Optimal grid configuration a simple test case

Patrick Panciatici

November 2023

1 Introduction

Optimization of the grid configuration (topology) for the operation of the power grid to reduce of electricity costs while ensuring an acceptable level of reliability of the electricity supply.

In the context of the energy transition, the power grid has to cope with more and more uncertainties and volatility on the power injections and withdrawals. It must adapt in real time to rapidly changing conditions while ensuring an acceptable level of reliability. Operators supported by automatic digital controllers, are supervising and controlling the power system in real time. Their role is to keep the system within its security limits in real time, i.e. under current conditions (the base case), but also taking into account short-term future developments and plausible contingencies based on defined reliability criteria (typically, our historical "N-1" rule: the failure of a single system component should not result in a violation of the security limits under the most likely future conditions).

To maintain the system in its secured operation domain, the operator has some levers that can be activated preventively or after the occurrence of a disturbance (low probability contingency or large forecast errors). One consists in changing the power injections/withdrawals – which has an impact on the grid users (producers and consumers) and is therefore costly. One other consists in re-configuring the grid by acting on the switching devices which has the huge benefit of being costless.

To illustrate the problem, we will solve the following: How should substations be configured within an electrical grid to maximize power exchange between two areas?

We have created a simple toy example in DC approximation that focuses on substation reconfiguration.

This simple test case has been created so that the problem is very easy to understand for people outside the power grid community and doesn't require a lot of data.

The complexity of real use cases is much larger in many directions: size, line switching, AC formulation, security constraints, PST, HVDC in AC network, optimization over a time window with temporal constraints, stochastic formulation (chance constraints, ...), ...

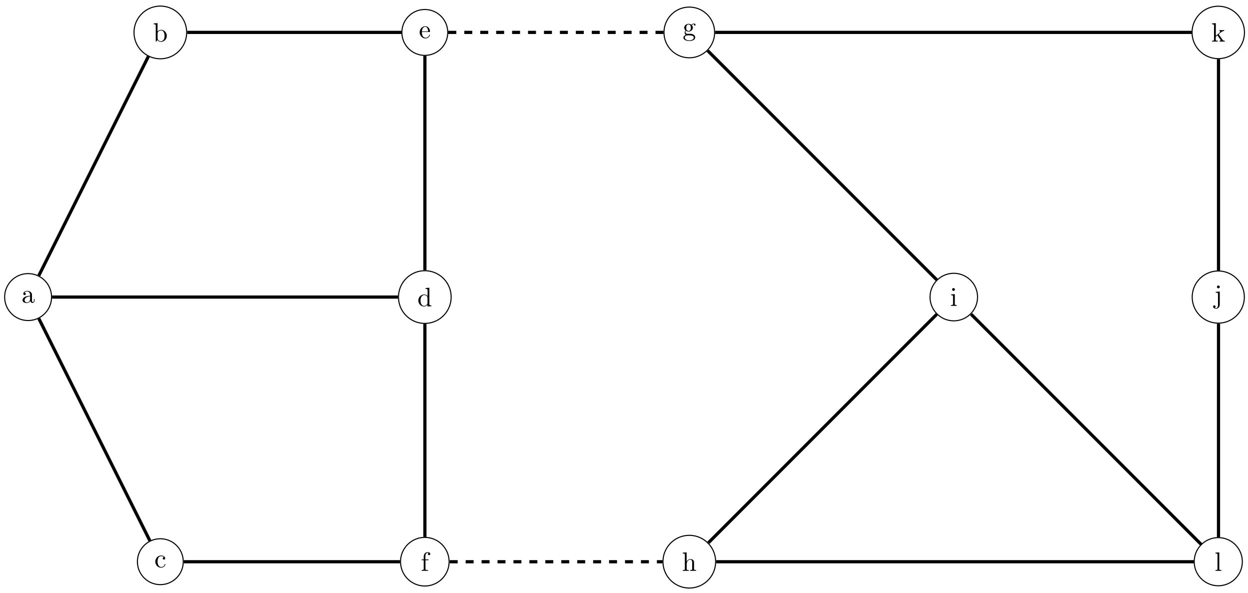
It's not totally representative of the real problem, but we think it captures some intrinsic characteristics that may be useful for analyzing complexity and selecting promising solutions.

We have done ampl implementations using a MILP to solve the problem. See attachments: Grid_MaxFlow_V(.run,.dat,.mod) without randomization and: Grid_MaxFlow_Vrandom and we provide some results in this document. These implementations fully illustrate the problem formulation as a MILP.

This simple test case could be useful for evaluating new, machine-learning based methods.

The substation designs used in this test case are not entirely realistic. They were chosen for their simplicity.

2 Simple example of grid configuration optimization

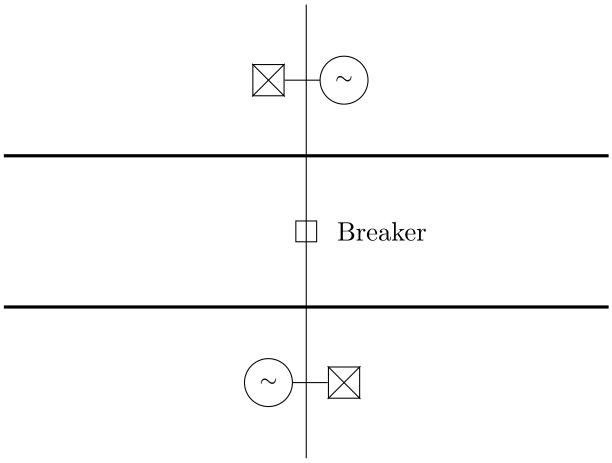


The dashed lines define the boundary between two zones. All circuits have double power lines and the nodes are substations with an internal configuration of busbars.

We want to estimate the maximum possible export from zone 1 in the left to zone 2 in the right. We increase the generation in each busbar proportionally to the initial generation in zone 1 and we increase the load in each busbar proportionally to the initial load in Zone 2. The power lines have maximum capacity.

We can select the substation configuration. We define in the following sections three simplified types of substations.

3 Substation with 2 neighbors with two powerlines each

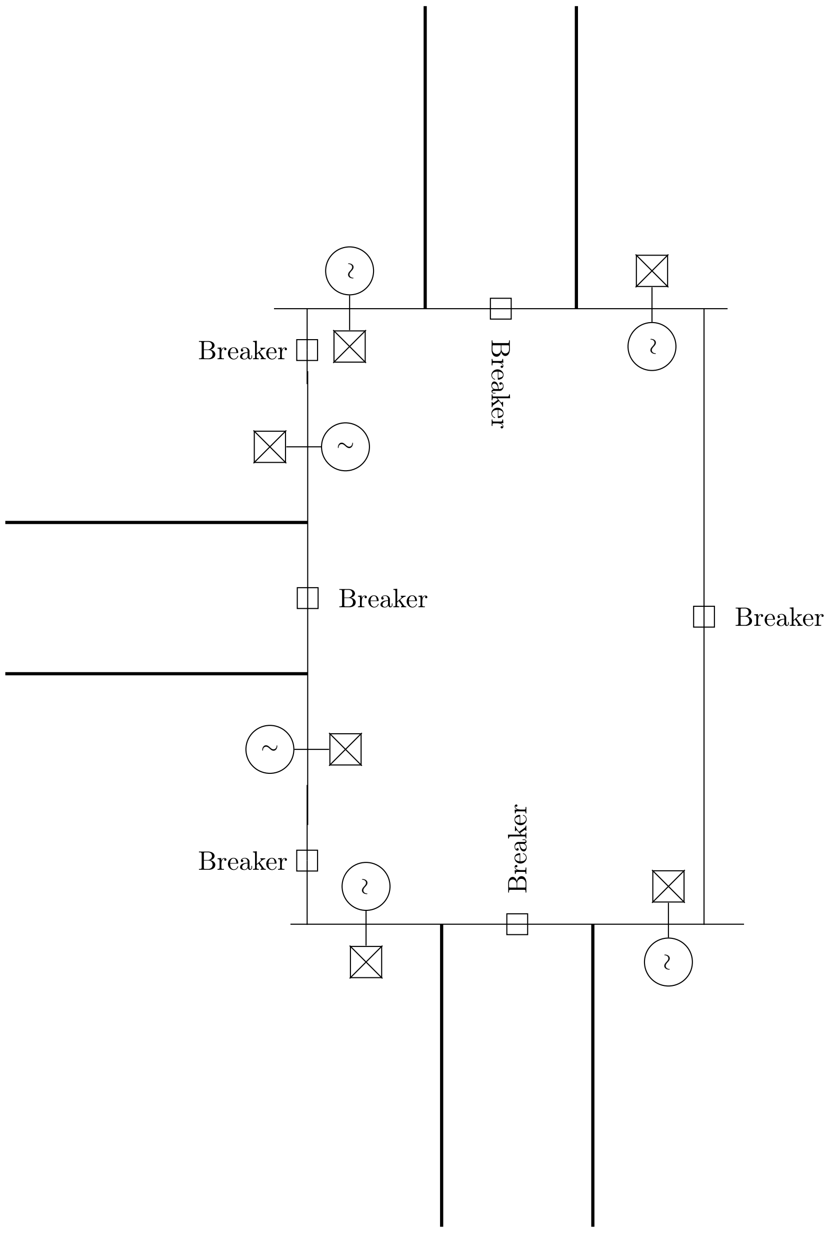


One breaker, 2 busbars
Case of nodes : b,c,j

4 Substation with 3 neighbors with two powerlines each

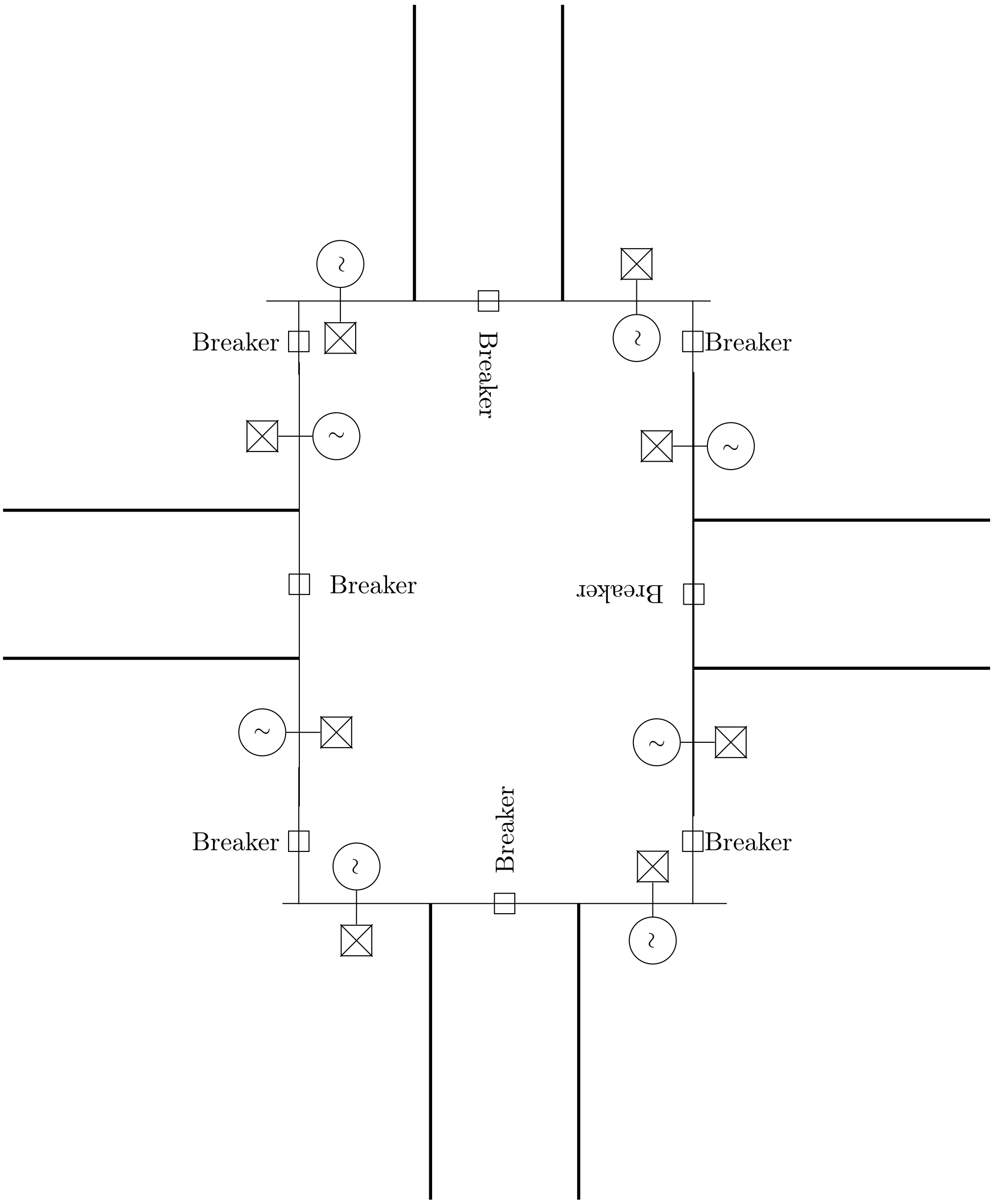
Six breakers, 6 busbars

Case of nodes : a,d,e,f,g,h,k,l



5 Substation with 4 neighbors with two powerlines each

Eight breakers, 8 busbars
Case of nodes : i



6 Degrees of freedom

i: 8 breakers
a,d,e,f,g,h,k,l : 8 x 6 = 48 breakers
b,c,j: 3 breakers, 6 busbars

Total: 59 binary variables and 62 continuous ones.

7 Problem Statement

The dashed lines define the boundary between the two zones.

We want to select circuit breaker states to maximize exports from the left zone to the right zone.

Maximization of export from zone 1 to zone 2: equivalent to maximization of power flows in lines (e-g) and (f-h): positive from left to right.

We can also solve the problem in the other direction (g-e) and (h-f), maximizing imports.

The initial loads and generations are given and the initial system state is balanced :

$$\sum_{i=1}^{N_G} G_i = \sum_{j=1}^{N_L} L_j$$

We increase the generation in zone 1 proportionally to λ and the load in zone 2 proportionally to μ while keeping the system balanced.

$$\begin{cases} \lambda.\sum_{i\in Zone_1} G_i + \sum_{i\in Zone_2} G_i = \sum_{j\in Zone_1} L_j + \mu.\sum_{j\in Zone_2} L_j \\ \mu = \alpha.\lambda + \beta \\ \alpha = \frac{\sum_{i\in Zone_1} G_i}{\sum_{j\in Zone_2} L_j} \\ \beta = \frac{(\sum_{i\in Zone_2} G_i - \sum_{j\in Zone_1} L_j)}{\sum_{j\in Zone_2} L_j} \end{cases}$$

All circuits have double lines. All reactances have the same value (1 p.u.), except for the interconnections (e-g) and (f-h) (dashed lines) for which it is double (2 p.u.).

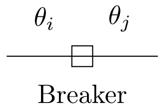
The states of the circuit breakers must be selected in order to keep a fully connected graph or at least balanced islands.

A DC approximation is used leading to a linear relationship between flows and injections (loads and generations) for a given topology (given set of breaker states).

8 Mathematical formulation

We create a variable θ_k by busbar. Each busbar is connected to another one through a breaker.

Breaker model:



If the breaker is closed then $\theta_i = \theta_j$ and the flow is a free variable, if it is open then θ_i and θ_j are free variables and the flow is zero.

Flow model for power line k connecting busbar i and j with the reactance x_k:

$$F_k = K.\frac{(\theta_i - \theta_j)}{x_k}$$

$$-F_k^{max} \leq F_k \leq F_k^{max}$$

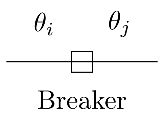
Balance equation for each busbar k:

$$\sum_{i \in \mathbb{L}_k} Flow_i^L + \sum_{i \in \mathbb{B}_k} Flow_i^B + Load_k - Generation_k = 0$$

Where $\mathbb{L}_k$ is the set of powerlines connected to busbar k, $\mathbb{B}_k$ is the set of breakers connected to busbar k, $Load_k$ is the load connected to busbar k and $Genaration_k$, the generation connected to busbar k.

9 Implementation as a MILP

For a convenient implementation, we introduce flows on breakers and binary states of breakers.



Using a BIG_M formulation, the breaker model becomes:

$$\begin{cases} Flow_{Breaker}(i,j) = -Flow_{Breaker}(j,i) \\ Flow_{Breaker}(i,j) \leq BIG_M.State_{Breaker}(i,j) \\ Flow_{Breaker}(j,i) \leq BIG_M.State_{Breaker}(i,j) \\ \theta_i - \theta_j \leq BIG_M.(1 - State_{Breaker}(i,j)) \\ \theta_j - \theta_i \leq BIG_M.(1 - State_{Breaker}(i,j)) \end{cases}$$

We develop a full implementation in ampl, see the attached documents:
Grid_MaxFlow_V.run, Grid_MaxFlow_V.dat, Grid_MaxFlow_V.mod.

10 Results

We present here some results defined in the test case Grid_MaxFlow_V(.run,.dat,.mod) without randomization. There is a unique set of nodal generations and loads (see in .dat file).

For identical F_{max} for all the power lines except for the interconnection lines without limits, we ran two test cases, one for $F_{max} = 400$ and the other $F_{max} = 800$.

For $F_{max} = 400$ the solution is:

Best integer solution found 1.210253082
8 integer solutions have been found
2055 branch and bound nodes

48 closed breakers

For $F_{max} = 800$ the solution is:

Best integer solution found 1.957819598
12 integer solutions have been found
25120 branch and bound nodes

45 closed breakers

Breaker	$Status_{800}$	$Status_{400}$
1 2	0	1
2 3	1	0
3 4	0	1
4 5	1	0
5 6	1	1
6 1	0	1
7 8	1	1
9 10	1	1
11 12	0	1
12 13	1	1
13 14	1	0
14 15	1	1
15 16	0	1
16 11	1	1
17 18	1	1
18 19	1	1
19 20	0	0
20 21	1	1
21 22	1	1
22 17	1	1
23 24	1	1
24 25	1	1
25 26	0	0
26 27	1	1
27 28	1	1
28 23	1	1
29 30	1	1
30 31	1	1
31 32	1	1
32 33	1	1
33 34	1	0
34 29	0	1
35 36	1	1
36 37	0	1
37 38	1	1
38 39	1	0
39 40	1	1
40 35	1	1
41 42	1	1
42 43	1	1
43 44	1	1
44 45	1	1
45 46	0	1
46 47	1	0
47 48	1	1
48 41	1	1
49 50	0	1
51 52	1	1
52 53	1	0
53 54	0	0
54 55	1	1
55 56	1	1
56 51	1	1
57 58	0	1
58 59	1	1
59 60	0	1
60 61	1	1
61 62	1	1
62 57	1	0

11 Other possible configurations to test generalization capabilities

When using a machine learning approach to solve this problem, it is important that the resulting agent has the ability to generalize to other unseen graphs during the learning phase.

In order to test whether some generalization possibilities exist, we propose here two modifications of the initial graph.

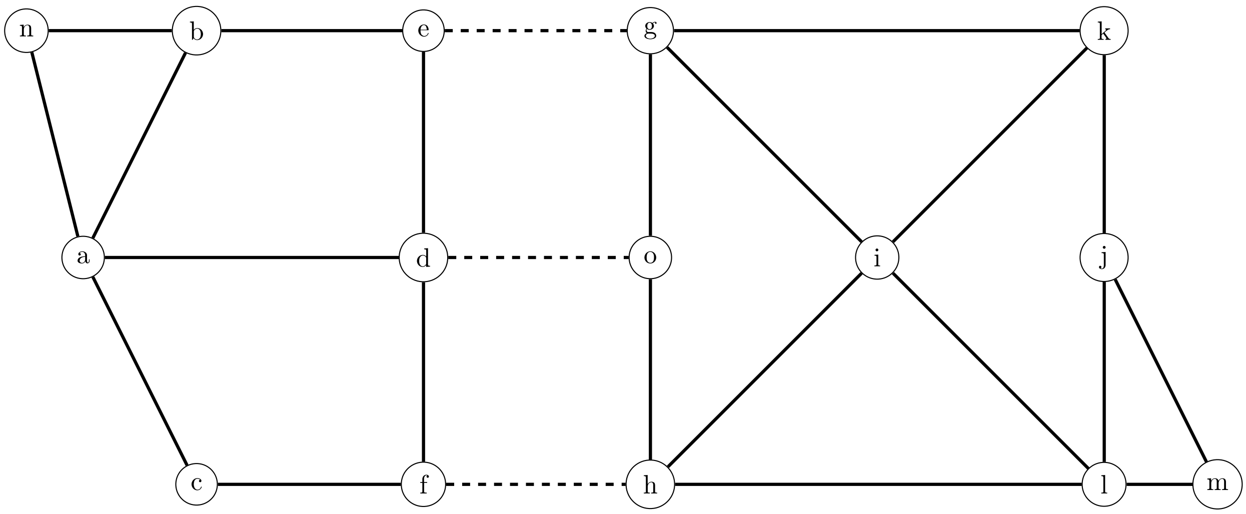
11.1 Modification 1

Addition of substations and associated connections.

substations : m,n,o
m connected to j and l.
n connected to a and b.
o connected to d, g and h (new interconnection)
Total of 15 substations
l becomes a substation with 4 neighbors
j becomes a substation with 3 neighbors
a becomes a substation with 4 neighbors
b becomes a substation with 3 neighbors
d becomes a substation with 4 neighbors
g becomes a substation with 4 neighbors
h becomes a substation with 4 neighbors

i,l,a,d,g,h: 6 x 8 = 48 breakers
e,f,k,b,o,j : 6 x 6 = 36 breakers
c,n,m: 3 breakers, 6 busbars

Total: 87 binary variables and 90 continuous ones.



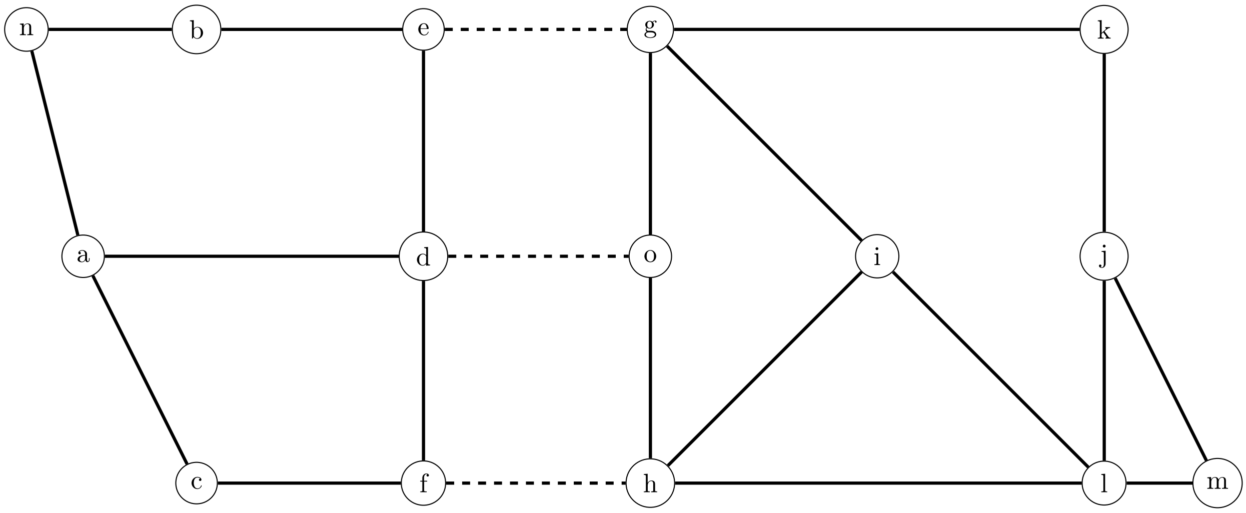
11.2 Modification 2

Removal of two connections after modification 1.

removal of connections a-b and k-i.
a becomes a substation with 3 neighbors
b becomes a substation with 2 neighbors
a becomes a substation with 3 neighbors
k becomes a substation with 2 neighbors
i becomes a substation with 3 neighbors

l,d,g,h: 4 x 8 = 24 breakers
a,e,f,i,o,j : 6 x 6 = 36 breakers
b,c,n,m,k: 5 breakers, 10 busbars

Total: 65 binary variables and 70 continuous ones.


n
b
e
g
k
a
d
o
i
j
c
f
h
l
m