ms	25.4MiB	2.6%	12.7MiB
PowerLoad	2	124ms	1.2%	61.8ms	4.00MiB	0.4%	2.00MiB
Build pre-step	1	27.6ms	0.3%	27.6ms	2.05MiB	0.2%	2.05MiB
CopperPlateP...	1	8.39ms	0.1%	8.39ms	7.16MiB	0.7%	7.16MiB
Objective	1	4.02ms	0.0%	4.02ms	3.32MiB	0.3%	3.32MiB
Transformer2W	2	1.16ms	0.0%	581μs	342KiB	0.0%	171KiB
Services	2	2.19μs	0.0%	1.09μs	0.00B	0.0%	0.00B
Initialize Simul...	1	197ms	1.9%	197ms	10.4MiB	1.1%	10.4MiB
Initialize Simul...	1	2.80ms	0.0%	2.80ms	1.59MiB	0.2%	1.59MiB
Serializing Simu...	1	667μs	0.0%	667μs	10.6KiB	0.0%	10.6KiB
Check Steps	1	28.4μs	0.0%	28.4μs	352B	0.0%	352B

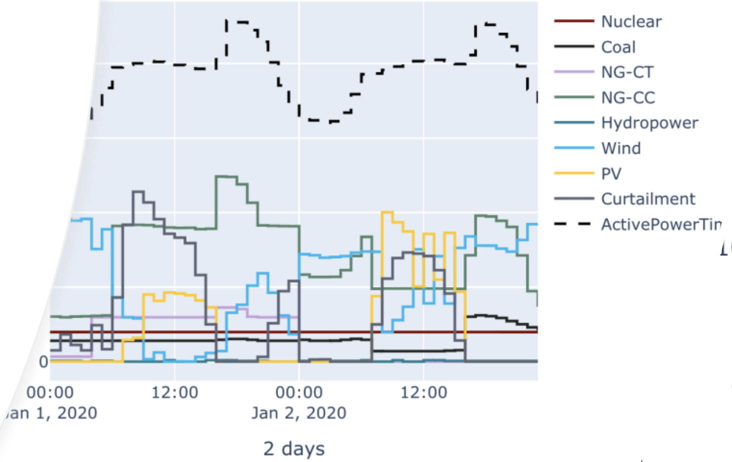
BuildStatus.BUILT = 0

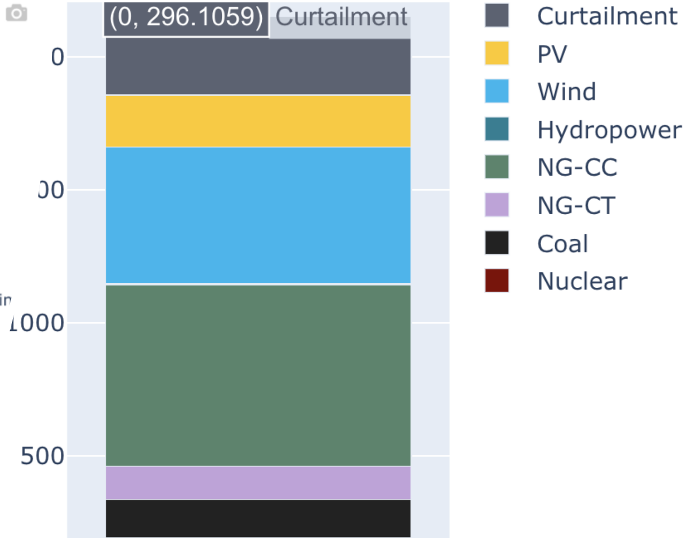
PowerGraphics.jl

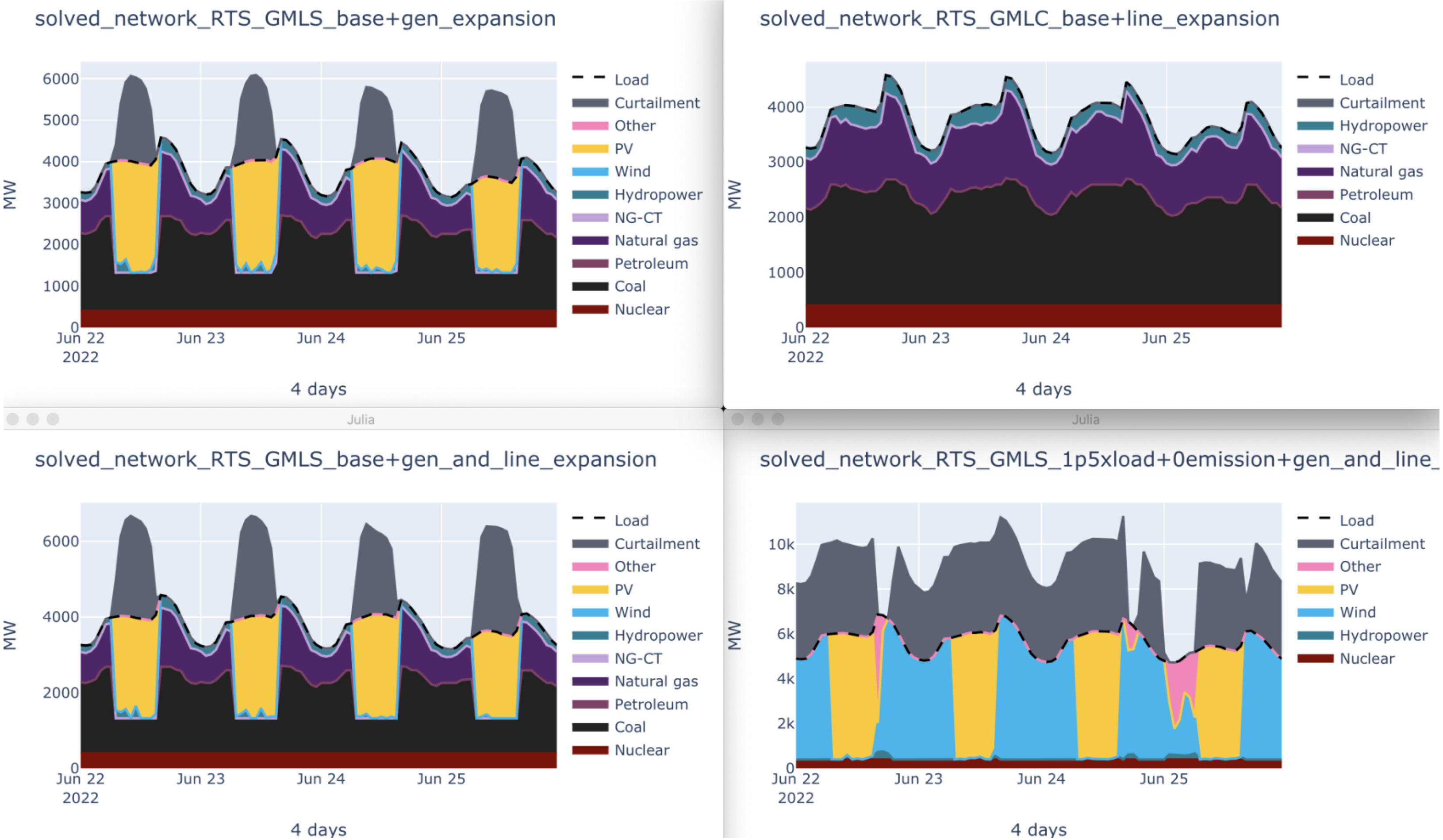
- **Plot types:** bar, stack, line, (coming soon: networks)
- **Data:** System, PSI.Results, (coming soon: PSID.Results)
- **Backends:** GR (static), PlotlyJS (basic interactivity)



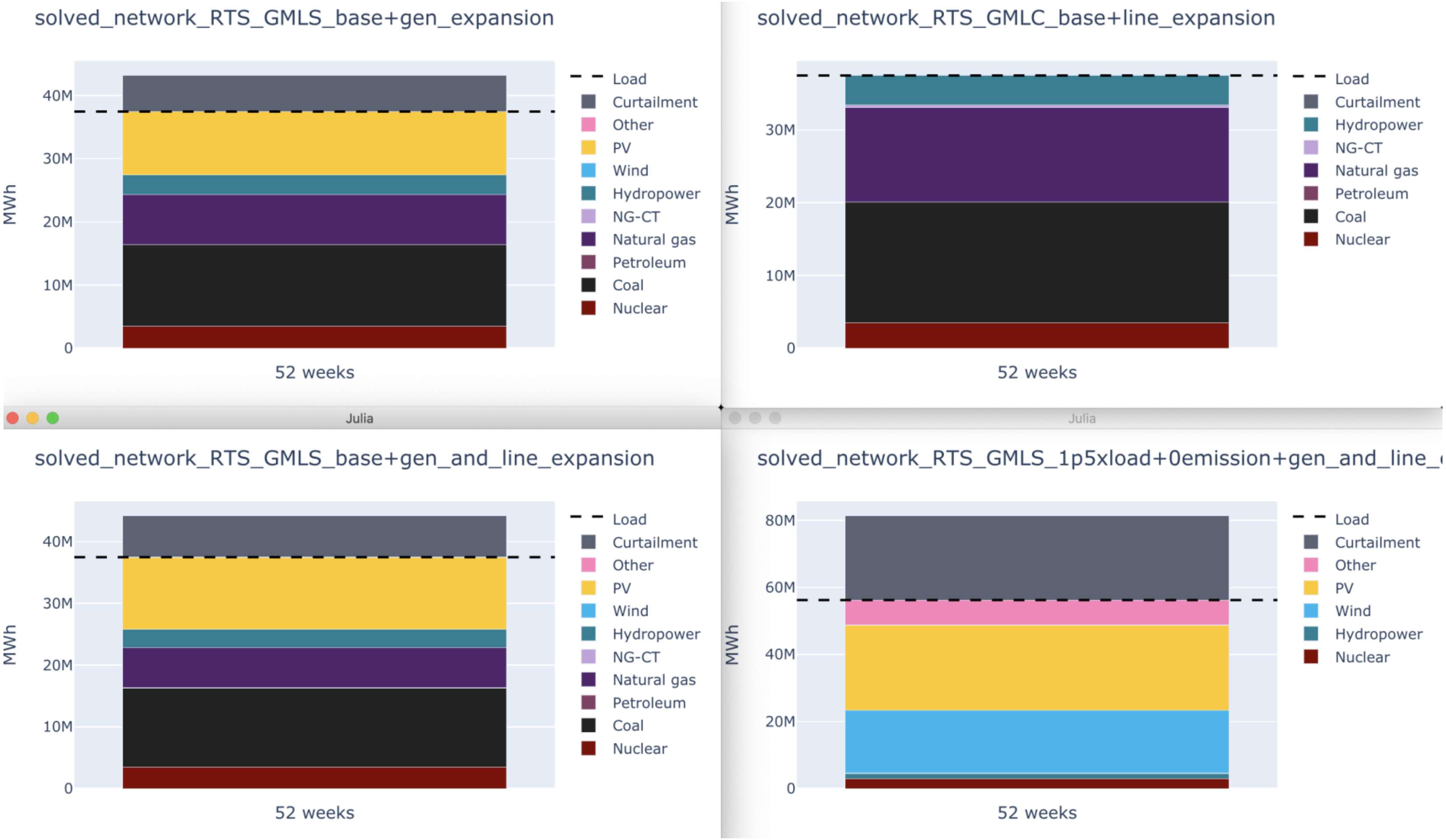








Example Scenario Results

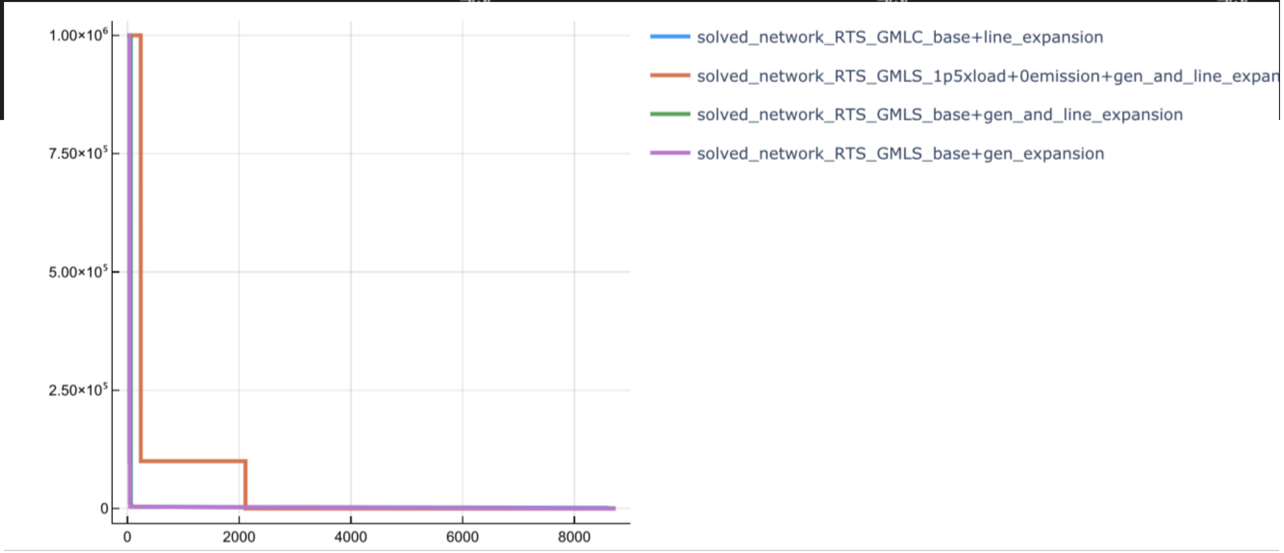


Example Scenario Results: Annual Generation

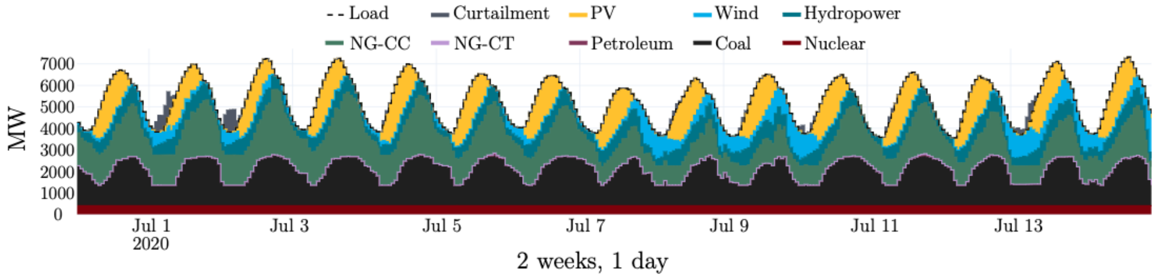
All results are accessible in DataFrames

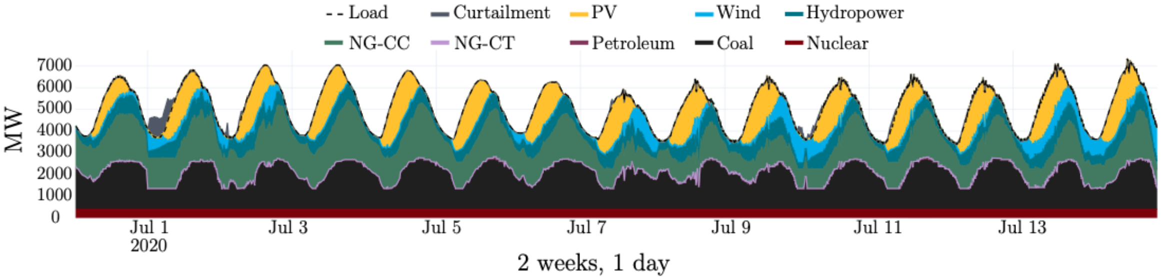
```
julia> prices_ts
8736×5 DataFrame
```

Row	DateTime DateTime	solved_network_RTS_GMLC_base+line_expansion Float64	solved_network_RTS_GMLS_1p5xload+0emission+gen_and_line_expansion Float64	solved_network_RTS_GMLS_base+gen_and_line_expansion Float64	solved_network_RTS_GMLS_base+gen_expansion Float64
1	2022-06-22T00:00:00	2184.39	-230.0	2296.85	2296.85
2	2022-06-22T01:00:00	2164.73	-230.0	2273.25	2273.25
3	2022-06-22T02:00:00	2218.63	-230.0	2307.0	2307.0
4	2022-06-22T03:00:00	2273.25	-230.0	2320.67	2320.67
5	2022-06-22T04:00:00	2320.67	-230.0	2575.9	2575.9
6	2022-06-22T05:00:00	2590.83	-230.0	2775.48	2775.48
7	2022-06-22T06:00:00	2632.43	-230.0	2775.48	2775.48
8	2022-06-22T07:00:00	2461.74	-0.0	-0.0	-0.0
9	2022-06-22T08:00:00	2590.83	-0.0	-0.0	-0.0
10	2022-06-22T09:00:00	2461.74	-0.0	-0.0	-0.0
11	2022-06-22T10:00:00	2387.54	-0.0	-0.0	-0.0
12	2022-06-22T11:00:00	2365.77	-0.0	-0.0	-0.0
13	2022-06-22T12:00:00	2365.77	-0.0	-0.0	-0.0
14	2022-06-22T13:00:00	2365.77	-0.0	-0.0	-0.0
15	2022-06-22T14:00:00	2365.77			
16	2022-06-22T15:00:00	2461.74			
17	2022-06-22T16:00:00	3246.22			
18	2022-06-22T17:00:00	3053.02			
19	2022-06-22T18:00:00	3053.02			
20	2022-06-22T19:00:00	2705.06			



PowerSimulations.jl Benchmarks



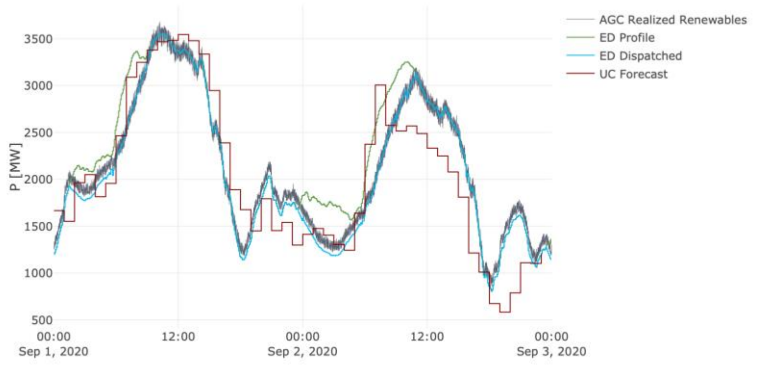


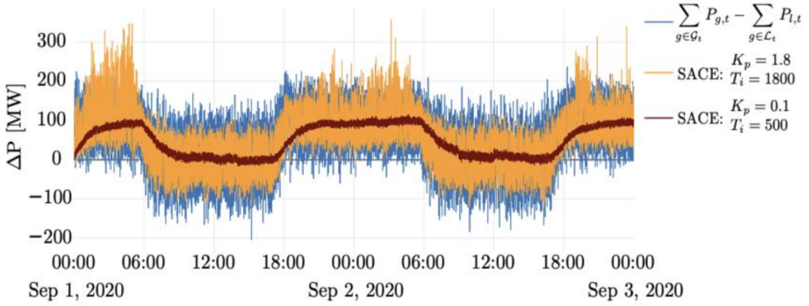
Variable	PLEXOS UC	PLEXOS ED	PSI UC	PSI ED
Gas - CC [MW]	5.26228×10^5	5.06635×10^5	5.29018×10^5	5.05629×10^5
Combustion Turbine [MW]	1751.0	1807.75	1651.0	1550.35
Hydropower [MW]	2.21594×10^5	2.21534×10^5	2.21594×10^5	2.21534×10^5
Nuclear [MW]	1.44×10^5	1.44×10^5	1.44×10^5	1.44×10^5
Steam [MW]	6.15146×10^5	6.142×10^5	6.13056×10^5	6.14076×10^5
Renewables [MW]	3.90612×10^5	3.53815×10^5	3.90013×10^5	3.54877×10^5
Total Cost [$]	5.56758×7	$2.708285 \times 7^{\dagger}$	5.56247×7	$2.68851 \times 7^{\dagger}$

†The total cost comparison for the ED stage is done for the fuel cost only due to the reporting from PLEXOS

PowerSimulations.jl for reserve deployment modeling

- Model the allocation and deployment mechanisms of Frequency Regulation Reserve (FRR) in a quasi-steady state model.
- Use mathematical programming to pose it as an optimization problem.
- Use fast optimization solvers to accelerate the studies.
- Evaluate a study case with a Unit Commitment (UC) model, an Economic Dispatch (ED), and our proposed AGC formulation.





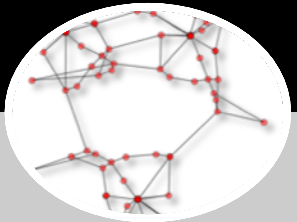
Lara, José Daniel, et al. "AGC Simulation Model for Large Renewable Energy Penetration Studies." *2020 52nd North American Power Symposium (NAPS)*. IEEE, 2021.

Scalable Integrated Infrastructure Planning for Power Systems

SIIP::Power

PowerSystems.jl
PowerSimulations.jl
PowerSimulationsDynamics.jl

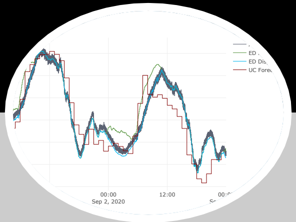
NREL-SIIP



PowerSystems.jl

Rigorous power system data model:

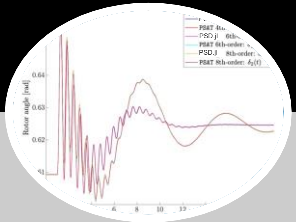
- Parsers
- Time series
- Quasi-static model data
- Dynamic model data
- Basic power-flow calculations



PowerSimulations.jl

Mathematical formulations and simulation assemblies:

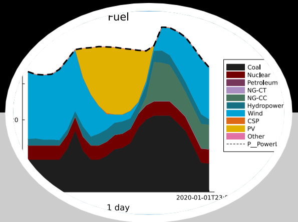
- Quasi-static problems and simulations
- PCM, UC/ED, OPF
- Reserve co-optimization
- AGC/ACE simulation
- Integrated with PowerModels.jl



PowerSimulations Dyanmics.jl

Scalable stability modeling:

- Advanced AD
- Small signal stability
- Full dynamic simulations
- Low inertia simulation capabilities
- Modular separation between device model and numerical integrator



PowerGraphics.jl

Lightweight interactive visualizations:

- Extensible and configurable graphics
- Interactive visualizations with PlotlyJS
- Supports results generated with PowerSimualtions.jl

Extensions

SIIPExamples.jl

PowerModelsInterfaces.jl

HydroPowerSimulations.jl

PowerSimulationsDemand Response.jl

ReliablePowerSimulations.jl

PowerSystemCaseBuilder.jl

Outcomes and Lessons

Clayton Barrows
Senior Engineer, NREL

Need for better data specifications and formats

- Common Information Model (CIM) designed for operational data exchange
- Model specific (MATPOWER, PSS/e .raw) formats are incomplete
- Is there a need for something in-between?
  - RTS-GMLC (not well defined, but could be a starting point)

Outcomes

- PyPSA2PowerSystems.jl
- PowerSystems.jl -> PowerModels.jl -> MATPOWER .m
- RTS-GMLC -> PowSyBl-Metrix
- RTS-GMLC -> PyPSA
- RTS-GMLC -> pandapower
- Improved pandapower -> PyPSA

Q&A

globalpst.org/

GLOBAL PST
CONSORTIUM

Thank you!

globalpst.org/

GLOBAL PST
CONSORTIUM

Sample panel questions

- What aspects of each tool demonstration are open/closed source?
- How do you obtain support?
- What can be done to improve these types of workflows?
- Is there a need for a "planning" data specification?
  – Yes, (partially) addressed by PEMMDB for planning in Europe (pan-european market model database)
  – Opportunities for coordination on defining specification