

Morenet documentation

Tillmann Mühlfordt

May 4, 2020

Abstract

This documents sketches the idea to solve the distributed power flow problem as a distributed nonlinear least-squares problem.

1 Problem formulation

The mathematical formulation for the decentralized power flow problem reads

$$g_i^{\text{pf}}(x_i, z_i) = 0, \quad (1a)$$

$$g_i^{\text{bus}}(x_i) = 0, \quad (1b)$$

$$\sum_{i=1}^{n^{\text{reg}}} A_i \begin{bmatrix} x_i \\ z_i \end{bmatrix} = 0, \quad (1c)$$

where the consensus matrices $A_i \in \mathbb{R}^{4n^{\text{conn}} \times (4n_i^{\text{core}} + 2n_i^{\text{copy}})}$ enforce equality of the voltage angle and the voltage magnitude at the copy buses and their respective original buses. Mathematically speaking, Problem 1 is a system of nonlinear equations; there are as many equations as there are unknowns. In principle, Problem 1 can be solved by Newton's method. However, we would like to explore alternatives to Newton's method, such as distributed optimization.

2 Problem solution

Currently, we solve Problem 1 as a *distributed feasibility problem*, namely

$$\min_{x_i, z_i \forall i \in \{1, \dots, n^{\text{reg}}\}} 0 \quad \text{s. t.} \quad (2a)$$

$$g_i^{\text{pf}}(x_i, z_i) = 0, \quad (2b)$$

$$g_i^{\text{bus}}(x_i) = 0, \quad (2c)$$

$$\sum_{i=1}^{n^{\text{reg}}} A_i \begin{bmatrix} x_i \\ z_i \end{bmatrix} = 0. \quad (2d)$$

From our experience so far, we can say that Aladin is a viable method, but ADMM is not. Solving a distributed feasibility problem, however, is not the only way. We can also think of solving Problem 1 as a

distributed least-squares problem of the form

$$\min_{x_i, z_i \forall i \in \{1, \dots, n^{\text{reg}}\}} \sum_{i=1}^{n^{\text{reg}}} \left\| \begin{bmatrix} g_i^{\text{pf}}(x_i, z_i) \\ g_i^{\text{bus}}(x_i) \end{bmatrix} \right\|^2 \quad \text{s. t.} \quad \sum_{i=1}^{n^{\text{reg}}} A_i \begin{bmatrix} x_i \\ z_i \end{bmatrix} = 0. \quad (3a)$$

Clearly, the solution to Problem 2 is the solution to Problem 3 (why?). Hence, we can think of solving the least-squares Problem 3 as a necessary condition.

2.1 Next steps

The goal is to explore how to solve the distributed power flow problem via a distributed least-squares Problem 3. To do so, there are a couple of immediate next steps

- Familiarize with nonlinear least-squares (Gauss-Newton, Levenberg-Marquardt, etc)
- Solve a prototypical distributed power flow problem.
- Use the Jacobian matrix that is given analytically as a return value from [generate_distributed_problem](#).
- Work out how the Aladin algorithm simplifies in the absence of both equality and inequality constraints. (Note that inequality constraints in the form of lower/upper bounds might still be necessary, i.e. $\underline{x} \leq x \leq \overline{x}$)
- Exploit the problem structure as much as possible. As a general rule, the bigger the problem, the more it pays off to exploit sparsity etc.

Another thing to explore is to interface Ipopt directly from Matlab. This is somewhat painful to [set up](#) but might be worth the speed up anyhow.

[Figure 1](#) shows all possible *paths* that are worth taking. Insights thus far are

- The path “feasibility problem (2) > ADMM” is a dead end; this is a valuable insight in itself.
- The path “feasibility problem (2) > Aladin” works with
 - Casadi & Ipopt for small problems (< 300 buses total) in a reasonable time; Casadi is the main bottleneck, Ipopt is not.
 - fmincon for medium-sized problems (< 1000 buses total); fmincon becomes simply too slow.
- The path “least-squares problem (3) > ADMM” has not been explored yet.
- The path “least-squares problem (3) > Aladin” is being investigated, and promising.

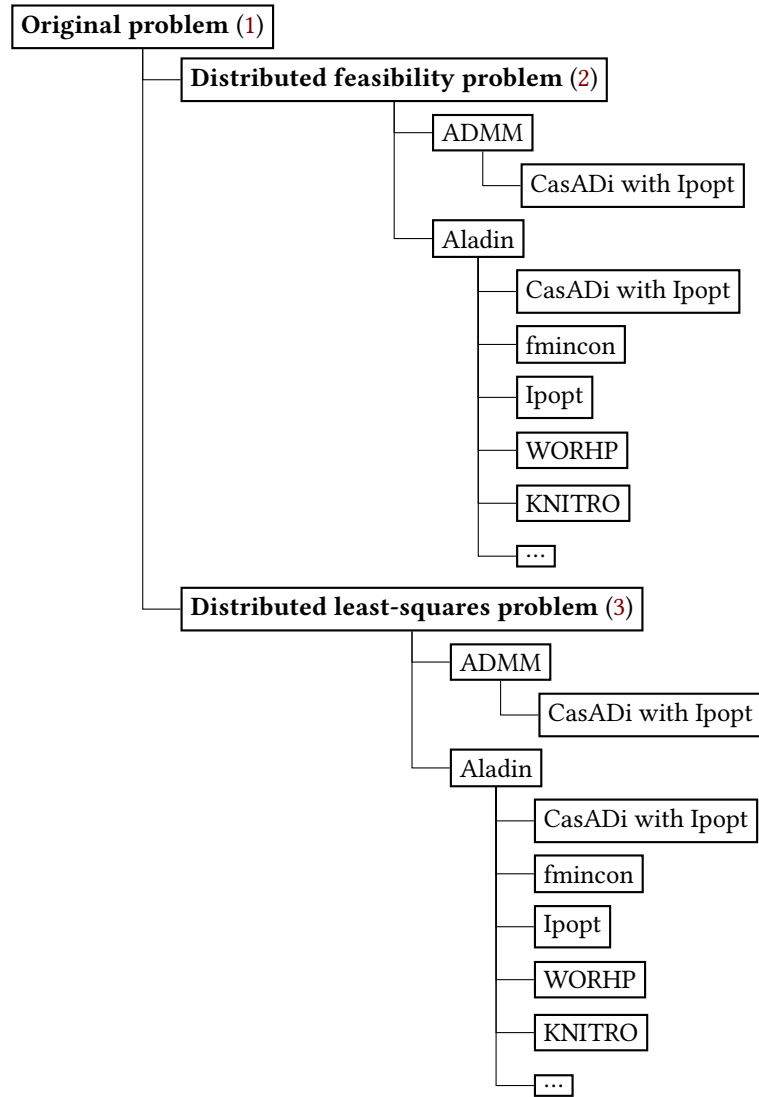


Figure 1: Variants of tackling the distributed power flow problem (1).