More on Security Constrained Optimal Power Flow

1.0 Notation

In the last class, we represented the OPF and the SCOPF as below.

\[ \min f(P) \leftarrow \text{OBJECTIVE} \Rightarrow \min f(P) \]

**Subject to:**

\[ g(P) = 0 \leftarrow \text{Power Flow Eqs} \Rightarrow \quad g(P) = 0 \]
\[ h_{\min} \leq h(P) \leq h_{\max} \leftarrow \text{"Normal Condition" constraints} \Rightarrow \quad h_{\min} \leq h(P) \leq h_{\max} \]
\[ \text{Contingency constraints} \Rightarrow \quad h'_{\min} \leq h'(P) \leq h'_{\max} \]

We will change notation now. Instead of using the notation \( h' \) (h-prime) to indicate the constraints under contingencies, we will subscript the constraints, where the subscript indicates the contingency state. For example, the optimal power flow (OPF) problem can be written as below. We will call this problem \( P_0 \).

\[ \text{Min } f_0(x_0, u_0) \]
\[ \text{s.t. } g_k(x_k, u_0) = 0 \quad k = 0 \]
\[ h_k(x_k, u_0) \leq h_k^{\max} \quad k = 0 \]

Here, \( g_k(x_k, u_0) = 0 \) represents the power flow equations and \( h_k(x_k, u_0) \leq h_k^{\max} \) represents the line-flow constraints. The state variables \( x_0 \) denote the bus voltage magnitudes and angles under pre-contingency conditions. The index \( k=0 \) indicates this problem is posed for only the pre-contingency condition,” i.e., the condition with no contingencies. Thus, this problem is just the OPF.
Now let’s consider the security-constrained OPF (SCOPF). Its problem statement is given as problem $P_p$:

$$\min \ f_0(x_0, u_0)$$

$$s.t. \ g_k(x_k, u_0) = 0 \quad k = 0, 1, 2, ..., c$$

$$h_k(x_k, u_0) \leq h_k^{\max} \quad k = 0, 1, 2, ..., c$$

Notice that there are $c$ contingencies to be addressed in the SCOPF, and that there are a complete new set of constraints for each of these $c$ contingencies. Each set of contingency-related equality constraints is exactly like the original set of equality constraints (those for problem $P_0$), except it corresponds to the system with an element removed. Each set of contingency-related inequality constraints is exactly like the original set of inequality constraints (those for problem $P_0$), except it corresponds to the system with an element removed and, for line flow constraints and for voltage magnitudes, the limits will be different.

Also notice that the constraints are a function of $x_k$, the voltage magnitudes and angles under the pre-contingency ($k=0$) and contingency conditions ($k>1, 2, ..., c$), and $u_0$, the controls which were set under the pre-contingency conditions ($k=0$).

### 2.0 Reducing computation time for SCOPF

Denote the number of constraints for the OPF, Problem $P_0$, as $N$.

Assumption: Let’s assume that running time $T$ of the algorithm we use to solve the above problem is proportional to the square of the number of constraints, i.e., $N^2$. For simplicity, we assume the constant of proportionality is 1, so that $T = N^2$.

So the SCOPF must deal with the original $N$ constraints, and also another set of $N$ constraints for every contingency. Therefore, the total number of constraints for Problem $P_p$ is $N + cN = (c+1)N$. 

2
Under our assumption that running time is proportional to the square of the number of constraints, then the running time will be proportional to \([c+1]^2\)(c+1)^2N^2=(c+1)^2T.

What does this mean?
It means that the running time of the SCOPF is \((c+1)^2\) times the running time of the OPF. So if it takes OPF 1 minute to run, and you want to run SCOPF with 100 contingencies, it will take you \(101^2\) minutes, or 10,201 minutes to run the SCOPF. This is 170 hours, about 1 week!!!!

Many systems need to address 1000 contingencies. This would take about 2 years!

So this is what you do…..

![Diagram](image_url)

Fig. 1: Decomposition solution strategy

The solution strategy first solves the OPF (master problem) and then takes contingency 1 and re-solves the OPF, then contingency 2 and resolves the OPF, and so on (these are the subproblems). For any contingency-OPFs which require a redispatch, relative to the \(k=0\) OPF, an appropriate constraint is generated, at the end of the cycle, these constraints are gathered and applied to the \(k=0\) OPF. Then the \(k=0\) OPF is resolved, and the cycle starts again. Experience has it that such an approach usually requires only 2-3 cycles.
Denote the number of cycles as \( m \).

Each of the individual problems has only \( N \) constraints and therefore requires only \( T \) minutes.

There are \((c+1)\) individual problems for every cycle.

There are \( m \) cycles.

So the amount of running time is \( m(c+1)T \).

If \( c=100 \) and \( m=3 \), \( T=1 \) minute, this approach requires 303 minutes. That would be about 5 hours (instead of 1 week).

If \( c=1000 \) and \( m=3 \), \( T=1 \) minute, this approach requires about 50 hours (instead of 2 years).

In addition, this approach is easily parallelizable, i.e., each individual OPF problem can be sent to its own CPU. This will save even more time.

Figure 2 [1] compares computing time for a 6-bus system (Fig. 2a) and a 24 bus test system (Fig. 2b). The comparison is between a full SCOPF, a decomposed SCOPF (DSCOPF), and a decomposed SCOPF where individual OPF problems are sent to separate CPUs. This kind of algorithm is formalized as Benders decomposition.
There is a rich literature on using decomposition methods for solving SCOPF and SCUC. Searching on *Benders* and *(optimal power flow or unit commitment)* returns 54 hits in IEEE Xplore.

### 3.0 Preventive vs. Corrective

In this section, we desire to distinguish between two kinds of security-related actions, i.e., two kinds of control. We recall the security-state diagram of Fig. 3 considered before.
Preventive control is an action taken to move from the alert state to the normal state. Preventive control is taken to prevent an undesirable operating condition from occurring if a contingency occurs. Since there is no immediate consequence of such a state, preventive control is not typically concerned with how much time a particular action requires.

Corrective control is an action taken to move from the emergency state to the alert state or from the emergency state to the normal state. Since an emergency state is experiencing an existing undesirable operating condition, it is important to move out of the emergency state quickly. As a result, corrective control is heavily concerned with how much time a particular action will take.

4.0 Preventive SCOPF
The preventive SCOPF is the one we have already posed as problem $P_p$, repeated below for convenience.

$$\begin{align*}
\text{Min} & \quad f_0(x_0, u_0) \\
\text{s.t.} & \quad g_k(x_k, u_0) = 0 \quad k = 0, 1, 2, \ldots, c \\
& \quad h_k(x_k, u_0) \leq h_k^{\text{max}} \quad k = 0, 1, 2, \ldots, c
\end{align*}$$

As already mentioned in Section 1.0, the constraints are a function of $x_k$, the voltage magnitudes and angles under the pre-contingency ($k=0$) and contingency ($k>1, 2, \ldots, c$) conditions, and $u_0$, the controls which were set under the pre-contingency conditions ($k=0$).

The fact that the controls were restricted to their pre-contingency condition settings, thus denoted $u_0$, makes this a preventive SCOPF.
5.0 Fully corrective SCOPF

The fully corrective SCOPF is posed below as problem $P_{c1}$.

\[
\begin{align*}
\text{Min} & \quad f_0(x_0, u_0) \\
\text{s.t.} & \quad g_0(x_0, u_0) = 0 \\
& \quad h_0(x_0, u_0) \leq h_0^{\text{max}} \\
& \quad g_k(x_k, u_k) = 0 \quad k = 1, 2, \ldots, c \\
& \quad h_k(x_k, u_k) \leq h_k^{\text{max}} \quad k = 1, 2, \ldots, c
\end{align*}
\]

This problem is considered “corrective” because post-contingency $(k=1, 2, \ldots, c)$ controls $u_k$ are allowed to move in order to satisfy the post-contingency constraints. The problem is considered “fully” corrective because we allow post-contingency constraints to be satisfied independent of pre-contingency conditions, i.e., $g_k$ and $h_k$, $k>0$, do not include $u_0$ as an argument. The effect of this is that we are not allowing preventive control in this problem since we do not allow movement of pre-contingency controls $u_0$ in order to satisfy post-contingency constraints.

6.0 Preventive-corrective SCOPF

The preventive-corrective SCOPF is posed below as problem $P_{c2}$.

\[
\begin{align*}
\text{Min} & \quad f_0(x_0, u_0) \\
\text{s.t.} & \quad g_0(x_0, u_0) = 0 \\
& \quad h_0(x_0, u_0) \leq h_0^{\text{max}} \\
& \quad g_k(x_k, u_k) = 0 \quad k = 1, 2, \ldots, c \\
& \quad h_k(x_k, u_k) \leq h_k^{\text{max}} \quad k = 1, 2, \ldots, c \\
& \quad |u_k - u_0| \leq \Delta u_k^{\text{max}} \quad k = 1, 2, \ldots, c
\end{align*}
\]
Here, the amount of corrective control that can be expended is limited by an amount $\Delta u_k^{\text{max}}$ and the pre-contingency control setting $u_0$ through the last constraint. The following observations should be made:

1. The right-hand side of the last constraint, $\Delta u_k^{\text{max}}$, is the maximum change for the post-contingency control variables. It is computed as a product of
   - the assumed time horizon allowed for corrective actions $T_k$
   - and an assumed rate (typically maximum) of change of control variables in response to contingency, $\frac{du_k}{dt}$, i.e.,

   $$\Delta u_k^{\text{max}} = T_k \left. \frac{du_k}{dt} \right|_{\text{max}} k = 1, 2, \ldots, c$$

2. The post-contingency control levels $u_k$ do not appear in the objective function, i.e., the only values that affect the objective function are $u_0$.

3. If there are no violated post-contingency constraints, then $u_0$ will be selected based only on the objective function and the pre-contingency constraints.

4. If there are violated post-contingency constraints, the algorithm will try to satisfy them using only post-contingency control levels $u_k$, because this does not affect the objective function. This is using the “corrective control” part of the algorithm.

5. If the violated post-contingency constraints cannot be satisfied using only post-contingency control levels $u_k$, then the algorithm will use pre-contingency control levels $u_0$ to satisfy them. This is using the “preventive control” part of the algorithm.

6. It is important to realize that the reason we use corrective control first, and preventive control only if necessary, is that
   - The corrective control is perceived not to cost very much if the contingency occurs, because the “contingency state” is not expected to last very long. In addition, the
contingency likely will not occur, in which case the corrective control will cost nothing at all!

- In contrast to the previous bullet, any change to pre-contingency control variables $u_0$, a preventive control, moves the system away from the optimal economic point independent of whether a contingency occurs or not, and therefore, this change will always cost money!

7. Because the post-contingency control levels $u_k$ are not included in the objective function, it is possible to find different corrective controls that will provide feasibility for the same objective function value. Thus, we see that the preventive-corrective SCOPF can have multiple solutions. To distinguish between the various solutions, one could add post-contingency control costs to the objective function, but since the contingencies might or might not happen, one would have to condition those post-contingency control costs for each contingency on the probability of that contingency.

7.0 **Risk-based preventive-corrective OPF**
We saw in the slides presented in the last class that the risk-based OPF, relative to the SCOPF, removed the post-contingency inequality constraints and replaced them with a risk constraint.

<table>
<thead>
<tr>
<th>Model 1: (SCOPF)</th>
<th>Model 2: (RBOPF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\min f(P)$</td>
<td>$\min f(P)$</td>
</tr>
<tr>
<td><strong>Subject to:</strong></td>
<td><strong>Subject to:</strong></td>
</tr>
<tr>
<td>$g(P) = 0$</td>
<td>$g(P) = 0$</td>
</tr>
<tr>
<td>$h_{\min} \leq h(P) \leq h_{\max}$</td>
<td>$h_{\min} \leq h(P) \leq h_{\max}$</td>
</tr>
<tr>
<td>$h'<em>{\min} \leq h'(P) \leq h'</em>{\max}$</td>
<td>$\text{Risk}(P) \leq \text{RMAX}$</td>
</tr>
</tbody>
</table>
We can apply the same idea to our preventive-corrective SCOPF, as indicated below, denoted as Problem $P_{c2}$.

$$\begin{align*}
\text{Min} & \quad f_0(x_0, u_0) \\
\text{s.t.} & \quad g_j(x_0, u_0) = 0 \\
& \quad h_j(x_0, u_0) \leq h_0^{\text{max}} \\
& \quad g_k(x_k, u_k) = 0 \quad k = 1, 2, \ldots, C \\
& \quad |u_k - u_0| \leq \Delta u_k^{\text{max}} \quad k = 1, 2, \ldots, C
\end{align*}$$

Problem $P_{c2}$ would indeed be an interesting problem to explore further. One should start this process by reviewing the literature, at least the following ones: [1, 2, 3, 4], together with the citations in these four publications.

### 8.0 Risk-based LMPs

When exploring the LP-OPF, we have seen how the OPF relates to LMPs as the Lagrange multipliers on the power flow equations. I derive the traditional form of the LMPs in a set of notes that I have loaded on the web. I will not go through that here. Reference [5] derives them under the RB-OPF. The solutions are given below.

**Deterministic:**

$$DLM_{P_k} = \pi \left(1 - \frac{\partial L}{\partial P_k} + \mu_s \frac{\partial F_{eq}}{\partial P_k} + \mu_t \frac{\partial F_{eq}}{\partial P_k}\right)$$

**Energy component**

**Loss component**

**Congest component for one contingency, (line s, t above 100%, line u at 92%)**

**Risk-based:**

$$R_{LM_{P_k}} = \pi \left(1 - \frac{\partial L}{\partial P_k} + \phi \cdot p_q \left(\frac{\partial S_{eq}}{\partial P_k} + \frac{\partial S\text{ev}(P_{eq})}{\partial P_k} + \frac{\partial S\text{ev}(F_{eq})}{\partial P_k}\right)\right)$$

10
The risk-based LMP, denoted RLMP above, may be more attractive than the deterministic LMPs, denoted DLMP above, in three main ways:

- Control of risk level is uniform
- Price signal for risk-relief is more effective
- LMPs are less volatile

This is another interesting area to explore.