Modeling at the Edge: Navigating Trade-offs in Energy System Analysis

The RESILIENT experience

Antonio Frangioni[1]

Davide Fioriti[2]

. . . & many others

[1]Dipartimento di Informatica, Università di Pisa

[2]Dipartimento di Ingegneria dell'Energia, dei Sistemi, del Territorio e delle Costruzioni, Università di Pisa

Energy Innovation Summit – EIS2025

Berlin, June 25, 2025

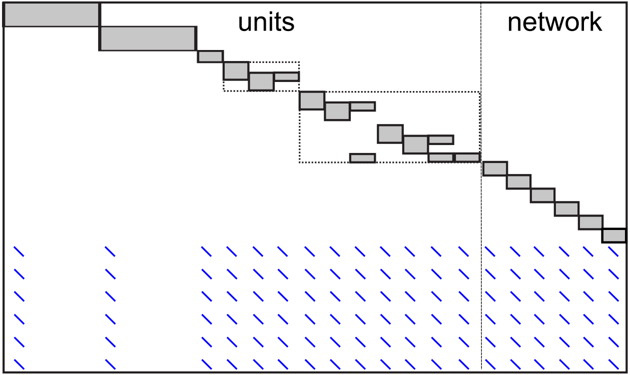
Outline

Energy Optimization bottom-up: Unit Commitment

- I don't need to convince you that energy optimization is important, but which of the many energy optimization problems??
- Operational level = Unit Commitment[1]: schedule generating units over time horizon (hours / 15m in day / week) to satisfy (forecasted) demand
- Different types of production units, different constraints:
  - Thermal (comprised nuclear): min / max production, min up / down time, ramp rates on production increase / decrease, start-up cost depending on previous downtime, others (modulation, . . .)
  - Hydro (valleys): min / max production, min / max reservoir volume, time delay to get to the downstream reservoir, others (pumping, . . .)
  - Non programmable (ROR hydro) intermittent units (solar / wind, . . .)
  - Fancy things (small-scale storage, demand response, smart grids, . . .)
- Plus the interconnection network (AC / DC, transmission / distribution, OTS, . . .) and reliability (primary / secondary reserve, $n - 1$ units, . . .)

[1] van Ackooij, Danti Lopez, F., Lacalandra, Tahanan "Large-scale Unit Commitment Under Uncertainty [. . .]" *AOR* 2018

Many Different Structures Already

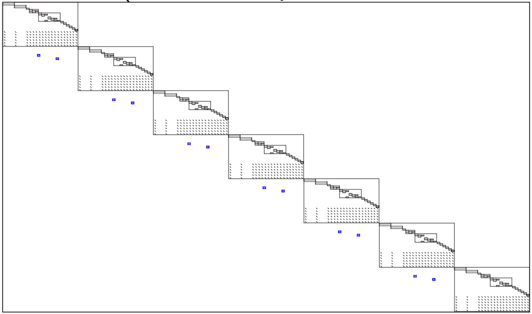


- Many different structures: thermal units[2], hydro units[3], Energy Communities[4], stochastic[5], AC-OPF[6], OTS[7], ...

[2] Bacci, F., Gentile, Tavlaridis-Gyparakis "New MINLP Formulations for the Unit Commitment Problem [...]" *OR* 2024
[3] van Ackooij et. al. "Shortest path problem variants for the hydro unit commitment problem" *Elec. Notes Disc. Math.* 2018
[4] Fioriti, F., Poli "Optimal Sizing of Energy Communities with Fair Revenue Sharing [...]" *Applied Energy* 2021
[5] Scuzziato, Finardi, F. "Comparing Spatial and Scenario Decomposition for Stochastic [...]" *IEEE Trans. Sust. En.* 2018
[6] Bienstock, Escobar, Gentile, Liberti "[...] formulations for the alternating current optimal power flow" *Ann. O.R.*, 2022
[7] Numan et. al. "The role of optimal transmission switching in enhancing grid flexibility: A review" *IEEE Access*, 2023

The tactical level: Seasonal Storage Valuation

- Mid-term (1y) cost-optimal management of water levels in reservoirs considering uncertainties (inflows, temperatures, demands, . . .)



- Very large size, nested structure (one UC per stage)
- Perfect structure for Stochastic Dual Dynamic Programming[8,9]
- SDDP needs dual variables, but Lagrangian dual convexifies[10,11]

[8] Pereira, Pinto "Multi-stage stochastic optimization applied to energy planning" *Math. Prog.*, 1991

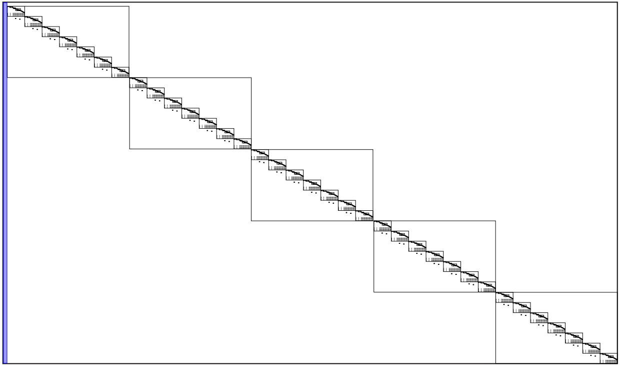
[9] van-Ackooij, Warin "On conditional cuts for Stochastic Dual Dynamic Programming" *EURO J. on Comp. Opt.*, 2020

[10] Lemaréchal, Renaud "A geometric study of duality gaps, with applications" *Math. Prog.* 2001

[11] F. "About Lagrangian Methods in Integer Optimization" Annals of O.R., 2005

Energy System Investment

- Investment on generating units / transmission lines
- Using stochastic independent representative years to evaluate system cost



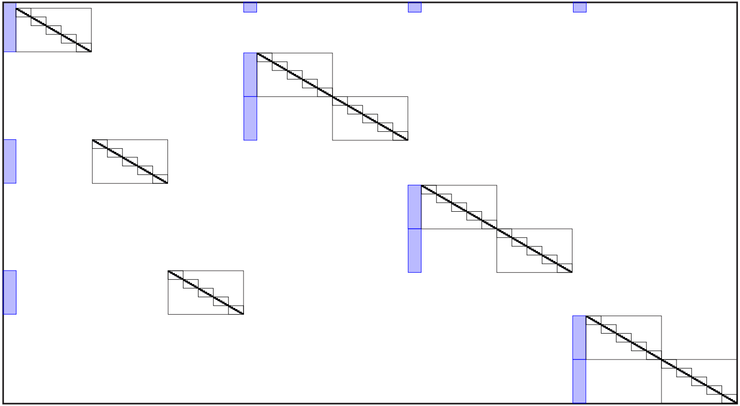
- Very few investment variables, can be taken continuous, identical copies
- Would be perfect for Benders'-like[12,13], if given dual information

[12] Geoffrion "Generalized Benders Decomposition" *JOTA*, 1972

[13] van Ackooij, F., de Oliveira "Inexact Stabilized Benders' Decomposition Approaches [...]" *COAP*, 2016

Strategic Energy System Investment

- Long-term (30y) optimal (cost, pollution, CO_2 emissions, ...) planning of production/transmission investments considering multi-level uncertainties scenarios (technology, economy, politics, ...)



- Many scenarios, huge size, multiple nested structure $\Longrightarrow$ multiple nested Benders' or Lagrangian decomposition and/or SDDP??

Outline

A HUGE LOT OF
ELBOW GREASE,
BLODSHED AND TEARS

or

Quite a lot of elbow grease and

https://gitlab.com/smspp/smspp-project

"For algorithm developers, from algorithm developers"

- Open source (LGPL3)
- 1 "core" repo, 1 "umbrella" repo, 12+ problem and/or algorithmic-specific repos (public, more in development), tests & tools
- Extensive Doxygen documentation https://smspp.gitlab.io

What SMS++ is

- A core set of `C++-20` classes implementing a modelling system that:
  - explicitly supports the notion of `Block` $\equiv$ nested structure
  - separately provides "semantic" information from "syntactic" details (list of constraints/variables $\equiv$ one specific formulation among many)
  - allows exploiting specialised `Solver` on `Block` with specific structure
  - manages any dynamic change in the `Block` beyond "just" generation of constraints/variables
  - supports reformulation/restriction/relaxation of `Block`
  - has built-in parallel processing capabilities
  - should be able to deal with almost anything (bilevel, PDE, ...)
- An hopefully growing set of specialized `Block` and `Solver`
- In perspective an ecosystem fostering collaboration and code sharing: a community-building effort as much as a (suite of) software product(s)

Outline

It starts deceivingly simple

`ThermalUnitBlock : Block`

$$\min \sum_{i \in P} c^i(\mathbf{p}^i, \mathbf{u}^i) = \sum_{i \in P} \left(c^i(\mathbf{u}^i) + \sum_{t \in \mathscr{T}} c^i_t(p^i_t) \right)$$

$$\bar{p}^i_{\min} u^i_t \leqslant p^i_t \leqslant \bar{p}^i_{\max} u^i_t \quad t \in \mathscr{T}$$

$$p^i_t \leqslant p^i_{t-1} + u^i_{t-1}\Delta^i_+ + (1 - u^i_{t-1})\bar{l}^i \quad t \in \mathscr{T}$$

$$p^i_{t-1} \leqslant p^i_t + u^i_t \Delta^i_- + (1 - u^i_t)\bar{u}^i \quad t \in \mathscr{T}$$

$$u^i_t \geqslant u^i_r - u^i_{r-1} \quad t \in \mathscr{T}, \quad r \in [t - \tau^i_+, t-1]$$

$$u^i_t \geqslant 1 - u^i_{r-1} - u^i_r \quad t \in \mathscr{T}, \quad r \in [t - \tau^i_-, t-1]$$

$$u^i_t \in \{0, 1\} \quad t \in \mathscr{T}$$

```
T   a[] b[] c[]  Dp[]
p_min[] p_max[] Dm[]
```

- For a man with a hammer everything is a nail

It starts deceivingly simple

`ThermalUnitBlock : Block`

$$\min \sum_{i \in P} c^i(\mathbf{p}^i, \mathbf{u}^i) = \sum_{i \in P} \left(c^i(\mathbf{u}^i) + \sum_{t \in \mathscr{T}} c^i_t(p^i_t) \right)$$

$$\bar{p}^i_{\min} u^i_t \leqslant p^i_t \leqslant \bar{p}^i_{\max} u^i_t \quad t \in \mathscr{T}$$

$$p^i_t \leqslant p^i_{t-1} + u^i_{t-1} \Delta^i_+ + \left(1 - u^i_{t-1}\right) \bar{l}^i \quad t \in \mathscr{T}$$

$$p^i_{t-1} \leqslant p^i_t + u^i_t \Delta^i_- + \left(1 - u^i_t\right) \bar{u}^i \quad t \in \mathscr{T}$$

$$u^i_t \geqslant u^i_r - u^i_{r-1} \quad t \in \mathscr{T}, \quad r \in \left[t - \tau^i_+, t - 1\right]$$

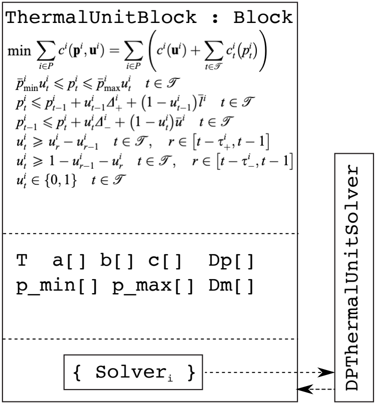
$$u^i_t \geqslant 1 - u^i_{r-1} - u^i_r \quad t \in \mathscr{T}, \quad r \in \left[t - \tau^i_-, t - 1\right]$$

$$u^i_t \in \{0, 1\} \quad t \in \mathscr{T}$$

```
T   a[] b[] c[]  Dp[]
p_min[] p_max[] Dm[]
```

- For a man with a solver everything is a `Block` (call me blockhead )
- `Block` = abstract class representing the general concept of "(fragment of) mathematical model with a well-understood semantic"
- Each `:Block` a model with specific structure: `ThermalUnitBlock : Block` = a single-(thermal)-unit commitment problem

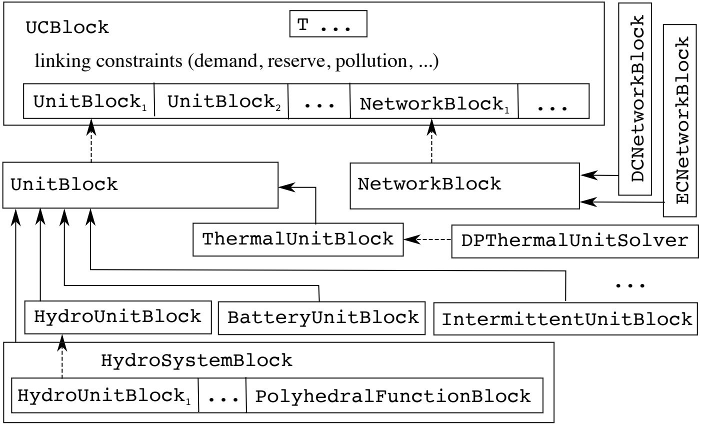
A `Block` ∃ (≈) because a (specialised) `Solver` ∃



- Any number of `Solver` can be attached to a `Block`
- Any specific `:Block` (e.g., `ThermalUnitBlock`) can have specialised $\Longrightarrow$ fast `:Solver` (e.g., `DPThermalUnitSolver`[14])
- Can be wrapper classes to efficient existing (`C++`) libraries

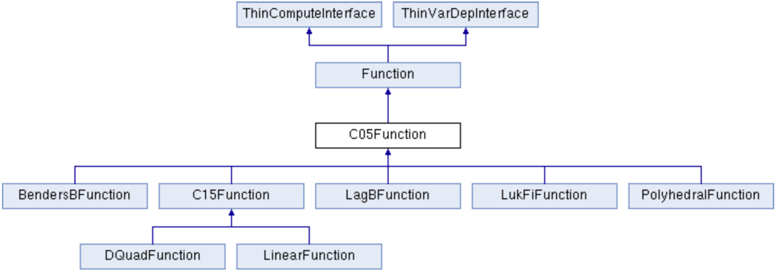
[14] F., Gentile "Solving Nonlinear Single-Unit Commitment Problems with Ramping Constraints" *Op. Res.*, 2006

A Block is (almost) always just a (small) part



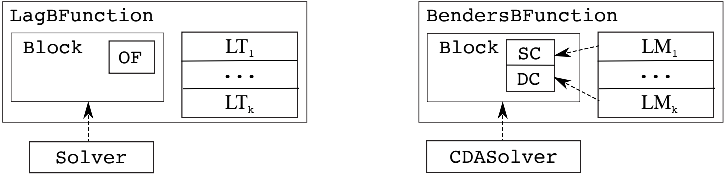
- A Block can have any # of sub-Block, recursively (Block *); e.g., UCBlock : Block has k :UnitBlock and T :NetworkBlock recursively
- Problem data split between them (energy constraints only in UCBlock)

Another necessary step: `Function`



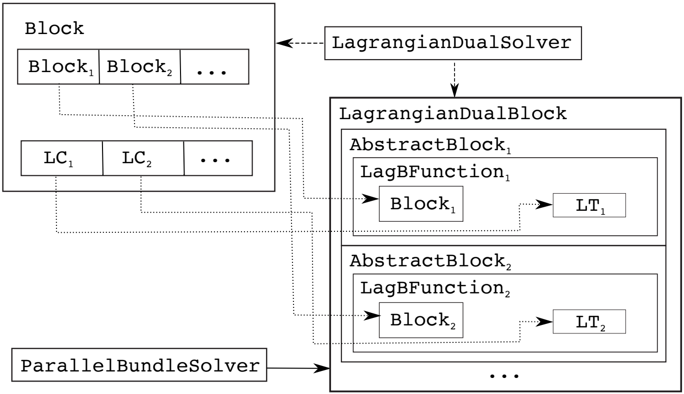
- Real-valued `Function` of a set of `[Col]Variable` (a single $\mathbb{R}$ / $\mathbb{Z}$)
- Must be `compute()`-d w.r.t. the current value of the `[Col]Variable`, possibly a costly operation (`:ThinComputeInterface`)
- `C05Function` / `C15Function` have (not necessarily continuous) 1st / 2nd order information (vertical / diagonal linearizations)
- Local / global pools of linearizations
- "Easy" `Function` (linear, quadratic, polyhedral, ...) with no overhead

LagBFunction & BendersBFunction



- `LagBFunction` $\equiv$ dual function $\varphi(\lambda) = \min\{ f(x) + (\lambda LT)x \,:\, x \in X \}$ for (almost) any `Block` (B) $\min\{ f(x) \,:\, x \in X \}$
- `BendersBFunction` $\equiv$ value function
  $v(y) = \min\{ f(x) \,:\, g(x) \leq LMy \,:\, x \in X \}$
  for (almost) any `Block` (B) $\min\{ f(x) \,:\, g(x) \leq 0 \,,\, x \in X \}$
- Both are `:Block` and `:C05Function`, with (B) being the only sub-`Block`
- Use generic `[CDA]Solver` to `compute()` ($\approx$ just call its `compute()`)
- Store pools of primal / dual `Solution` corresponding to linearizations
- Any change in (B) is mapped in changes of F-values / the pools

All this $\mapsto$ LagrangianDualSolver



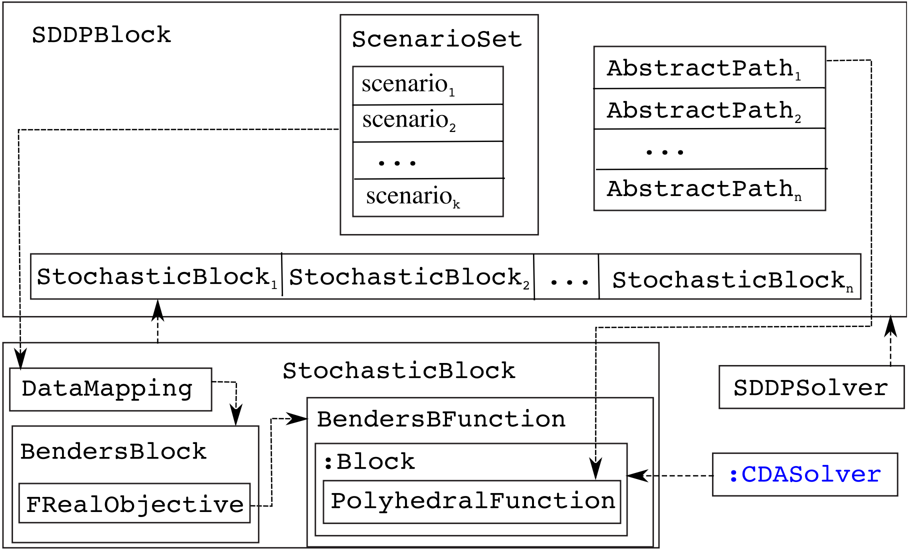
- Forms (hidden) `LagrangianDualBlock`, attaches parallel[15] `Solver`
- Provides primal (convexified $\equiv$ "better"[10,11]) and dual solutions
- Good foundations for heuristic approaches[16,17] & the next steps

[15] Cappanera, F. "[...] Parallelization of [...] Algorithm for Multi-Commodity Flow Problems" *INFORMS JoC*, 2003

[16] Borghetti, F., Lacalandra, Nucci "Lagrangian [...] for Hydrothermal Unit Commitment", *IEEE Trans. Power Sys.* 2003

[17] Scuzziato, Finardi, F. "Solving Stochastic [...] Unit Commitment with [...] Lagrangian Solutions" *IJEPES*, 2021

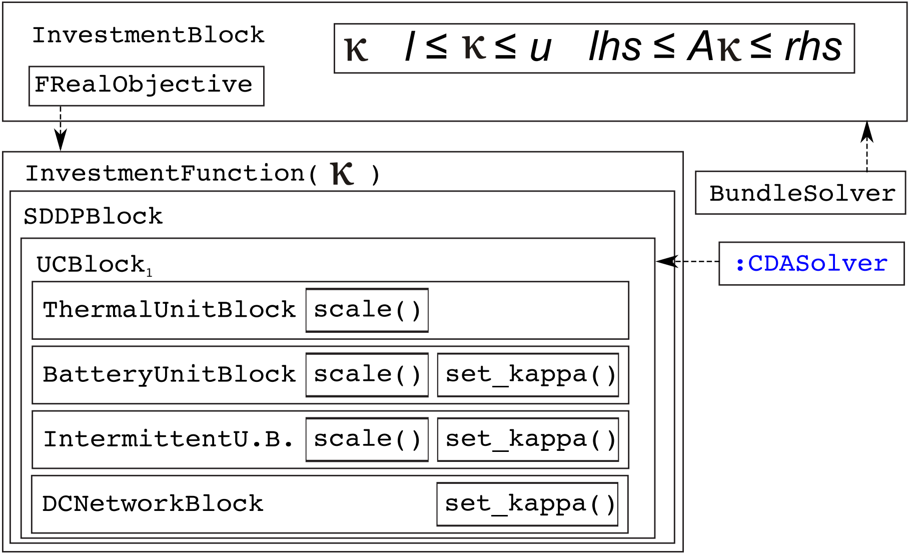
SDDPBlock, StochasticBlock and their Solver



- SDDPSolver: wrapper for StOpt[18] + SDDPGreedySolver (simulator)

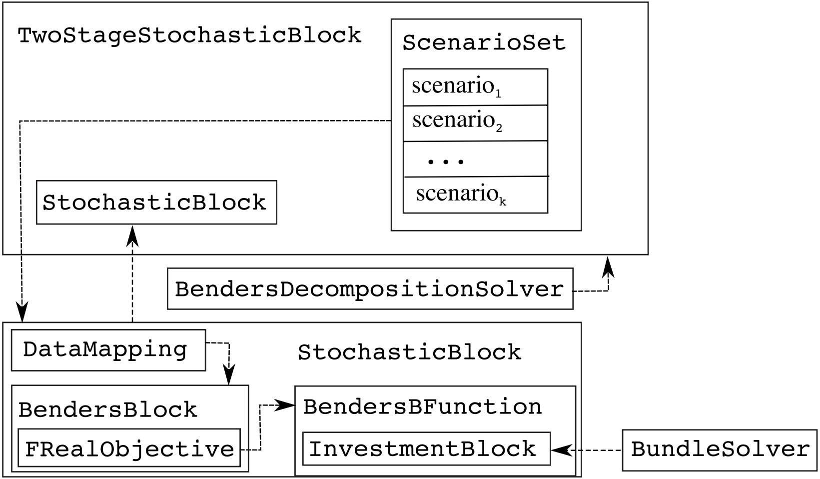
[18] https://gitlab.com/stochastic-control/StOpt

InvestmentBlock



- Scaling a :Block a general concept, may be upcasted to base Block

Strategic Investment Problem in SMS++-speak



- Not all here yet, `TwoStageStochasticBlock` still under active development, `BendersDecompositionSolver` yet to come
- Clearly extremely challenging problem, need all the help we can get

SMS++ support to (coarse-grained) parallel computations

- `Block` can be (write) `lock()`-ed to ensure atomic changes
- `lock()`-ing a `Block` automatically `lock()`s all inner `Block` (recursively)
- Analogously for `read_lock()`, any # of concurrent reads
- `lock()` (but not `read_lock()`) sets an owner and records its `std::thread::id`; other `lock()` from the same thread fail (`std::mutex` would not work there)
- Write starvation not handled yet
- `Solver`'s `compute()` must be thread-safe (`std::recursive_mutex`)
- `Solver`/`ThinComputeInterface` can be "lent ID" (solving a sub-`Block`)
- `Solver`'s `list<Modification>` under an "active guard" (`std::atomic`)
- General `State` of `Solver` for checkpointing (and reoptimization)
- New `Change` concept: `Modification` + data, automatic `undo_Change`, can be de/serialize-d on `netCDF` file as everything $\Longrightarrow$ message-passing distributed `Solver` available one day (soon-ish?)

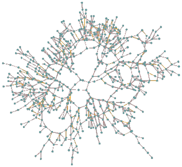
Outline

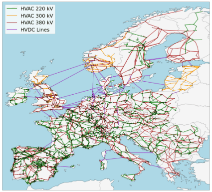
QUITE SOME ELBOW GREASE AND netCDF FILES MANGLING

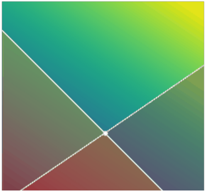
or

Some elbow grease and

PyPSA **PyPSA-Eur** **Linopy**







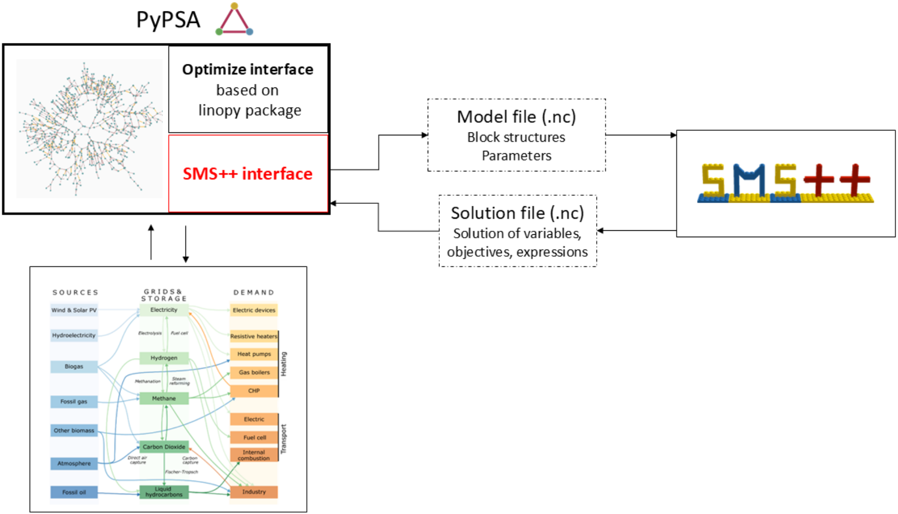
https://pypsa.org

$+$

RESILIENT

https://resilient-project.github.io

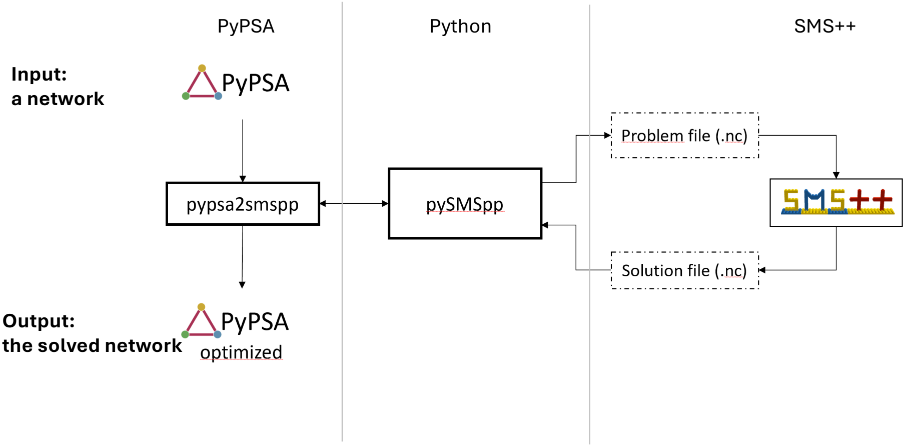
What the project wants to do



PyPSA-Eur model

- Modelling prowess of PyPSA + solution prowess of `SMS++`

A bit more in details

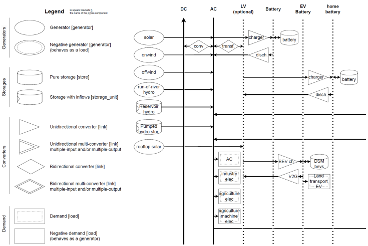


- Requires careful mapping of PyPSA network $\mapsto$ `SMS++:Block`
- Especially important for freshly developed PyPSA stochastic extension

A glimpse of the mapping

Equation	Symbol	Generator	Link	Line	Storage unit	Store	Condition	Example
Size bound	SB	X	X	X	X	X		$\underline{G}_{i,r} \leq G_{i,r} \leq \overline{G}_{i,r}$
Modularity	MD	X	X	X	X	X		$G_{i,r} = \underline{G}^{mod}_{i,r} n_{i,r}$
Power bound	PB	X	X	X	X	X		$\underline{g}_{i,r,t} G_{i,r} \leq g_{i,r,t} \leq \overline{g}_{i,r,t} G_{i,r}$
Power unit commitment	PB_{UC}	X	X				committable	$\delta_{i,r,t} \underline{g}_{i,r,t} G_{i,r} \leq g_{i,r,t} \leq \delta_{i,r,t} \overline{g}_{i,r,t} G_{i,r}$
Minimum time	MT	X	X					$\sum_{t'=t}^{t+T_{minup}} \delta_{k,t'} \geq T_{minup} (\delta_{k,t} - \delta_{k,t-1})$
Total energy produced	PSUM	X	X					$E^{sum}_{min} \leq \sum_{t \in T} w^G_t g_{i,r,t} \leq E^{sum}_{max}$
Start up/shut down cost	SC	X	X					$suc_{k,t} \geq suc_k (\delta_{k,t} - \delta_{k,t-1})$
Ramp up/down	RUD	X	X					$(g_{i,r,t} - g_{i,r,t-1}) \leq ru_{i,r} G_{i,r}$
Kirchhof's law	KL			X				$\sum_l C_{l,c} x_l p_{l,t} = 0$
Line losses	LL			X				$P^{loss}_l = \alpha_l + \beta_l p_{l,t}$
Energy storage level	ESL				X	X		$e_{i,s,t} = e_{i,s,t-1} + w^S_t h_{i,s,t}$
Energy storage bound	ESB				X	X		$0 \leq e_{i,s,t} \leq E_{i,s} e^{max}_{i,s}$
Initial energy level	IEL_S				X	X		$e_{i,s,0} = e_{i,s,init}$
Cyclic energy level	IEL_{CS}				X	X	cyclic_state_of_charge	$e_{i,s,0} = e_{i,s,\lvert T \rvert}$

	A	B	C	D	E
1	Technology name	Category	Physical compone	Option	Carrier
2	co2 atmosphere	co2	N		co2
3	Co2 storage	co2	Y		co2 stored
4	Sequestration link	co2	N		co2 sequested
5	Sequestration store (e.g. underground)	co2	Y/N		co2 sequested
6	CO2 vent co2 from storages	co2	?	co2_vent	co2 vent
7	CO2 pipelines	co2	Y	co2_network	CO2 pipeline
8	Allam (gas) cycle	electricity	Y	allam	allam
9	Direct Air Capture	co2	Y	dac	co2
10	Conventional generators	electricity	Y	conventional_generation	electricity
11	Haber-Bosch process	ammonia	Y	ammonia	Haber-Bosch
12	Ammonia cracker	ammonia	Y	ammonia	ammonia cracker
13	Ammonia storage	ammonia	Y	ammonia	ammonia store
14	Electricity distribution	electricity	Y	electricity_distribution_grid	low voltage
15	rooftop solar	electricity	Y	electricity_distribution_grid	solar rooftop



- . . . but no-one needs bother besides us (or new features developers)

Outline

What we want to hear from you

- From model users:
  - Would you really like to solve such huge problems to solve?
  - Do you believe that all that level of detail is necessary?
  - Do you believe in stochastic information / have it?
  - Would you be ready to dabble with complex solution algorithms?
- From system / algorithm developers:
  - What do you use to solve your largest and most complex problems?
  - Have you ever been set back by the lack of proper development tools?
  - Do you believe in deeper integration of models and algorithms?
- Let the fun begin!