Contents

0.1 Idea

For the OPF model construction it is convenient to model the network as directed graph $(\mathbf{B}, \mathbf{L})$ where $\mathbf{B}$ is the set of Buses and and $\mathbf{L} \subset \mathbf{B} \times \mathbf{B}$ is the set of branches of the network and for each adjacent buses k, m both (k, m) and (m, k) are in $\mathbf{L}$. So the line l adjacent to k, m is modeled by two edges in the arc $\{(k, m), (m, k)\}$. L can be partitioned in L_0 and L_1 with $|L_0| = |L_1|$ where every line l, adjacent to the buses k, m and with a transformer at k, is oriented so that $(k, m) \in L_0$ and $(m, k) \in L_1$. We also consider a set $\mathcal{G}$ of generators, partitioned into (possibly empty) subsets $\mathcal{G}_k$ for every bus $k \in \mathbf{B}$. We consider the following convex Jabr relaxation of the OPF problem:

$$\inf_{\substack{P_g^G, Q_g^G, c_{km}, \\ s_{km}, S_{km}, P_{km}, Q_{km}}} \sum_{g \in \mathcal{G}} F_g(P_g^G) \tag{1}$$

Subject to: $\forall km \in \mathbf{L}$

$$c_{km}^2 + s_{mk}^2 \leq c_{kk} c_{mm} \quad \text{Jabr constraint} \tag{2}$$

$$P_{km} = G_{kk} c_{kk} + G_{km} c_{km} + B_{km} s_{km} \tag{3}$$

$$Q_{km} = -B_{kk} c_{kk} - B_{km} c_{km} + G_{km} s_{km} \tag{4}$$

$$S_{km} = P_{km} + j Q_{km} \tag{5}$$

Power balance constraints: $\forall k \in \mathbf{B}$

$$\sum_{km \in L} S_{km} + P_k^L + i Q_k^L = \sum_{g \in \mathcal{G}(k)} P_g^G + i \sum_{g \in \mathcal{G}(k)} Q_g^G \tag{6}$$

Power flow, Voltage, and Power generation limits:

$$P_{km}^2 + Q_{km}^2 \leq U_{km} \tag{7}$$

$$V_k^{\min^2} \leq c_{kk} \leq V_k^{\max^2} \tag{8}$$

$$P_g^{\min} \leq P_g^G \leq P_g^{\max} \tag{9}$$

$$c_{kk} \geq 0 \tag{10}$$

$$c_{km} = c_{mk}, \; s_{km} = -s_{mk}. \tag{11}$$

This relaxation is in general not exact. We can recover exactness thanks to the following result:

Proposition 0.1.1. *Model* (1) *with the additional* loop constraint (12) *for every loop in a cycle basis of* $(\mathbf{B}, \mathbf{L})$ *is exact, we refer to this new model as the* Exact Jabr formulation

$$\sum_{k=0}^{\lfloor n/2 \rfloor} \sum_{\substack{A \subset [n] \\ |A| = 2k}} (-1)^k \prod_{h \in A} s_{k_h k_{h+1}} \prod_{h \in A^c} c_{k_h k_{h+1}} = \prod_{k=1}^{n} c_{k_i, k_i}. \tag{12}$$

This result suggests the following approaches to either find a feasible solution or move along the space of feasible solutions.

- Such relaxation is exact on tree Networks (also known as radial networks). Our objective is, given a network $\mathcal{N} = (\mathbf{B}, \mathbf{L})$ which can also not be a tree, consider a radial subnetwork $\mathcal{N}' = (\mathbf{B}, \mathbf{L}')$, with $\mathbf{L}' \subset \mathbf{L}$ and consider the Jabr model on $\mathcal{N}'$. This solution is not necessarily feasible for the original problem $\mathcal{N}$, our objective is to iteratively recover a feasible solution for $\mathcal{N}$. Since the Jabr relaxation is exact

  on $\mathcal{N}'$ it follows that the constraints 2 are respected, the constraints which are violated are the flow constraints on the leaves. We can try to recover feasibility my moving along the solution to the Jabr and Loop constraints.

- Given a feasible solution, find feasible directions.

0.2 Feasible directions

Let $x_0 = (P_0, Q_0, c_0, s_0)$ be a feasible solution of the OPF problem (1) with the loop constraints (12). We want to find feasible directions $x_0 = (dP, dQ, dc, ds)$, that is such that $x_1 = (P_0 + dP, Q_0 + dQ, c_0 + dc)$ is still a feasible solution of the OPF problem. We consider each constraint of the Exact Jabr Formulation separately do get feasible directions.

0.2.1 Jabr Constraint

Since x_0, the jabr equality holds: $c_{ii}c_{jj} = c_{ij}^2 + s_{ij}^2$. Adding the movement dx we want that $(c_{ii} + dc_{ii})(c_j j + dc_{jj}) = (c_{ij} + dc_{ij})^2 + (s_{ij} + ds_{ij})^2$. By expanding the terms and considering that the equality holds for x_0 this is equivalent to:

$$dc_{ii}dc_{jj} + 2c_{ii}dc_{jj} + 2dc_{ii}c_{jj} = dc_{ij}^2 + c_{ij}dc_{ij} + ds_{ij}^2 + s_{ij}ds_{ij} \tag{13}$$

For now we consider movements where dc_{ii} is not zero only on an independent set of nodes in the graph, this way the constraint simplifies to:

$$dc_{ij}^2 + c_{ij}dc_{ij} + ds_{ij}^2 + s_{ij}ds_{ij} - 2dc_{ii}c_{jj} = 0 \tag{14}$$

The solutions to this constraint can be found by minimizing the following minimization problem:

$$\min(dc_{ij}^2 + c_{ij}dc_{ij} + ds_{ij}^2 + s_{ij}ds_{ij} - 2dc_{ii}c_{jj})^2 \tag{15}$$

Which can be solved by gradient descent methods. Since we want to salve this for all $(i, j) \in \mathbf{L}$, we instead solve the following:

$$\min \sum_{(i,j)\in\mathbf{L}} (dc_{ij}^2 + c_{ij}dc_{ij} + ds_{ij}^2 + s_{ij}ds_{ij} - 2dc_{ii}c_{jj})^2 \tag{16}$$

G: Anche questo può essere risolto con metodo gradiente? Sappiamo che il minimo globale è zero, ma possono esserci minimi locali? Dobbiamo per forze imporre gli spostamenti solo su nodi indipendenti?