

Lab Assignment 3: DCOPF

Instructor: Parikshit Pareek

1. Introduction to DCOPF

In your previous course (EET109), you learned how to solve the **DC Power Flow (DCPF)** problem. The DCPF is a tool for simulation; it allows us to calculate power flows and voltage angles given a fixed set of generation and demand. In **Direct Current Optimal Power Flow (DCOPF)**, we move from simulation to optimization.

2. DCOPF: A Narrative Description

Read the following description carefully. Your first task in this assignment is to “extract the math” from this narrative and translate it into the standard Linear Programming (LP) form.

Imagine you are the system operator for a regional power grid (Check online how many such regional grids are in India). You are overseeing a network of several buses connected by high-voltage transmission lines. At various points in this grid, you have generators, each of which charges a specific price for every MW it produces. Your overarching goal is to run the entire system at the absolute minimum total cost to the consumers.

However, you are bound by the laws of physics and the physical limits of your equipment. First, at every single bus in the system, you must ensure a perfect *power balance*: the power produced by local generators, plus the net power flowing in from neighboring buses through the transmission lines, must exactly equal the power consumed by the local loads. Second, your transmission lines cannot carry infinite current; they have strict thermal limits. If the physics of the network tries to push more power through a line than its rated capacity, the line will overheat (and they will melt right!). Additionally, each generator has its own technical constraints— it cannot be turned down below a certain minimum stable level, nor can it be pushed beyond its maximum nameplate capacity. Finally, to make the math consistent across the network, you must pick one bus to serve as your “North Star,” (fancy way of saying reference is needed as angles are always relative) fixing its voltage angle to zero so all other angles have a reference point.

Student Task: Based on the narrative above, identify:

1. The **Decision Variables** (What are you choosing?).
2. The **Objective Function** (What are you minimizing?).
3. The **Equality Constraints** (Which part describes the “perfect balance” and “laws of physics”?).
4. The **Inequality Constraints** (Which part describes “thermal limits” and “technical constraints”?).
5. Get the form: $\min_{\mathbf{x}} \mathbf{c}^\top \mathbf{x}$ subject to $\mathbf{A}\mathbf{x} \leq \mathbf{b}$; $\mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq}$

3. Task 1: Mathematical Formulation and Duality

3.1 Standard Form Construction

Rewrite the DCOFP problem in the **Standard LP Form** shown above.

- Explicitly define the components of the vector \mathbf{x} .
- Construct the matrix \mathbf{A}_{eq} and vector \mathbf{b}_{eq} representing the nodal power balance (using the B -matrix notation).
- Construct the matrix \mathbf{A} and vector \mathbf{b} representing both the generation limits and the line flow limits.

3.2 The Dual Problem

In optimization, every primal problem has a **Dual**.¹

1. Formulate the Dual of your DCOFP standard form.
2. Assign the dual variable λ_i to the nodal power balance constraint at bus i .
3. **Analytical Question:** In the context of electricity markets, what is the economic significance of the dual variable λ_i ? How does it relate to Locational Marginal Pricing (LMP)? [Don't worry; you can read stuff online.]

4. Task 2: Implementation and Coding

In this task, you will solve the DCOFP for a benchmark system. You are required to use a generic optimization modeling language (Python/Pyomo or Julia/JuMP) and an LP solver (e.g., HiGHS, GLPK, or Gurobi).

4.1 Data and the B-Matrix

You must load a network file from the **PGLIB-OPF** dataset.² Use `case5_pjm.m`.

Reminder: You already know how to construct the B -matrix from your EET109 assignments. Use those same logic blocks to build your constraints. Specifically:

- **Nodal Balance:** $\mathbf{P}_g - \mathbf{P}_d = \mathbf{B}\boldsymbol{\theta}$ [Make sure to derive this by hand]
- **Line Flows:** $\mathbf{F}_{ij} = \frac{1}{x_{ij}}(\theta_i - \theta_j)$ [Make sure to derive this by hand]

4.2 Coding Requirements

Do **not** use specialized OPF solvers (like `pandapower.rundcopf` or `PowerModels.solve_dcopf`). You must code the constraints manually to show you understand the underlying LP.³

¹For a rigorous treatment of LP duality in power systems, see: Wood, A. J., Wollenberg, B. F., and Sheblé, G. B. (2013). *Power Generation, Operation, and Control*. John Wiley & Sons.

²The Power Grid Lib - Optimal Power Flow (pglib-opf) benchmarks are available at: <https://github.com/power-grid-lib/pglib-opf>

³Refer to the Pyomo documentation for constraint modeling: Hart, W. E., et al. *Pyomo—Optimization Modeling in Python*. Springer.

```
1 # Just An Example Script Idea #
2 # -----
3 import pyomo.environ as pyo
4 import numpy as np
5
6 model = pyo.ConcreteModel()
7
8 # 1. Define sets (Buses, Generators, Lines)
9 # 2. Define Variables (model.Pg, model.theta)
10 # 3. Define Objective: Minimize Total Cost
11 # 4. Define Constraints (Code these individually):
12 #     - Nodal Power Balance (using your B-matrix from EET109)
13 #     - Line Flow Limits (F_min <= B_line * theta <= F_max)
14 #     - Generation Limits (Pg_min <= Pg <= Pg_max)
15 #     - Reference Bus Angle (theta[ref] = 0)
16
17 # 5. Solve and extract duals (model.dual = pyo.Suffix(direction=pyo.Suffix.IMPORT)
    )
```

Listing 1: Suggested structure for Python/Pyomo

5. What you must have at the end!

1. Primal and Dual mathematical formulations of DCOPF
2. Your own commented Code (.py or .jl).
3. A table of results: Generation Dispatch (P_g) at each bus and the resulting LMPs (λ) at each bus.