1 Random ideas

2 Jabr Model

For the OPF model construction we 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 $\mathbf{L}_0$ 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 $(m, k) \in \mathbf{L}_1$. 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:

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

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 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.

3 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.

3.1 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 solve this for all $(i, j) \in \mathbf{L}$, we instead solve the following:

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

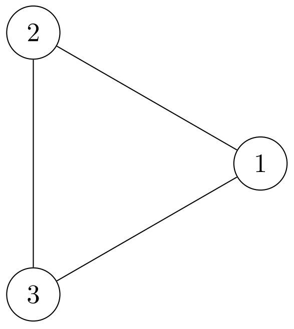
G: Anche questo può essere risolto con metodo gradiente? Sappiamo che il minimo globale è zero, ma possono esserci minimi locali? Dobbiamo per forze imporre gli spostamenti solo su nodi indipendenti?

3.2 Loop Constraints

Unfortunately, the number of monomials to computer for checking the various of loop constraints grows factorially with the length of the cycle. Thus trying to apply a similar method to find feasible directions satisfying the loop constraints as we did for the Jabr constraints would be computationally intractable. We can try the following approaches:

- Find the cycle basis with minimal weight
- Each cycle can be broken down by adding auxiliary edges in the Network, decomposing each loop constraints in loop contraints associated to cycle of the desired length. For each auxiliary edge $e = (e_0, e_1)$ we add the auxiliary variable c_e, s_e, the associated Jabr constraints: $c_e^2 + s_e^2 = c_{e_0} c_{e_1}$. We can now divide the cycles containing e_0 and e_1 in two cycles containing the edge e and thus obtaining smaller associated loop contraints.
- Loop constraints of cycles of length equal to 3 or 4 can be rewritten as 2 polynomial constraints of degree 2.

Let us consider the loop constraints on the following cycle:



This corresponds to the following polynomial constraints:

$$p_3 \coloneqq c_{12}(c_{23}c_{31} - s_{23}s_{31}) - s_{12}(s_{23}c_{31} + c_{23}s_{31}) - c_{11}c_{22}c_{33} \tag{17}$$

We also define the following polynomials associated to the Jabr constraints at each edge:

$$p_{ij} \coloneqq c_{ij}^2 + s_{ij}^2 - c_{ii}c_{jj} \tag{18}$$

and

$$q_3^1 = s_{12}c_{33} + c_{23}s_{31} + s_{23}c_{31} \tag{19}$$

$$d_3^2 = c_{12}c_{33} + c_{23}c_{31} + s_{23}s_{31} \tag{20}$$

Then, we have the following results which let's us to rewrite the constraints associated to a cycle of length three as two polynomial constraints of degree 2.

Proposition 2

$$\{(c,s) \mid p_3 = p_{12} = p_{23} = p_{31} = 0\} = \{(c,s) \mid q_3^1 = q_3^2 = p_{12} = p_{23} = p_{31} = 0\} \tag{21}$$

If we don't restrict the possible direction $dx = (d_c, d_s, d_P)$, imposing that constraints (19), (20) hold for $x + dx$ would impose polynomial constraints on dx. Alternatively, since for example q_3^1 is a bilinear polynomial respect to the vectors $v_1 = (s_{12}, c_{23}, s_{23})$ and $v_2 = (c_{33}, s_{31}, c_{31})$, we can fix the variables in v_2, that is $dv_2 = 0$ and move along v_1. This way constraint $q_3^1 = 0$ imposes a linear constraint on d_x. We can proceed on a similar way for q_3^2. We note that one can consider instead to move along v_2 and that there are other possible choices for v_1 and v_2.

G: The hope is that even though we are restricting the directions for a single step, that by concatenating steps we can obtain a "good" amount of feasible directions.

G: TODO: think well about possible directions

3.3 Flow constraints

The flow constraint (6) also induce linear constraints on the step dx:

$$dP_{km} = G_{kk} dc_{kk} + G_{km} dc_{km} + B_{km} ds_{km} \tag{22}$$

$$dQ_{km} = -B_{kk} dc_{kk} - B_{km} dc_{km} + G_{km} ds_{km} \tag{23}$$

$$dS_{km} = dP_{km} + j dQ_{km} \tag{24}$$

$$\sum_{km \in L} dS_{km} + dP_k^L + i dQ_k^L = \sum_{g \in \mathcal{G}(k)} dP_g^G + i \sum_{g \in \mathcal{G}(k)} dQ_g^G \tag{25}$$

3.4 Other constraints

Magnitude constraints are what limit the step size:

$$V_k^{\min^2} \le c_{kk} + dc_{kk} \le V_k^{\max^2} \tag{26}$$

$$P_g^{\min} \le P_g^G + dP_g^G \le P_g^{\max} \tag{27}$$

$$c_{kk} + dc_{kk} \ge 0 \tag{28}$$

G: what about this one?

$$P_{km}^2 + Q_{km}^2 \le U_{km}$$

3.5 Step Problem

Combining the considerations in the previous subsections we obtaing the following problem to calculate a feasible step. Where given a graph, we find a cycle basis, we decompose each cycle in a 3/4-cycles, for each of these smaller cycles we consider their loop constraints and pick which variables do move in the step dx.

G: how?

G: Da riscrivere meglio, i vincoli su q_e^1, q_e^2 sono per ciascun arco in un ciclo in una base di cicli, ma se lo stesso arco è in due basi di cicli diverse, se i due cicli non hanno nodi in comune allora sono due vincoli diversi

$$\min CJ(dx) = sum_{(i,j)\in\mathbf{L}}(dc_{ij}^2 + c_{ij}dc_{ij} + ds_{ij}^2 + s_{ij}ds_{ij} - 2dc_{ii}c_{jj})^2 \tag{29}$$

Subject to:

$$q_e^1 = ds_{12}c_{33} + dc_{23}s_{31} + ds_{23}c_{31} \tag{30}$$
$$q_e^2 = dc_{12}c_{33} + dc_{23}c_{31} + ds_{23}s_{31} \tag{31}$$
$$dP_{km} = G_{kk}dc_{kk} + G_{km}dc_{km} + B_{km}ds_{km} \tag{32}$$
$$dQ_{km} = -B_{kk}dc_{kk} - B_{km}dc_{km} + G_{km}ds_{km} \tag{33}$$
$$dS_{km} = dP_{km} + jdQ_{km} \tag{34}$$
$$\sum_{km\in L} dS_{km} = \sum_{g\in\mathcal{G}(k)} dP_g^G + i \sum_{g\in\mathcal{G}(k)} dQ_g^G \tag{35}$$
$$V_k^{\min^2} \leq c_{kk} + dc_{kk} \leq V_k^{\max^2} \tag{36}$$
$$P_g^{\min} \leq P_g^G + dP_g^G \leq P_g^{\max} \tag{37}$$
$$c_{kk} + dc_{kk} \geq 0 \tag{38}$$

If the optimal solutions of problem (29) are feasible steps if and only if their cost is 0. Leaven like this one obvious solution is $dx = 0$. Possible ideas.

- Impose that the generation cost must decrease, by putting the scalar product of dx with the gradient of the cost function F in x to be smaller than 0. $dx\nabla F(x) < 0$. With this additional constraint if it's an equality then the family of the possible direction doesn't have a decreasing cost direction.

Since the constraints are affine, problem (29) can be solved with a projected gradient descent algorithm.

G: If we don't consider magnitude constraints, we can consider directly the projection of the gradient on the subspace of feasible solutions, this would be easier than considering the projection on a generic