Step 2: Unit Commitment

Project Step 2 focuses on the Unit Commitment (UC) and Economic Dispatch (ED) problem over a 24-hour time horizon for the IEEE 24-bus system, taking into account the generation limits, load demands, transmission constraints, and reserve requirements across the network. This project is an essential power system optimization model while maintaining system reliability and minimizing operational costs.

Project Objectives

1. **Minimize Generation Costs:** Formulate and solve a cost-minimizing model that determines the optimal generation schedule for the 24-hour period.
2. **Ensure System Balance:** Match generation with the hourly system demand profile across the different buses.
3. **Manage Reserve Requirements:** Include reserve constraints to ensure system reliability.
4. **Account for Ramping Constraints:** Implement ramping up and down constraints.
5. **Incorporate Transmission Constraints:** Respect the limits of the transmission lines.

Mathematical Model Components

1. **Objective Function:** Minimize the total generation cost, including start-up, shutdown, and operational costs of each generating unit.
2. **Decision Variables:** Define binary and continuous variables for unit commitment status, generation levels, and reserve allocations.
3. **Constraints:**
    - **Power Balance:** Match total generation with demand for each hour across the network.
    - **Generator Limits:** Respect maximum and minimum generation limits for each unit.
    - **Reserve Constraints:** Ensure sufficient reserves are allocated as per system requirements. Use the same reserve constraints we used in class (Lecture 8, slide 21). Neglect columns R+, R-, C+,C-.

1. **Reserve requirement coverage:**
    - The reserve must be at least equal to the largest production of a single generator:

python Copy code

```python
m.addConstrs(r_req[t] >= p[i,t] for i in range(n_gen))
```

2. **Total reserve provision:**
    - The total reserve (spinning + non-spinning) from all generators must cover the reserve requirement:

python Copy code

```python
m.addConstr(sum(p_rs[i,t] + p_rns[i,t] for i in range(n_gen)) >= r
```

3. **Spinning reserve requirement:**
    - At least 50% of the reserve must come from spinning reserves:

python Copy code

```python
m.addConstr(sum(p_rs[i,t] for i in range(n_gen)) >= 0.5*r_req[t])
```

- **Ramping Constraints:** Impose ramping limits.
- **Transmission Constraints:** Ensure flow on each transmission line remains within capacity.

Mention how many decision variables and constraints your model has. Explain what each constraint fulfills.

Data and Inputs

- **Generating Units:** Data on maximum/minimum power output, reserve capacities, ramping limits, start-up/shutdown costs, and operational status.
- **Load Profile:** Hourly system demand across the 24-hour period and distribution of loads across different nodes.
- **Transmission Lines:** Reactance and capacity data for the transmission lines to account for grid limitations. **The system's slack bus is node 13**.
- **RES:** 24-hour renewable energy output.

- **Ideal Storage:** Add a 500 MWh ideal storage to Node 15. Assume it is filled half full at the start of the day, and it should be half full again for the next day. Assume the P_Max for the storage is 250 MW.