Powsybl – Metrix

Practicals on a 6-node grid

Linux Foundation Energy

22/06/2022

Introduction to PowSyBl

- PowSyBl (Power System Blocks) is an open-source framework written in Java and dedicated to electrical model grid 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-flow analysis
Simulations, OPFs, security constraints, dynamic simulation

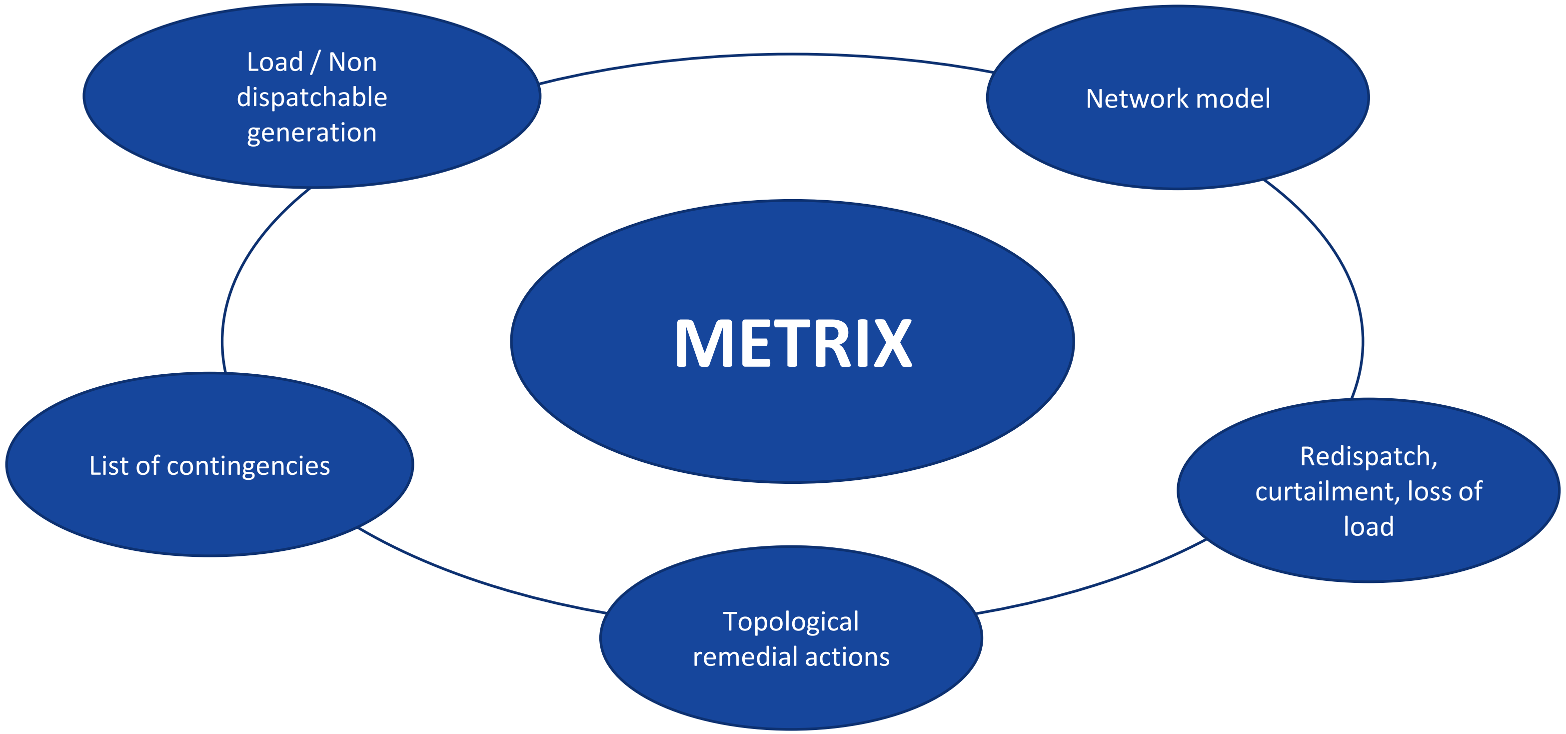
Advanced features
Analysis tools, vizualisation tools

Grid topology operations
Extensions, Extractions, Merges…

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

Introduction to Metrix (1/3)

- Metrix is an optimization model used to assess preventive and curative remedial actions to respect the network constraints on a high number of variants.
  - Created in 2010 (fully open-source including the linear optimizer since 2021)
  - Interfaced with PowSyBl



Introduction to Metrix (2/3)

- Files to launch a calculation

File name	Type	Role
`case-file`	`.iidm (.xml)`	Provides topology of the network
`metrix-dsl-file`	`.groovy`	Provides the calculation parameters and the definition of the outputs to be written into the result file
`mapping-file`	`.groovy`	Maps the timeseries to elements (ex: generation per unit for each timestep)
`time-series`	`.csv`	Provides the timeseries
`parades`	`.csv`	Provides the possible topological remedial action (parades in French) in the hand of the network manager
`contingencies-file`	`.groovy`	The list of contingencies to be covered during the computations.

- Example command using **itools**

itools metrix ***--case-file*** *data/reseau_6noeuds.xiidm* ***--mapping-file*** *data/mapping_file_gen_load.groovy* \
--contingencies-file *data/contingencies.groovy* ***--metrix-dsl-file*** *data/conf.groovy* \
--remedial-actions-file *data/parades.csv* ***--time-series*** *data/ts/time-series-tp.csv* \
--versions *1* ***--first-variant*** *0* ***--variant-count*** *3* \
--csv-results-file *results/results5E.csv* ***--chunk-size*** *3* ***--log****-archive logs* \
--network-point-file *results/output_network.xiidm*

Introduction to Metrix (3/3)

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

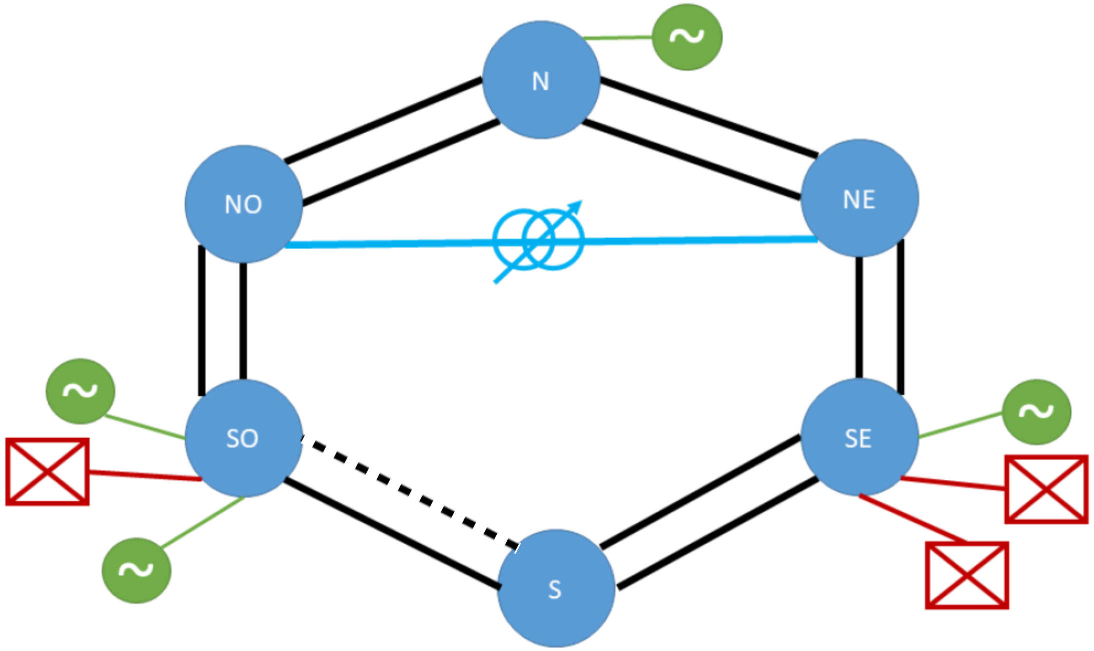
Inputs:
- *Same as DC security analysis*
- Available preventive and curative actions
  - Topological remedial actions
  - Redispatch costs

Results:
- *Same as SC-DCOPF without redispatching*
- Chosen preventive and curative actions
- Production and consumption adjustments (redispatch, curtailment, loss of load)

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

6 nodes model

- Study case building:
  - 6 nodes network
  - Mapping dispatchable generation initial set points and load timeseries (3 hourly time steps) to each node.
- Study computation:
  - N-K analysis: only one contingency (dashed line)
  - Available remedial actions: preventive and curative remedial actions (Topology, Phase-Shifter, Redispatching)
- Analysis (KPIs)
  - Flows on the lines
  - Localization of their threats (i.e. the contingency leading to the largest flow on a given line)
  - Redispatch costs

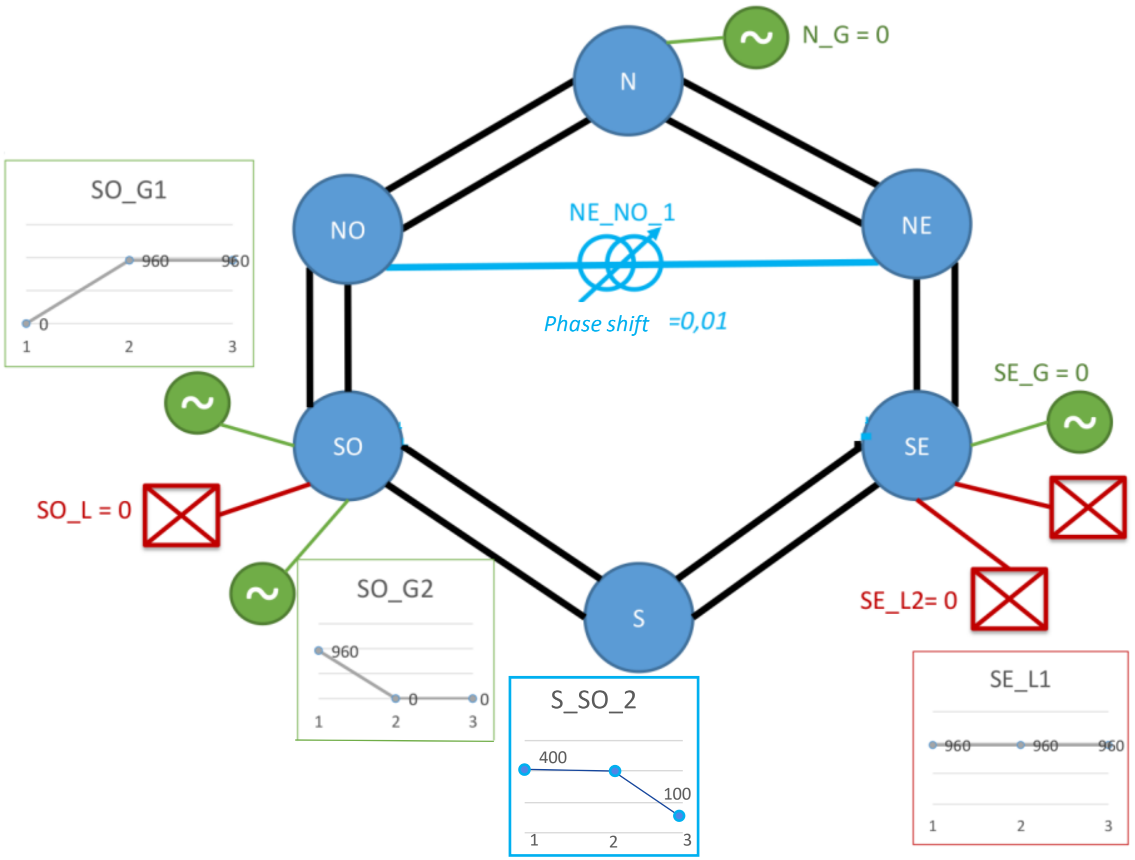


Network construction and mapping

- Mapping csv file in a table:
  - Generally speaking, any value in the iIDM grid model can be substituted by timeseries.
- 6 node model example:
  - The permanent limit of line S_SO_2 is mapped to "threshold_N" column
    - Blue box on the diagram
  - The generation of SO_G1 and SO_G2 units are respectively mapped to "SO_G1" and "SO_G2" columns
    - Green boxes on the diagram
  - The load SE_L1 is mapped to "SE_L1" columns
    - Red box on the diagram

Ts	Version	SE_L1	SO_G1	SO_G2	threshold_N
T01	1	960	0	960	400
T02	1	960	960	0	400
T03	1	960	960	0	100

- Illustration of the mapped network



SC Security Analysis

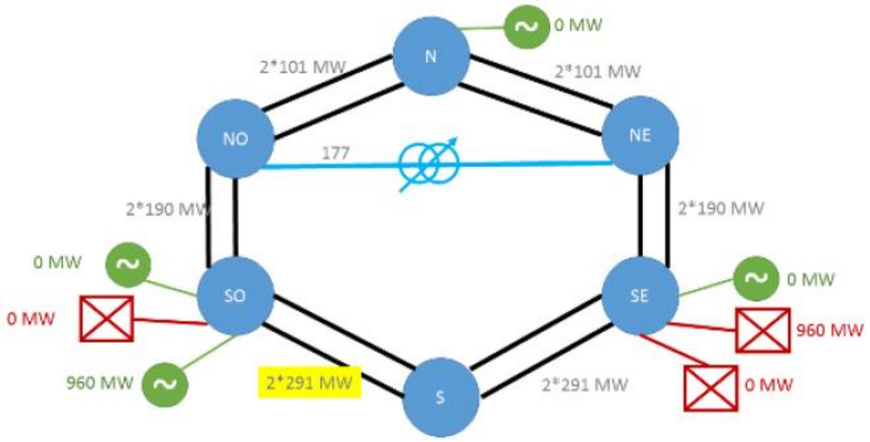
Load Flow calculation

- Security Analysis
  - "load-flow" mode
  - .csv results file

Ts	FLOW_S_SO_2	MAX_THREAT_1_FLOW_S_SO_1	MAX_THREAT_1_FLOW_S_SO_2	Threshold_N
T01	-290.5	-484.2	S_SO_1	400
T02	-290.5	-484.2	S_SO_1	400
T03	-290.5	-484.2	S_SO_1	100

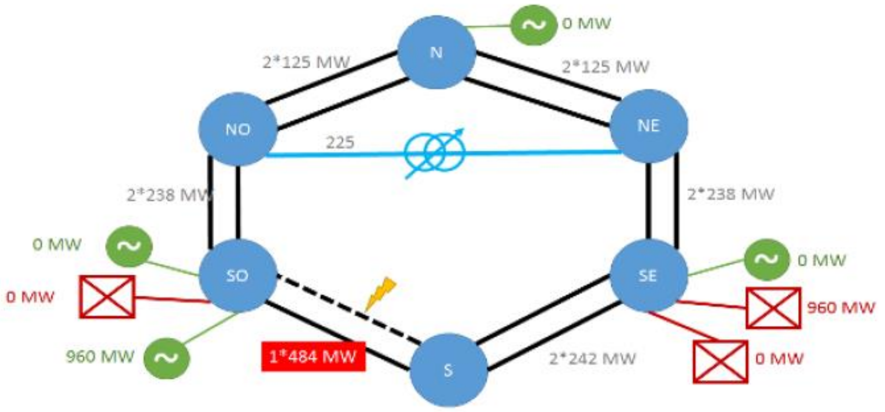
Calculated max threat is greater than mapped power limit because no optimization is performed in Security Analysis mode

Base case (N)



N-1 case

(S_SO_1 line contingency)



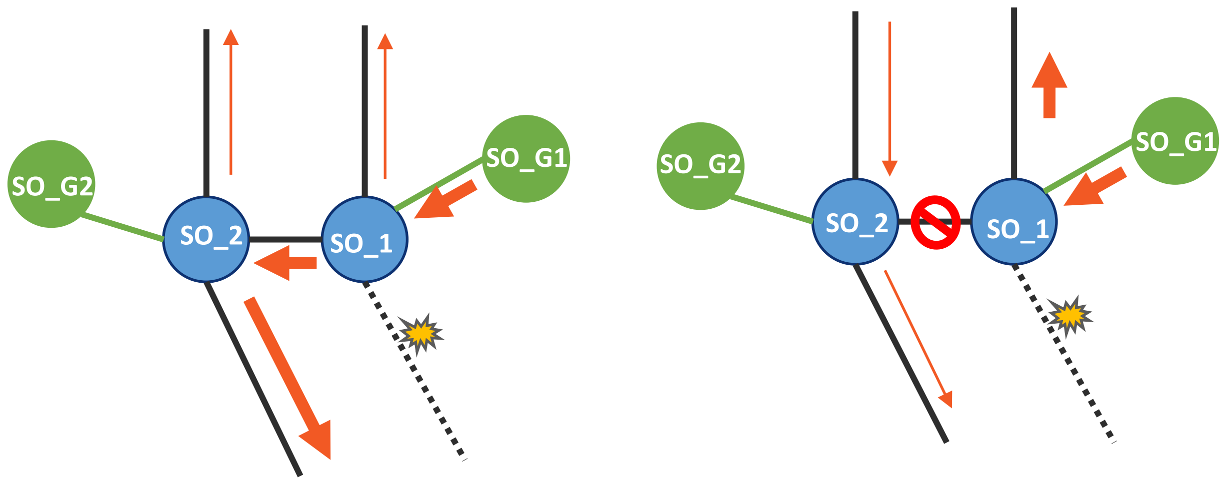
OPF without redispatching (1/3): curative remedial actions

- Topological curative actions list
  - .csv "parade" (remedial actions) file:

NB	4		
S_SO_1	1	SS1_SS1_DJ_OMN	
S_SO_1	1	SOO1_SOO1_DJ_OMN	
S_SO_1	2	SS1_SS1_DJ_OMN	SOO1_SOO1_DJ_OMN
S_SO_1	1	S_SO_2	

For the S_SO_1 contingency, 4 possible curative remedial action:

- Open bus coupler at S
  resulting in splitting S station into two nodes
- Open bus coupler at SO
  resulting in splitting SO station into two nodes
- Open both couplers
- Open S_SO_2 line

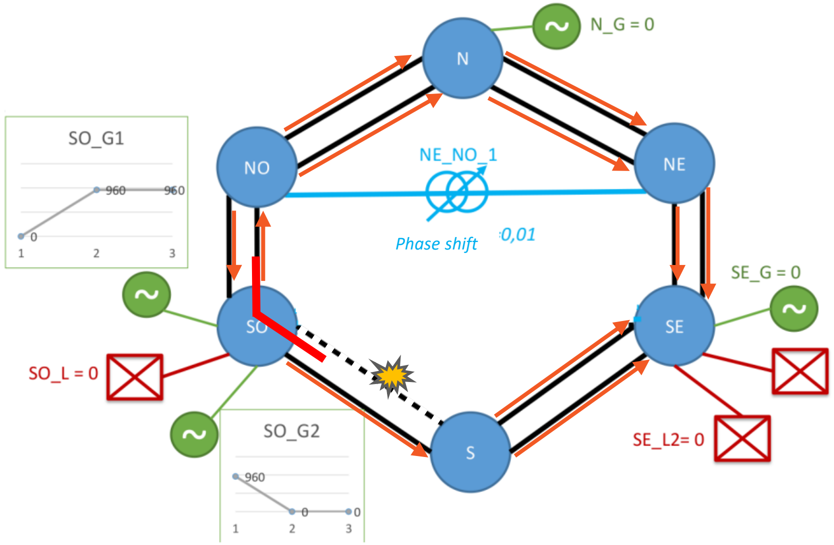


***Without** SOO1_SOO1_DJ_OMN* ***With** SOO1_SOO1_DJ_OMN*

OPF without redispatching (2/3)

Example

- Without the redispatching of generation and therefore only using non-costly topological parades allows to alleviate limit violations.
- Example: opening a coupler
  - In contingency S_SO_1, permanent limit is reached, as shown through SC Security Analysis.
  - At T01 and T02, Metrix opens bus coupler at SO (SOO1_SOO1_DJ_OMN) leading to two nodes: SO_1 and SO_2.
  - Therefore, instead of having a straightforward path from SO to S, flow has to do the following path: SO->NO->SO->S leading to increasing impedance. It limits the flow in SO_S_2 and increases flow going through the north (path SO->NO->N)



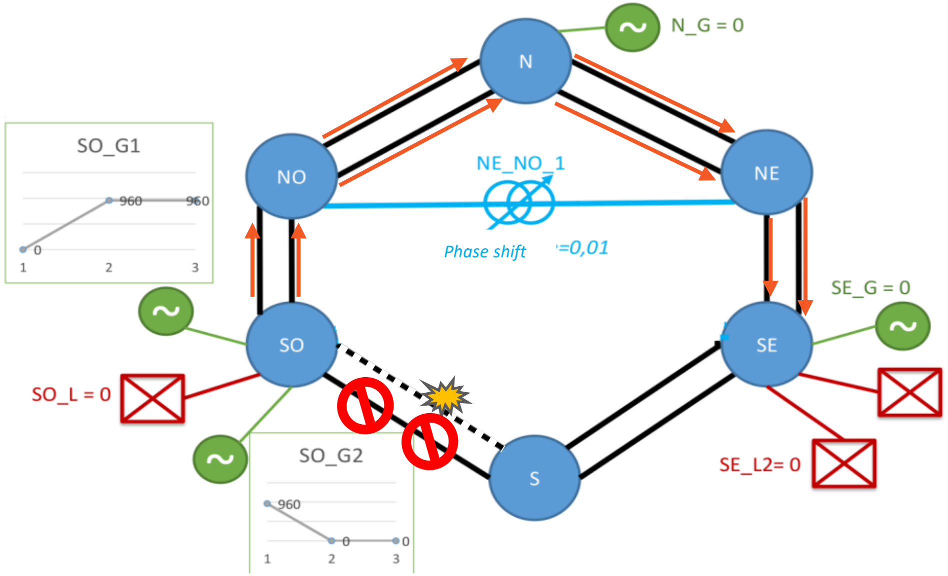
SOO1_SOO1_DJ_OMN flow (TS1, TS2)

OPF without redispatching (232)

Example

Example: opening a line

- If power flow threshold of the line S_SO_2 is mapped with “Threshold_N" (cf *Network construction and mapping* slide), flow limit is 100MW at T03: SOO1_SOO1_DJ_OMN is thus not enough anymore.
- Topological parade "open S_SO_2 line" is chosen to avoid limit violation.



S_SO_2 flow (TS3)

Optimal Power Flow

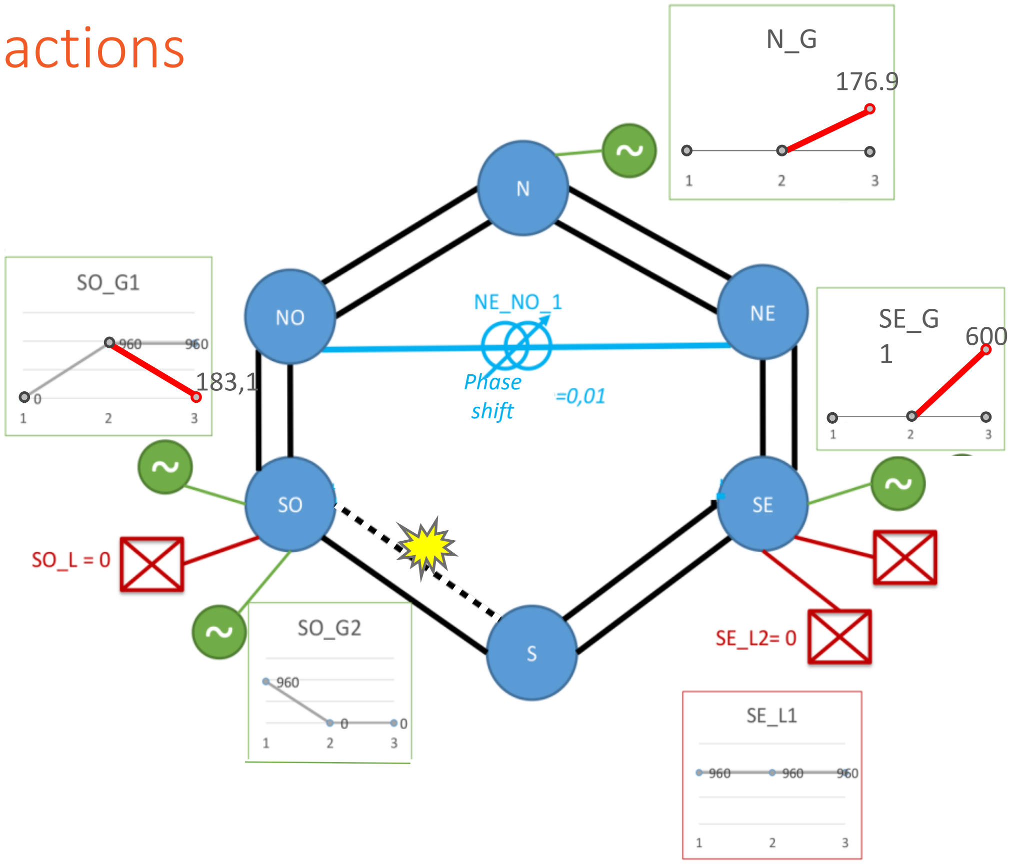
Preventive and curative remedial actions

If topological actions are disabled and redispatchir allowed

- If topological actions are disabled, curative redispatching will occur to lower the flow in S_SO_2. As a result, more expensive units are started (N_G, SE_G1) while already running units are stopped. *In the illustration, new generation set points are shown in red.*

If topological actions and curative redispatching allowed

- As topological curative actions are less expensive, Metrix prioritizes such solutions.
- In this case, topological solution exists (solution explained in [OPF without redispatching](#)).
- Therefore, no costly extra curative redispatching is made in this situation.



Topological actions disabled and curative redispatching allowed

Conclusion

To sum up

- PowSyBl-Metrix is a powerful SC DCOPF used to optimize preventive and curative remedial actions (Redispatching, topology, Phase-Shifter Transformers, HVDC)
- PowSyBl-Metrix is optimized to perform independent computations on each timestep of long timeseries (annual).

To get started with PowSybl-Metrix

- You will find all necessary infos for installation and practicals:
  - ↳ https://github.com/powsybl/powsybl-tutorials/tree/metrix