



This project has received funding from the European Union's Horizon2020 research and innovation programme under grant agreement N° 863819

FlexPlan

16/02/2023 – EnergyVille (Genk, BE)

Network expansion planning with FlexPlan.jl

Matteo Rossini

RSE

Summary

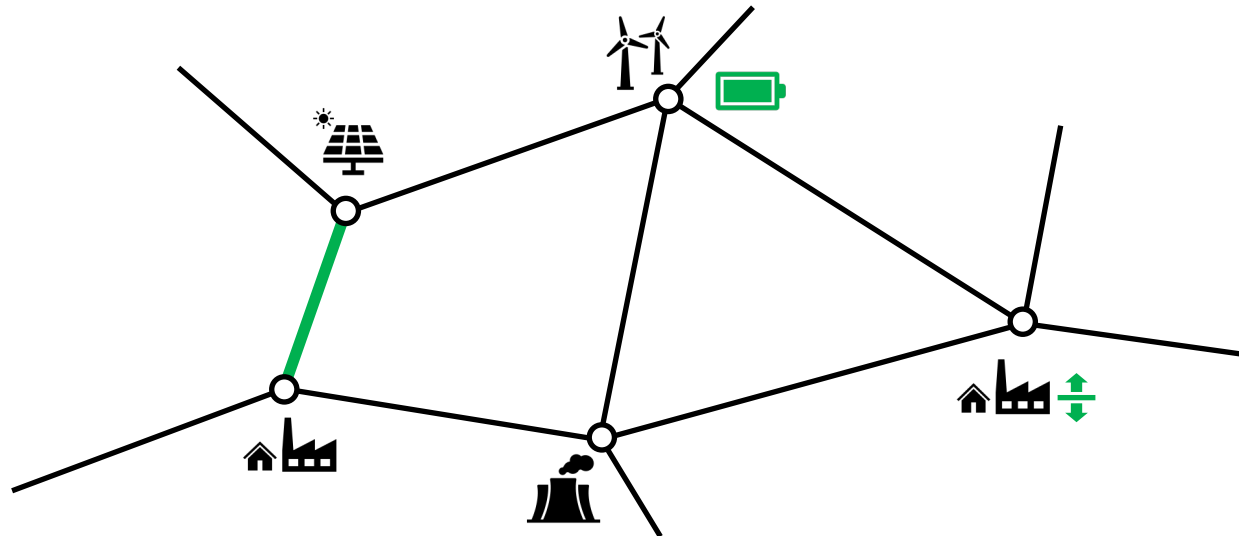
FlexPlan

1. The FlexPlan project and FlexPlan.jl
2. Overview of models
3. Insight into the implementation
4. T&D decoupling heuristics
5. Benders decomposition

The FlexPlan project

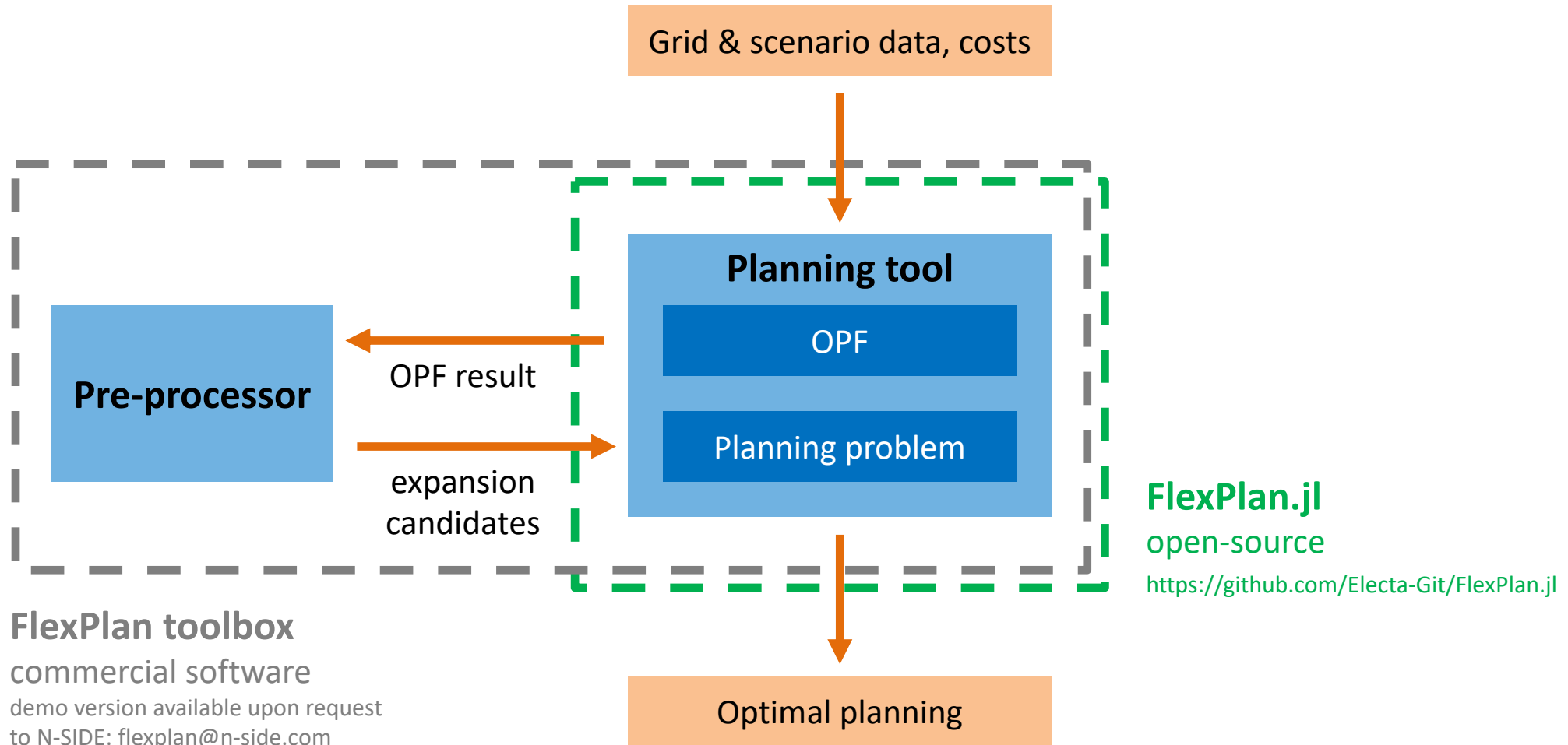
FlexPlan

Aim: *establish a new **grid planning methodology** considering the opportunity to introduce new **storage** and **flexibility resources** in T&D grids as an alternative to conventional **network expansion**.*



FlexPlan software components

FlexPlan



FlexPlan.jl is a “container” package

FlexPlan

Original content in FlexPlan.jl

- problem definition** stochastic network expansion planning
- network components** storage, flexible loads
- power flow models** simplified DistFlow (w. OLTCs and linearized apparent power bounds)
- data structures** for stochastic/multiperiod optimization
- grid interconnection model** T&D coupling through generators
 - algorithms** T&D decoupling heuristic, Benders decomposition
- test networks** 2 transmission, 2 distribution (w. candidates and stochastic time series)
- I/O functions** for data import/export and plotting

Cost minimization problem

FlexPlan

	operation periods	hour	constant continuous variables (power, voltage, ...)
Sets	scenarios	scenario	different timeseries for renewables and demand
	planning periods	year	constant binary variables (investment decisions)

Objective function

$$\min \sum_{y \in Y} \left([\text{CAPEX}]_y + \sum_{s \in S} \pi_s \sum_{t \in T} [\text{OPEX}]_{t,s,y} \right)$$

new lines, transformers, converters
new storage devices
flexibilization of demand

scenario probability

generation: cost, curtailment
demand: shifting, reduction, curtailment

Storage model

FlexPlan

$$E_{b,s,t,y} = (1 - \lambda_b)^{\Delta t} E_{b,s,t-1,y} + \Delta t \left(\eta_b^{abs} P_{b,s,t,y}^{abs} - \frac{P_{b,s,t,y}^{inj}}{\eta_b^{inj}} + \xi_{b,s,t,y} \right)$$

energy stored at time t self-discharge energy stored at time $t - 1$ power absorbed from network power injected into network power of external process

Constraints

- bounds on stored energy and absorbed/injected power
- fixed energy at beginning of planning horizon
- lower bound for energy at end of planning horizon

 No charge/discharge exclusivity constraint

Flexible load model

FlexPlan

$$P_{u,s,t,y}^d = P_{u,s,t,y}^{ref} + P_{u,s,t,y}^{sh,up} - P_{u,s,t,y}^{sh,dn} - P_{u,s,t,y}^{pred} - P_{u,s,t,y}^{curt}$$

| | | | | |

absorbed reference upward and downward voluntary involuntary
power power demand shift reduction curtailment

Constraints

- non-negative absorbed power
- bounds on demand shift and voluntary reduction as fraction of reference power
- sum of upward and downward demand shifts rebalanced periodically

Simplified DistFlow model

FlexPlan

linear approximation of branch flow model for distribution networks (requires radial topology)

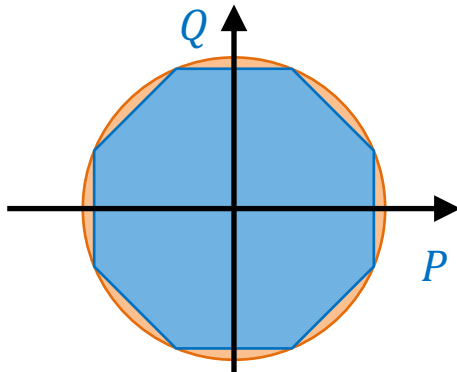
Variables

buses: squared voltage magnitude (W)

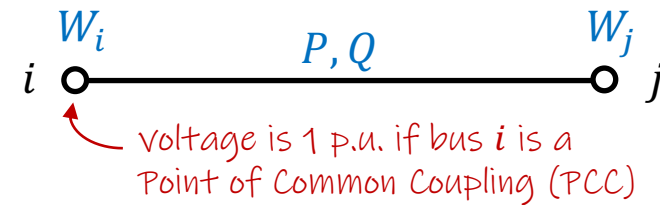
branches: active and reactive power (P, Q)

OLTCs: inverse squared transformation ratio (ρ)

Branch apparent power bound



linearized as regular octagon inscribed in $P^2 + Q^2 \leq S^2$ circle



Voltage drop constraint

line

$$W_i - W_j = 2(rP + xQ)$$

transformer connected to PCC

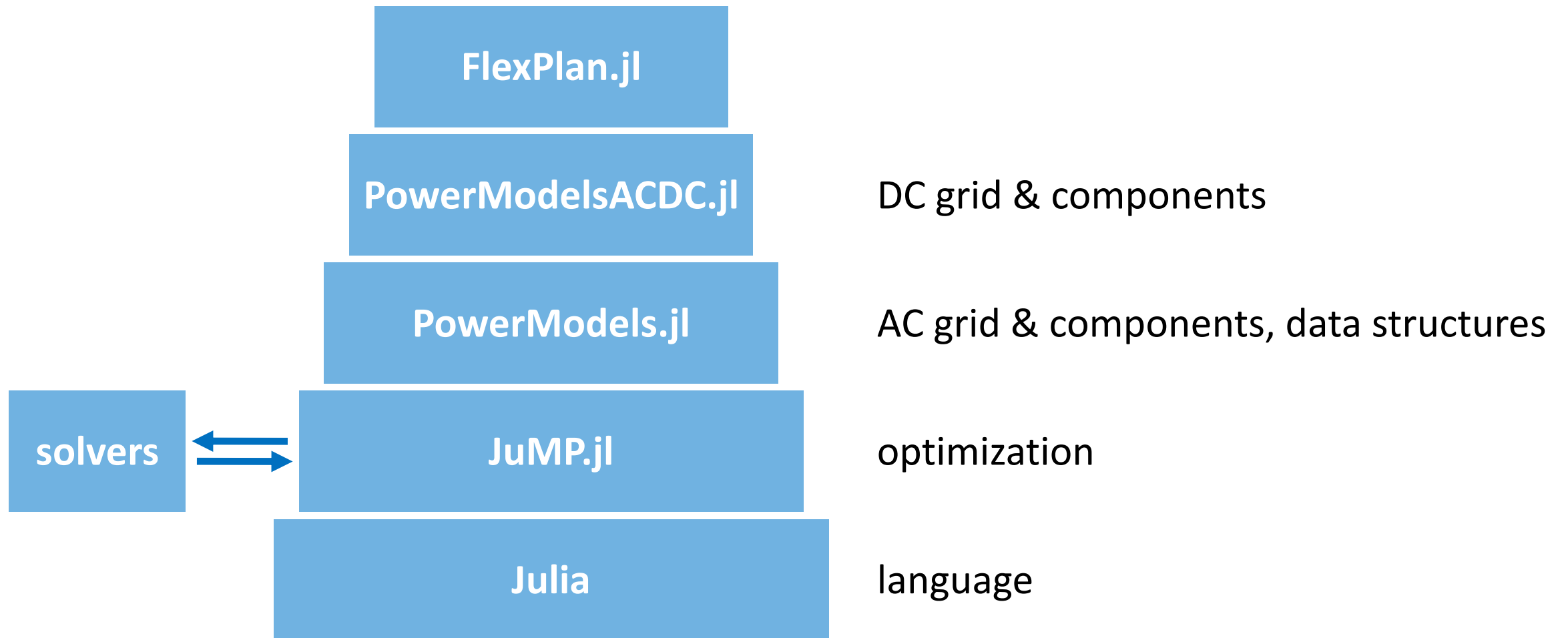
$$\frac{1}{\tau^2} - W_j = 2(rP + xQ)$$

OLTC connected to PCC

$$\rho - W_j = 2(rP + xQ)$$

Software stack

FlexPlan



Input

FlexPlan

Static data

Matpower file
with custom extensions

```
function mpc = case6
mpc.version = '2';
mpc.baseMVA = 100;
mpc.time_elapsed = 1.0;

%% buses
% bus_type pd qd gs bs bus_area vm va base_kv zone
mpc.bus = [
1 3 80 16 0 0 0 1 1.100 -0.00 240.0 1
2 1 240 40 0 0 0 1 0.935 7.26 240.0 1
3 2 40 8 0 0 0 1 0.900 -17.27 240.0 1
4 1 160 32 0 0 0 1 1.100 -0.00 240.0 1
5 1 240 40 0 0 0 1 0.935 7.26 240.0 1
6 2 0 0 0 0 0 1 0.900 -17.27 240.0 1
];

%% dispatchable generators
% gen_bus pg ng omx qsin vg nBase gen_status pmax pm
mpc.gen = [
1 148.0 54.0 48.0 -10.0 1.1 100.0 1 150.0 0.
3 170.0 -8.0 101.0 -10.0 0.9 100.0 1 180.0 0.
6 0.0 -4.0 103.0 -10.0 0.9 100.0 1 120.0 0.
];

% model startup shutdown ncost cost
mpc.genCost = [
2 0.0 0.0 2 49.6 0.0; % NG
2 0.0 0.0 2 94.1 0.0; % Coal
2 0.0 0.0 2 49.6 0.0; % NG
];

%% non-dispatchable generators
% locNum_nomatch gen_bus pref omx qsin gen_status cost_g0
mpc.ngen = [
3 100.0 101.0 -10.0 1 30.
];
```

Time series

Any file

```
Time (CET), Residential load [MW], Industrial load [MW]
01.01.2019 00:00 - 01.01.2019 00:15,42282,44200
01.01.2019 00:15 - 01.01.2019 00:30,43788,43604
01.01.2019 00:30 - 01.01.2019 00:45,43433,43205
01.01.2019 00:45 - 01.01.2019 01:00,43889,43000
01.01.2019 01:00 - 01.01.2019 01:15,42374,42660
01.01.2019 01:15 - 01.01.2019 01:30,41868,42178
01.01.2019 01:30 - 01.01.2019 01:45,41341,41608
01.01.2019 01:45 - 01.01.2019 02:00,41068,40714
01.01.2019 02:00 - 01.01.2019 02:15,40769,40524
01.01.2019 02:15 - 01.01.2019 02:30,40521,40387
01.01.2019 02:30 - 01.01.2019 02:45,40485,39715
01.01.2019 02:45 - 01.01.2019 03:00,40446,39514
01.01.2019 03:00 - 01.01.2019 03:15,40267,39792
01.01.2019 03:15 - 01.01.2019 03:30,40219,39432
01.01.2019 03:30 - 01.01.2019 03:45,40287,39381
01.01.2019 03:45 - 01.01.2019 04:00,40273,39356
01.01.2019 04:00 - 01.01.2019 04:15,40420,39358
01.01.2019 04:15 - 01.01.2019 04:30,40580,39298
01.01.2019 04:30 - 01.01.2019 04:45,40674,39295
01.01.2019 04:45 - 01.01.2019 05:00,40734,39335
01.01.2019 05:00 - 01.01.2019 05:15,41089,39328
01.01.2019 05:15 - 01.01.2019 05:30,40898,38956
01.01.2019 05:30 - 01.01.2019 05:45,40917,38801
01.01.2019 05:45 - 01.01.2019 06:00,40934,38480
01.01.2019 06:00 - 01.01.2019 06:15,40969,38996
01.01.2019 06:15 - 01.01.2019 06:30,40673,38172
01.01.2019 06:30 - 01.01.2019 06:45,40895,38292
01.01.2019 06:45 - 01.01.2019 07:00,41197,38111
01.01.2019 07:00 - 01.01.2019 07:15,41861,38560
01.01.2019 07:15 - 01.01.2019 07:30,42211,38838
01.01.2019 07:30 - 01.01.2019 07:45,43715,38869
01.01.2019 07:45 - 01.01.2019 08:00,44561,39228
01.01.2019 08:00 - 01.01.2019 08:15,45789,39271
01.01.2019 08:15 - 01.01.2019 08:30,45872,39382
```

```
data = parse_file(file; kwargs...)
scale_data!(data; kwargs...)
add_dimension!(data, name, ...)
```

user-provided function

Single-network dictionary (PowerModels-like)

Dictionary with vector values

```
make_multinetwork(data, time_series; kwargs...)
```

Multinetwork dictionary (PowerModels-like)

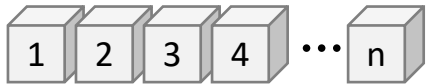
Sample networks: <https://github.com/Electa-Git/FlexPlan.jl/tree/master/test/data>

Dimensions

FlexPlan

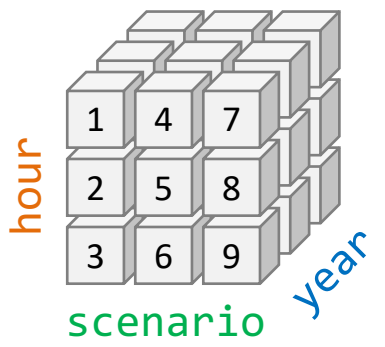
Data is stored as Dict{Int,Any}...

...but accessed as a multidimensional array



```
add_dimension!(data, :hour, 24)
add_dimension!(data, :scenario, Dict(
  1 => Dict("probability"=>0.7),
  2 => Dict("probability"=>0.3)))
add_dimension!(data, :year;
  metadata=Dict("scale_factor"=>10))
```

```
nw_ids(pm; kwargs...)
similar_ids(pm, n; kwargs...)
similar_id(pm, n; kwargs...)
first_id(pm, n, dimension...)
last_id(pm, n, dimension...)
prev_id(pm, n, dimension)
prev_ids(pm, n, dimension)
next_id(pm, n, dimension)
next_ids(pm, n, dimension)
and many other functions
```



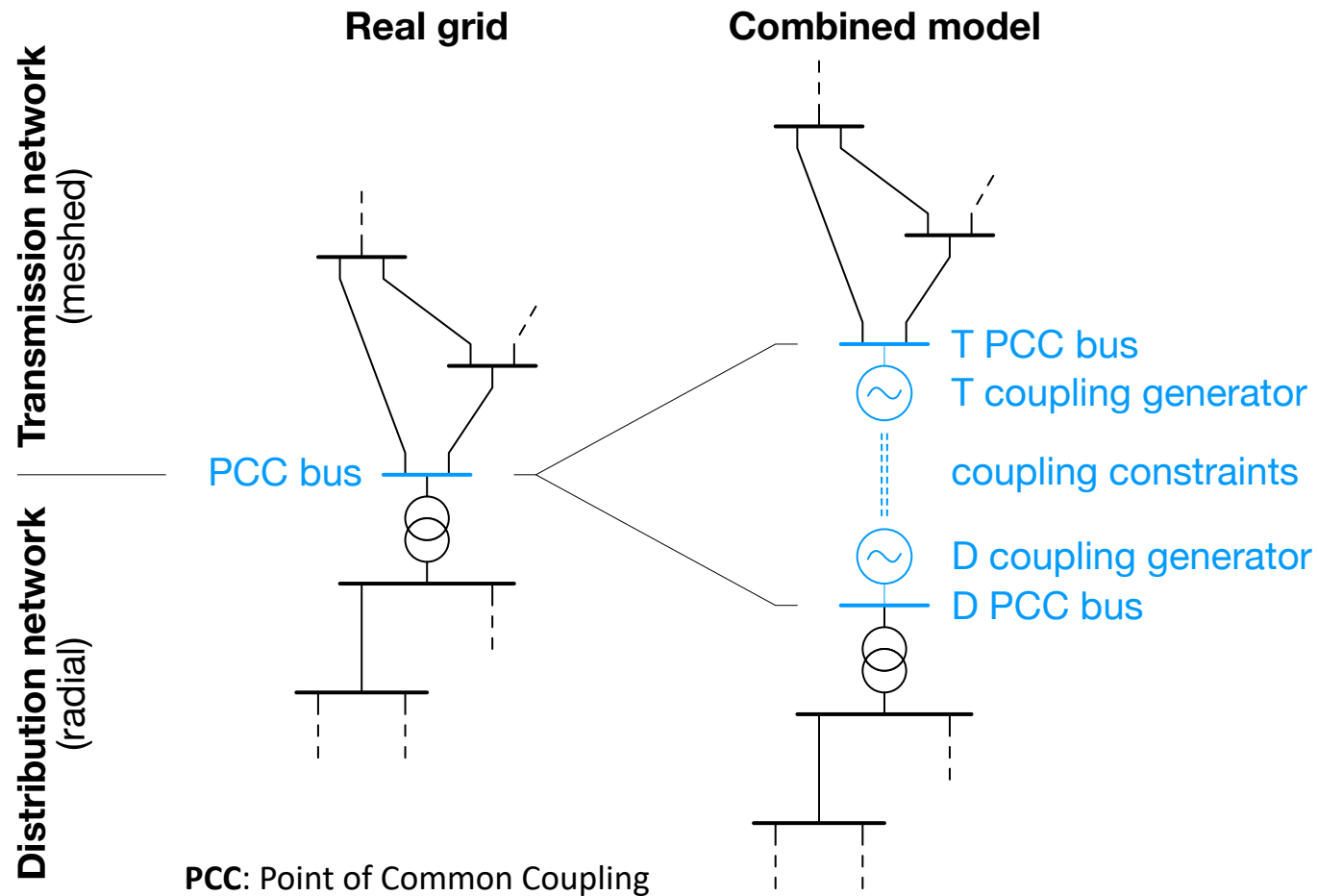
How to run optimizations

FlexPlan

```
1 import FlexPlan as _FP
2 import HiGHS
3 optimizer = _FP.optimizer_with_attributes(HiGHS.Optimizer, "output_flag"=>false)
4
5 sn_data = _FP.parse_file("my_matpower_file.m")
6 _FP.scale_data!(sn_data)
7 _FP.add_dimension!(sn_data, :hour, 24)
8 _FP.add_dimension!(sn_data, :scenario, Dict(1=>Dict("probability"=>0.7), 2=>Dict("probability"=>0.3)))
9 _FP.add_dimension!(sn_data, :year; metadata = Dict("scale_factor"=>10))
10
11 time_series = my_function_to_load_time_series()
12 mn_data = _FP.make_multinetwork(sn_data, time_series)
13
14 setting = Dict("output" => Dict("branch_flows"=>true))
15 result = _FP.simple_stoch_flex_tnep(mn_data, _FP.BFARadPowerModel, optimizer; setting)
```

Combined T&D model

FlexPlan



- ✓ Different power flow models for T&D
- ✓ Uses only standard network components
- ✓ Coupling constraints do not depend on the power flow model

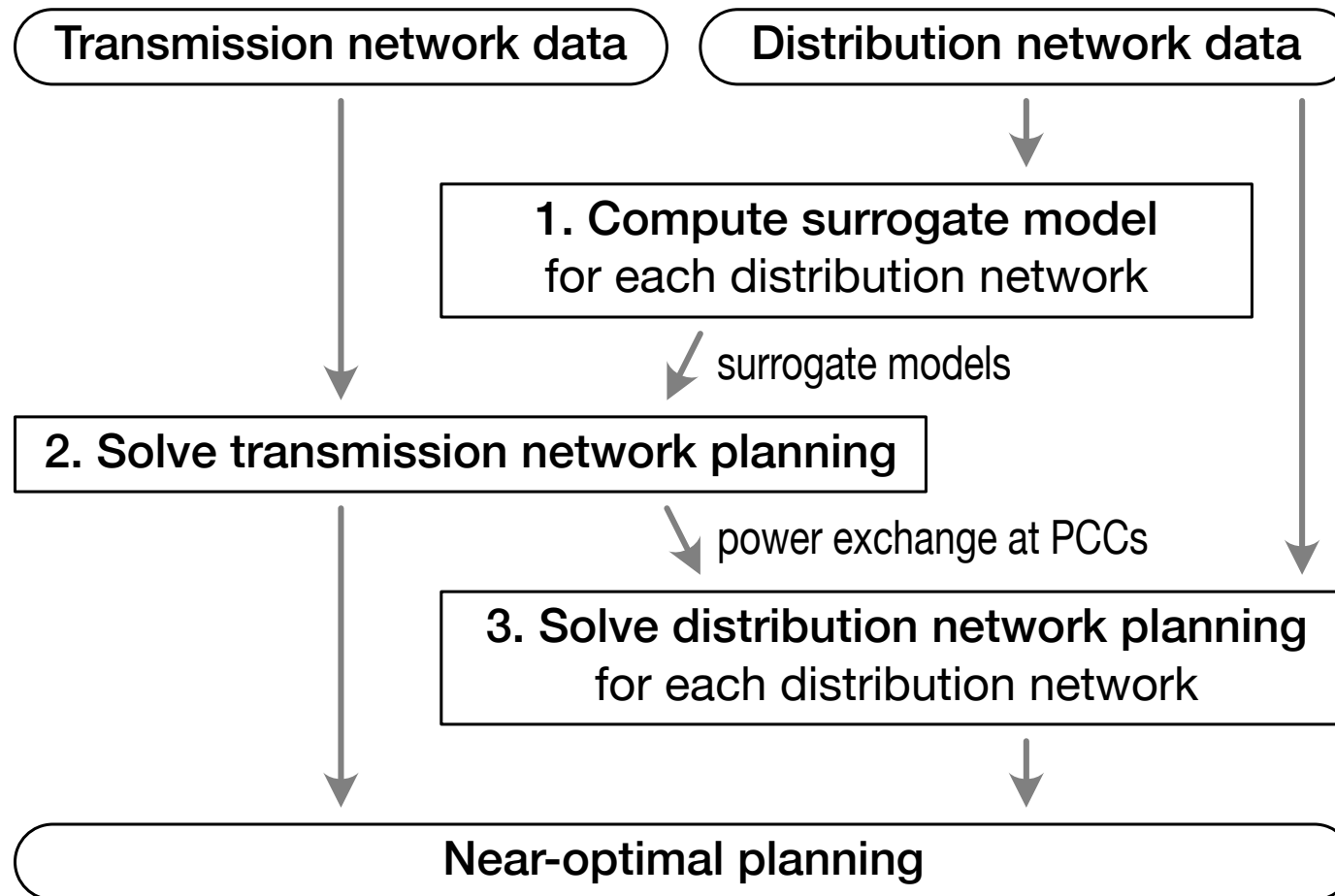
✗ Voltage variables are not coupled

Suitable for planning, not for operation

i The combined system is passed to functions as two arguments: a Dict for transmission grid and a Vector{Dict} for distribution grids

T&D decoupling heuristic

FlexPlan



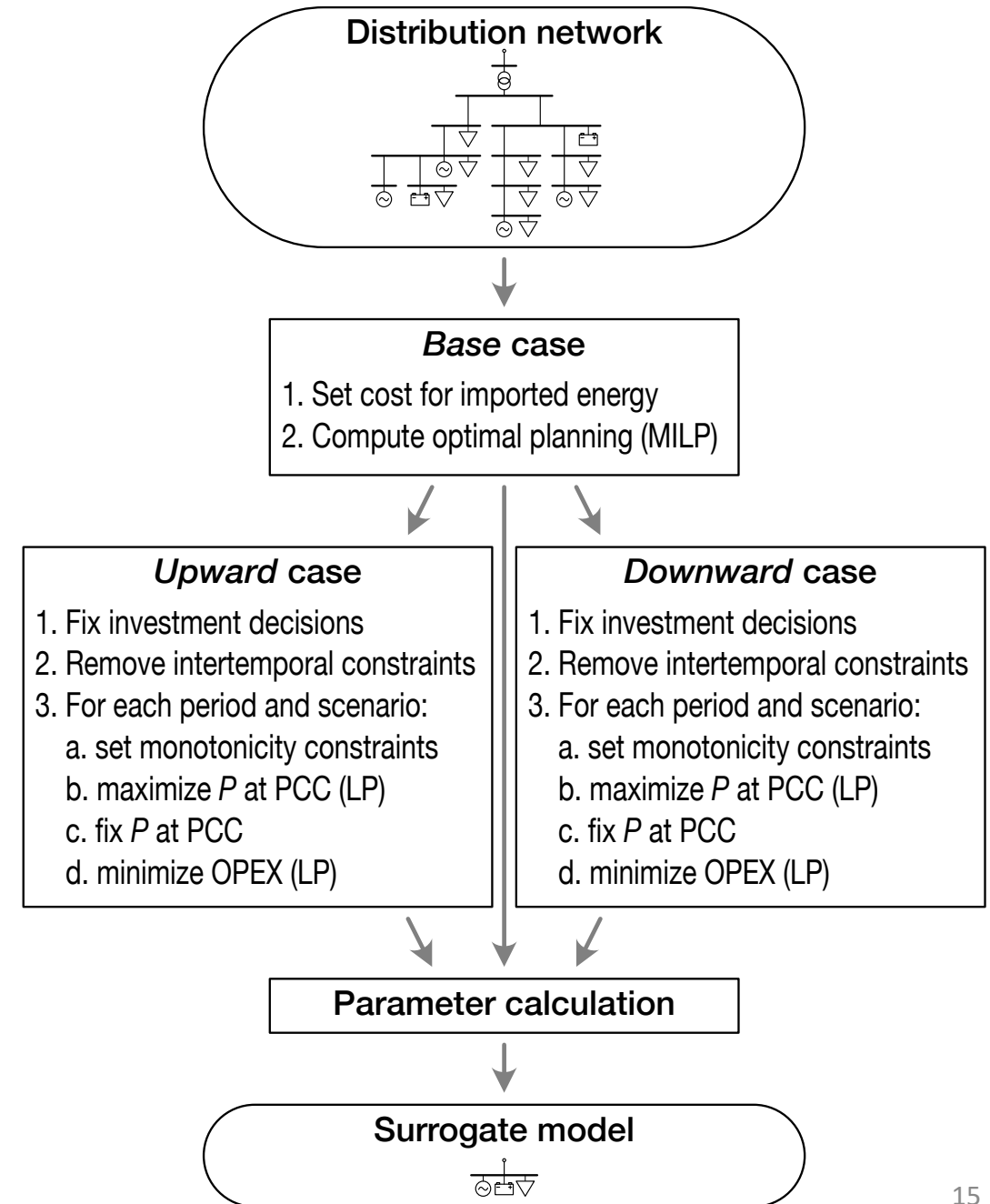
T&D decoupling heuristic

Surrogate model components:

- one generator
- one storage device
- one flexible load

with parameters such that:

- feasibility *implies* feasibility in original model
- cost *approximates* cost in original model



T&D decoupling heuristic

FlexPlan

Surrogate model assumptions

Independence of components

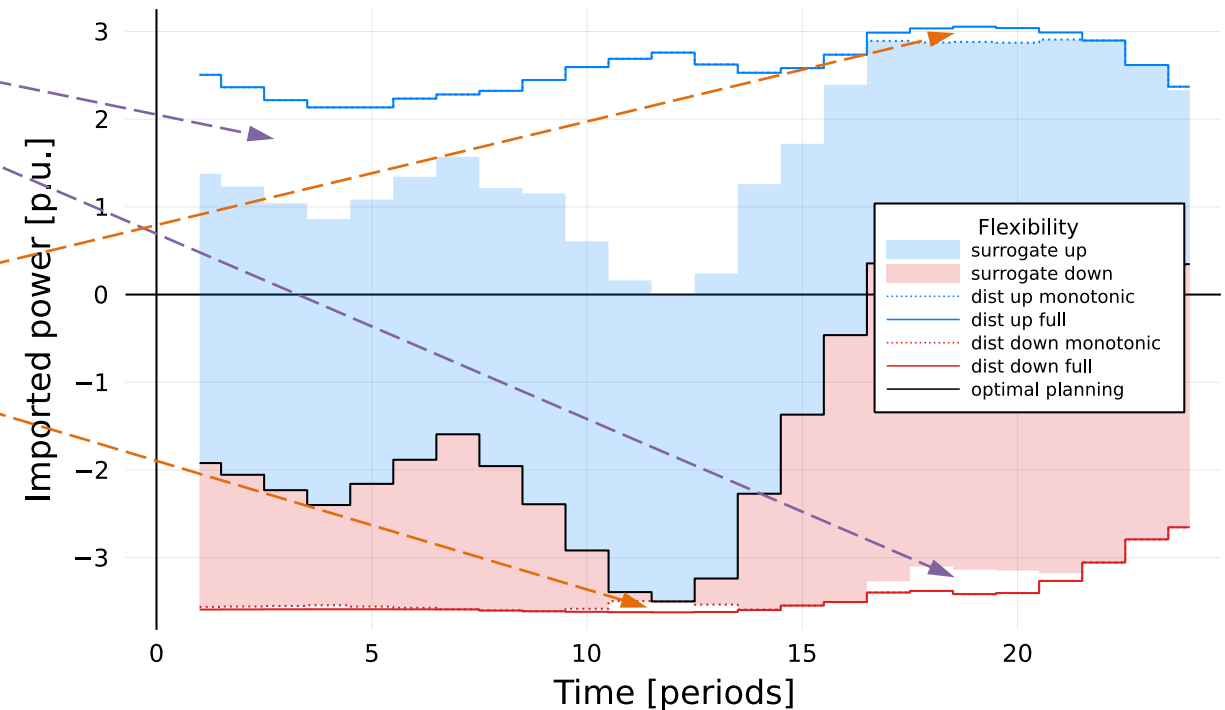
Generator, storage device and load can be used independently by transmission.

Monotonicity of power variations

A variation in power exchanged with transmission produces a variation of the same sign in components.

Example: if imported power increases, then the load cannot absorb less.

Effects on power exchange at PCC



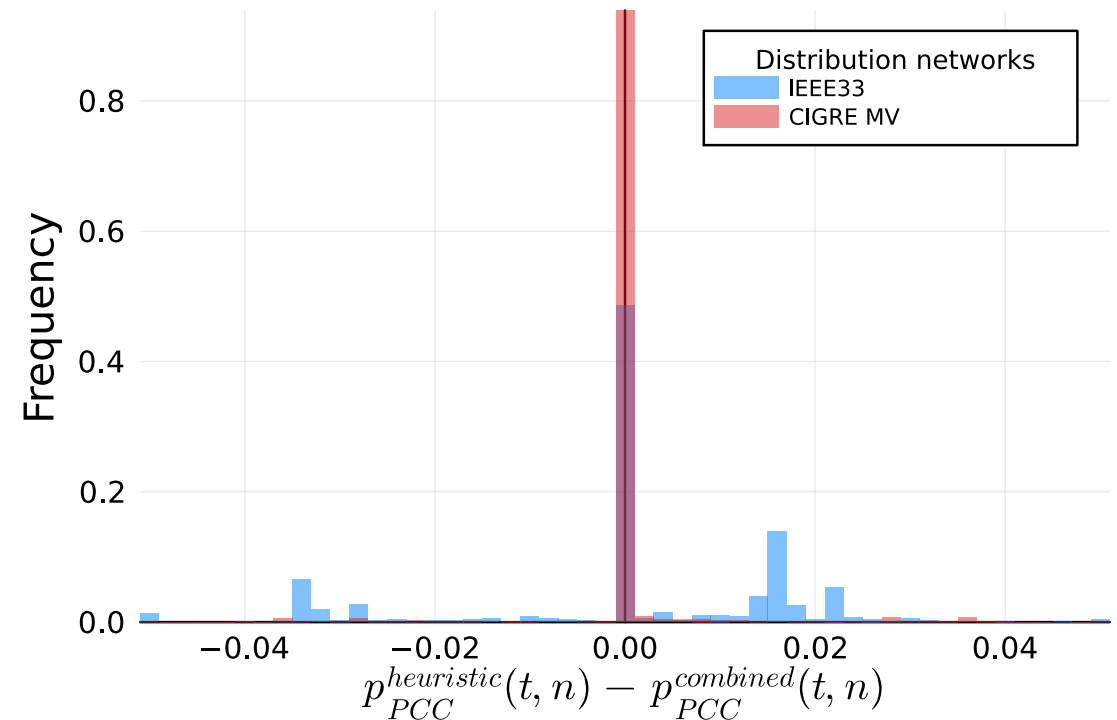
T&D decoupling heuristic

FlexPlan

Test: attach to a transmission network (case67) a variable amount N_d of distribution networks (either IEEE33 or CIGRE MV)

PERFORMANCE COMPARISON OF COMBINED MODEL AND DECOUPLING HEURISTIC

N_d	binary variables	CPU time			relative cost increase
		combined model [s]	decoupling heuristic [s]	ratio	
<i>case67 with N_d IEEE33 distribution networks</i>					
1	83	38	21	0.553	$-1.1 \cdot 10^{-5}$
4	158	148	25	0.169	$6.0 \cdot 10^{-7}$
16	458	1139	41	0.036	$1.4 \cdot 10^{-6}$
64	1658	4228	87	0.021	$6.6 \cdot 10^{-5}$
<i>case67 with N_d CIGRE MV distribution networks</i>					
1	88	39	23	0.590	$-5.3 \cdot 10^{-15}$
4	178	79	20	0.253	$6.2 \cdot 10^{-5}$
16	538	210	30	0.143	$5.3 \cdot 10^{-15}$
64	1978	6479	59	0.009	$4.1 \cdot 10^{-4}$



T&D decoupling heuristic

FlexPlan

Adopts a transmission-follows approach
(distribution investments decided in distribution, used also by transmission)

✓ Fast w.r.t. combined model

typical speedup: $10 \div 100 \times$

✓ Near-optimal result

typical relative cost increase: $< 10^{-4}$

✓ Good solution quality

typical power deviation at PCCs: $< 1\%$

✗ No voltage information shared

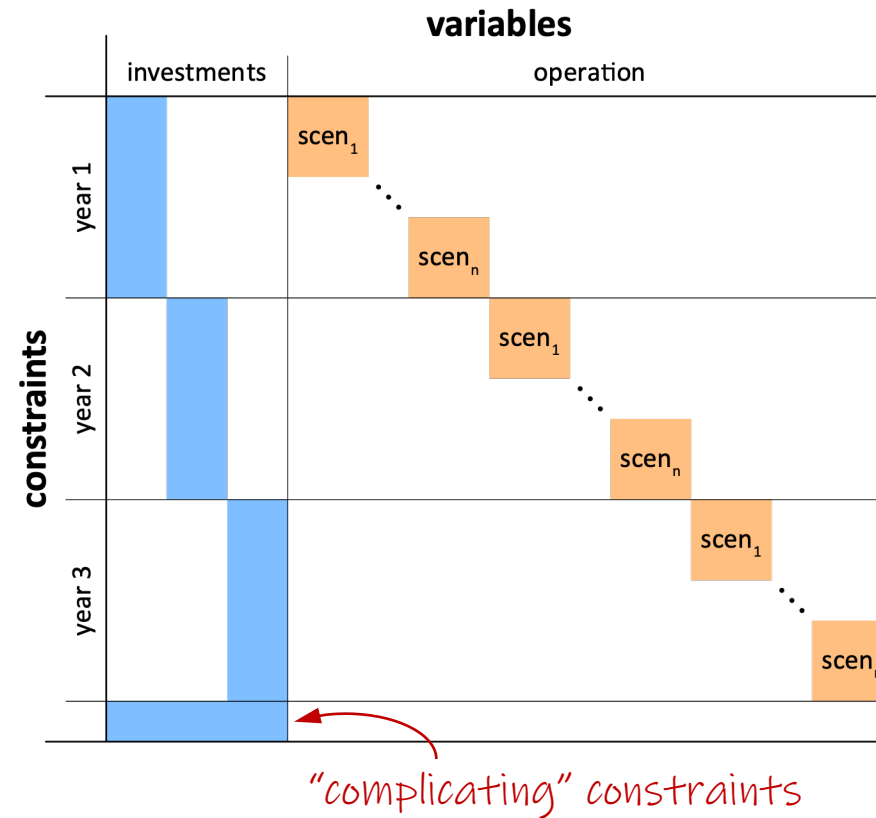
between T&D

as in combined T&D model

Benders decomposition

FlexPlan

The problem matrix has a block structure



Benders decomposition

FlexPlan

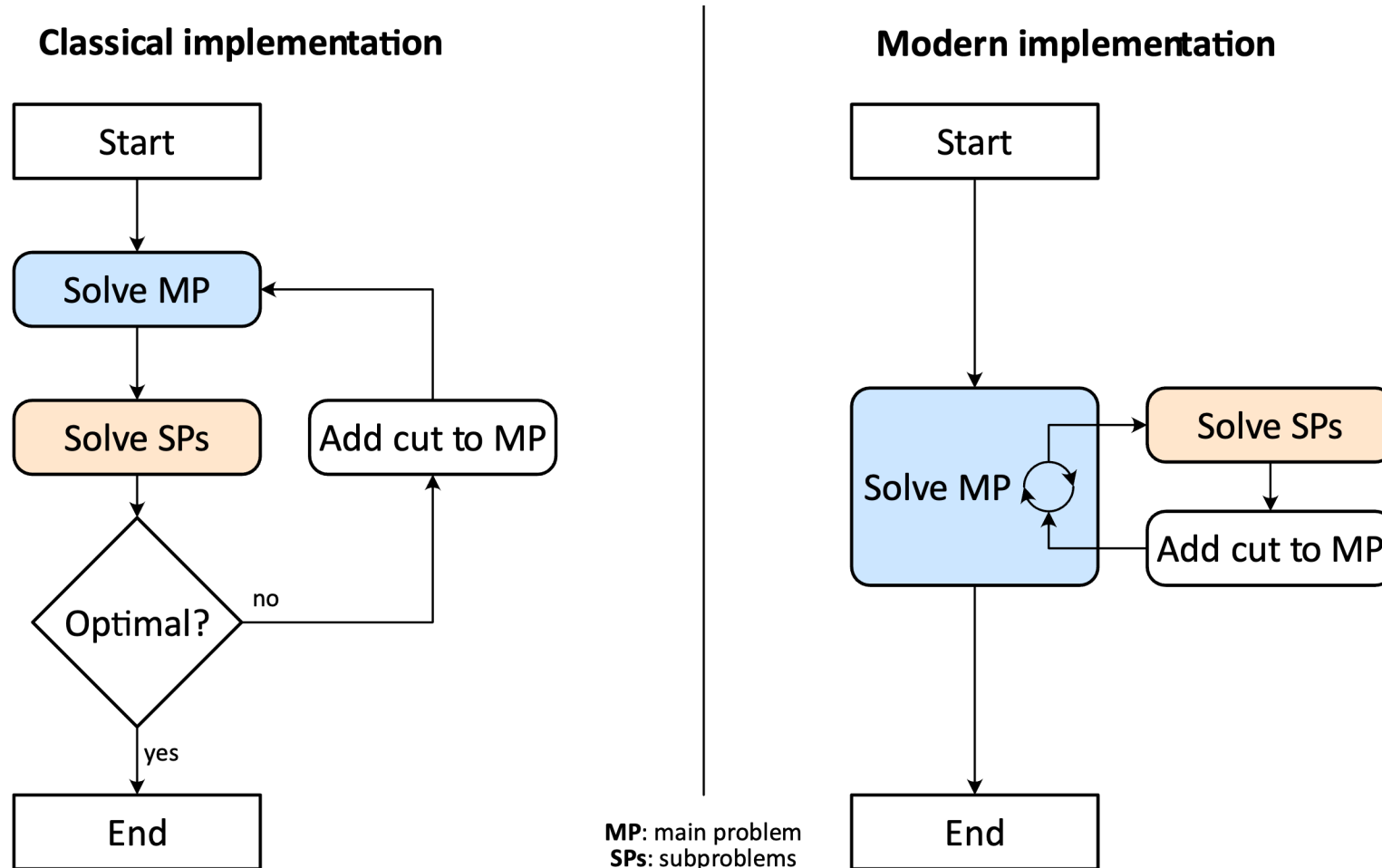
Benders algorithm (1962) applied to planning problem

Repeat until convergence:

1. Solve main problem
 - optimize investment (binary) variables
2. Solve subproblems (one per year/scenario)
 - fix investment decisions
 - optimize operation (continuous) variables
3. Add cut to main problem
 - cuts provide a lower bound of the operation cost

Benders decomposition

FlexPlan



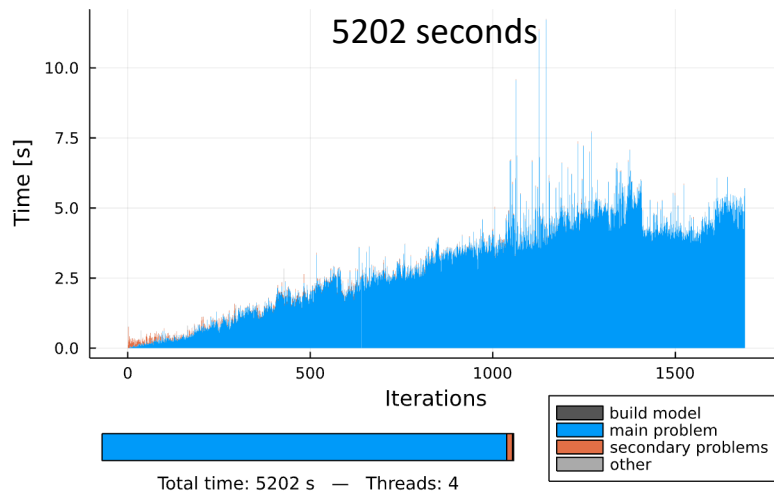
Benders decomposition

FlexPlan

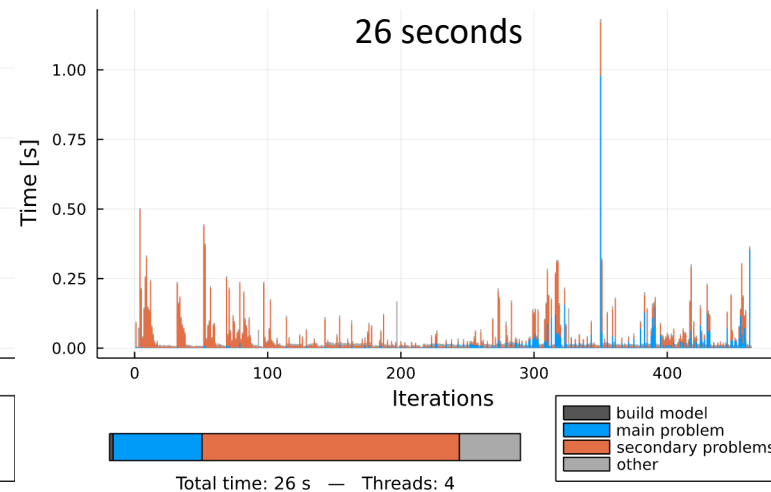
Test: comparison of solve time

Main problem: 357 binary variables and 705 constraints initially
9 subproblems: 2514 continuous variables and 8540 constraints each

Classical implementation



Modern implementation



Key features of implementation

✓ Subproblems are not dualized

- Variant of original Benders decomposition

✓ Multithreaded

- Main problem: multithreaded solver
- Subproblems: solved in parallel (multithread), with single-threaded solvers

✗ Basic implementation

- No regularization applied

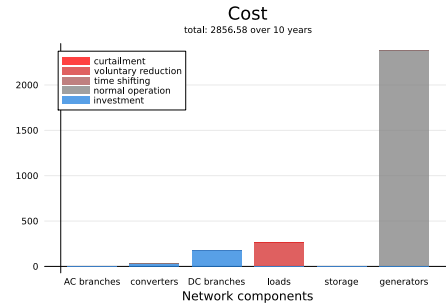


Performance of Benders decomposition implementation in FlexPlan.jl is very case-dependent. Consider using the Benders decomposition frameworks included in some solvers (SCIP, CPLEX) instead.

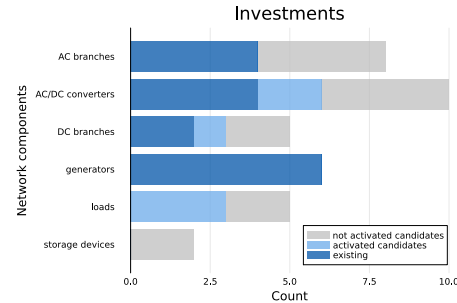
Functions for inspection of results

FlexPlan

`sol_report_cost_summary(...)`



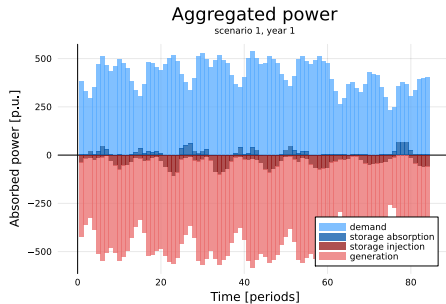
`sol_report_investment_summary(...)`



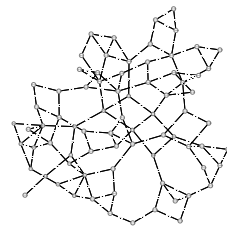
Outputs:

- **DataFrame** (returned value)
- numerical results as **CSV** files
- **plots** in any format supported by Plots.jl

`sol_report_power_summary(...)`



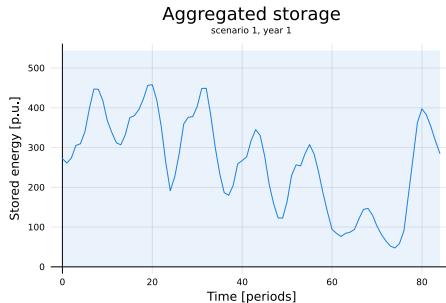
`sol_report_graph(...)`



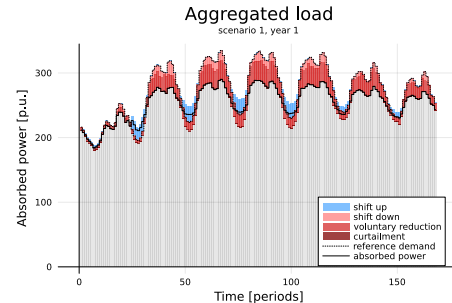
These functions are not part of FlexPlan module to avoid unwanted dependencies. Include them with:

```
import FlexPlan as _FP
const _FP_dir = dirname(dirname(pathof(_FP)))
include(joinpath(_FP_dir, "test/io/sol.jl"))
and import required packages.
```

`sol_report_storage_summary(...)`



`sol_report_load_summary(...)`



References

FlexPlan

The screenshot displays the GitHub repository for `Electa-Git/FlexPlan.jl`. The repository is public and has 728 commits. The README is open, showing the following content:

FlexPlan.jl

Status: CI passing coverage 72% Documentation passing

Overview

FlexPlan.jl is a Julia/JuMP package to carry out transmission and distribution network planning considering AC and DC technology, storage and demand flexibility as possible expansion candidates. Using time series input on renewable generation and demand, as well as a list of candidates for grid expansion, a mixed-integer linear problem is constructed which can be solved with any commercial or open-source MILP solver. Some modelling features are:

- Multi-period, multi-stage formulation to model a number of planning years, and planning hours within years for a sequential grid expansion plan
- Stochastic formulation of the planning problem, based on scenario probabilities for a number of different time series
- Linearized DistFlow model considering reactive power and voltage magnitudes for radial distribution grids
- Extensive, parametrized models for storage, demand flexibility and DC grids
- Different decomposition methods for solving the large-scale MILP problem

<https://github.com/Electa-Git/FlexPlan.jl>

To get started, see `/examples/` and `/test/scripts/` directories.

Documentation: <https://electa-git.github.io/FlexPlan.jl/dev>

FlexPlan.jl package

H. Ergun, M. Rossini, M. Rossi

FlexPlan.jl – An open-source Julia tool for holistic transmission and distribution grid planning
accepted to OSMSES 2023 – Aachen (DE), 27-29/03/2023

T&D decoupling heuristic

M. Rossini, M. Rossi, D. Siface

A surrogate model of distribution networks to support transmission network planning
submitted to CIRED 2023 – Rome (IT), 12-15/06/2023

Contact authors

Hakan Ergun – hakan.ergun@kuleuven.be

Matteo Rossini – matteo.rossini@rse-web.it

(full list of authors on repository website)

FlexPlan



FlexPlan-Project.eu

This presentation reflects only the author's view and the Innovation and Networks Executive Agency (INEA) is not responsible for any use that may be made of the information it contains.