Security Constrained Optimal Power Flow

1.0 Introduction and notation

Figure 1 below compares the optimal power flow (OPF) with the security-constrained optimal power flow (SCOPF).

OPF		SCOPF
$\min f(\underline{P})$	← OBJECTIVE →	$\min f(\underline{P})$
Subject to:		**Subject to:**
$\underline{g}(\underline{P}) = \underline{0}$	← Power Flow Eqts →	$\underline{g}(\underline{P}) = \underline{0}$
$\underline{h}_{\min} \le \underline{h}(\underline{P}) \le \underline{h}_{\max}$	← "Normal Condition" constraints →	$\underline{h}_{\min} \le \underline{h}(\underline{P}) \le \underline{h}_{\max}$
	Contingency constraints →	$\underline{h}'_{\min} \le \underline{h}'(\underline{P}) \le \underline{h}'_{\max}$

Fig. 1

Some comments about these different formulations:

- SCOPF solution will always have a cost $\geq$ OPF solution.
- If we ignore losses, then we can say that an OPF solution differs from an EDC solution (economic dispatch calculation, i.e., no transmission) only when a **normal** transmission constraint becomes binding.
  - Occurs when normal flow moves from just < 100% to $\geq$100% of *continuous* rating.
- SCOPF differs from an OPF solution only when a **contingency** transmission constraint becomes binding.
  - Occurs when post-contingency flow moves from just < 100% to $\geq$100% of *emergency* rating.

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.

$$
P_0 \qquad \begin{aligned} & Min && f(\underline{x}_0, \underline{u}_0) && \\ & s.t. && \underline{g}_k(\underline{x}_k, \underline{u}_0) = \underline{0} && k = 0 \\ & && \underline{h}_k(\underline{x}_k, \underline{u}_0) \le \underline{h}_k^{\max} && k = 0 \end{aligned}
$$

Here,

- $\underline{g}_k(\underline{x}_k,\underline{u}_0)=0$ represents the power flow equations;
- $\underline{h}_k(\underline{x}_k,\underline{u}_0)\le\underline{h}_k^{\max}$ represents the branch-flow constraints;
- $\underline{h}_k^{\max}$ represent branch continuous ratings.

The state variables $\underline{x}_0$ 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:

$$
P_p \qquad \begin{aligned} & Min && f(\underline{x}_0, \underline{u}_0) && \\ & s.t. && \underline{g}_k(\underline{x}_k, \underline{u}_0) = \underline{0} && k = 0,1,2,\ldots,c \\ & && \underline{h}_k(\underline{x}_k, \underline{u}_0) \le \underline{h}_k^{\max} && k = 0,1,2,\ldots,c \end{aligned}
$$

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. Observe:

- 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 branch flow constraints and for voltage magnitudes, the limits will be different.

Also notice that the constraints are a function of $\underline{x}_k$, the voltage magnitudes and angles under the pre-contingency (k=0) and contingency conditions (k>1,2,…,c), and $\underline{u}_0$, 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.

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

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…..

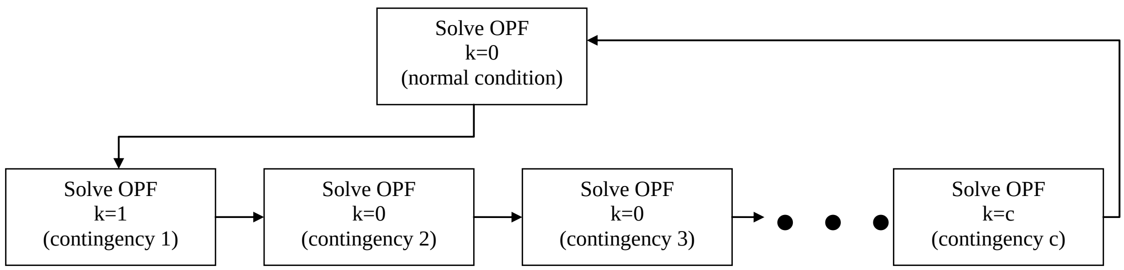


Fig. 1: Decomposition solution strategy

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.

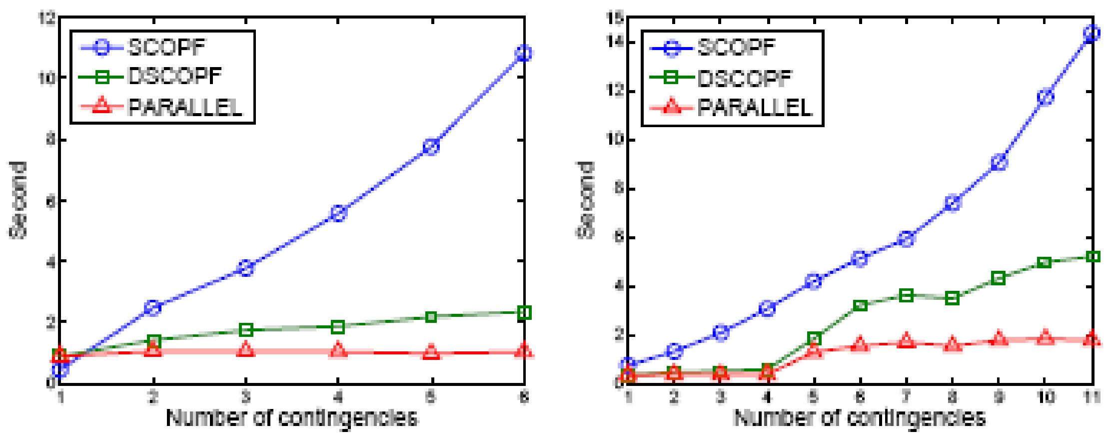
So the amount of running time is m(c+1)T.

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.



(a) Test system A6 (b) Test system B6

Fig. 2

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. Consider the security-state diagram of Fig. 3.

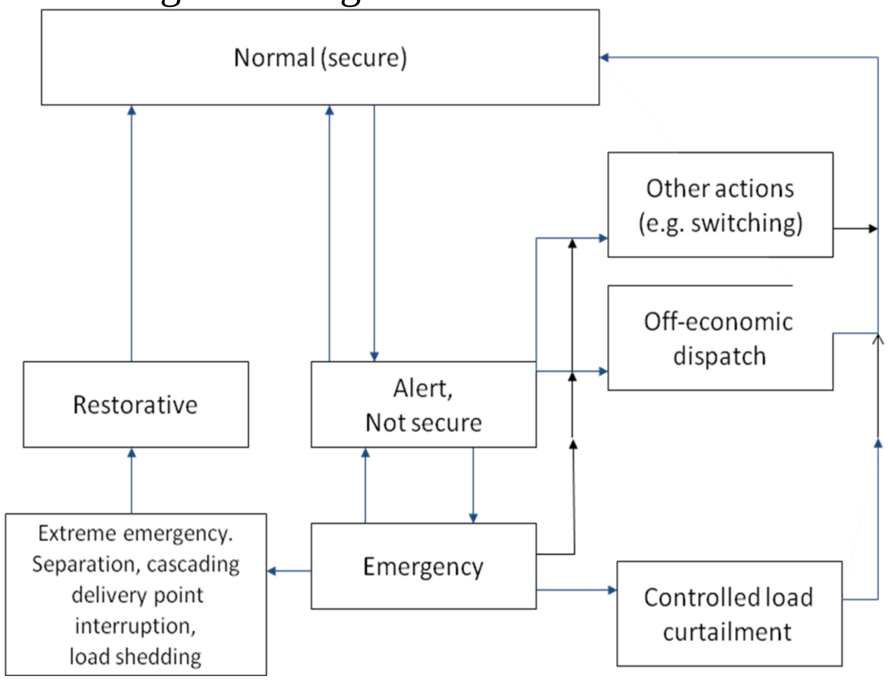


Fig. 3

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.

$$
P_p \qquad
\begin{array}{lll}
Min & f(\underline{x}_0, \underline{u}_0) & \\
s.t. & \underline{g}_k(\underline{x}_k, \underline{u}_0) = \underline{0} & k = 0,1,2,\ldots,c \\
 & \underline{h}_k(\underline{x}_k, \underline{u}_0) \leq \underline{h}_k^{\max} & k = 0,1,2,\ldots,c
\end{array}
$$

As already mentioned in Section 1.0, the constraints are a function of $\underline{x}_k$, the voltage magnitudes and angles under the pre-contingency (k=0) and contingency (k>1,2,…,c) conditions, and $\underline{u}_0$, the controls which were set under the pre-contingency conditions (k=0).

But what makes this a *preventive* SCOPF?

The fact that
 the controls are restricted to their
 pre-contingency condition settings,
 thus denoted $\underline{u}_0$,
makes this a *preventive* SCOPF.

That is, we must position the power system while in the normal condition (i.e., no contingency) to *prevent* operating conditions following occurrence of a contingency to exceed emergency ratings.

Note that here and elsewhere in these notes, "contingency" refers to any contingency on a specified contingency list but not a contingency that is not on the specified contingency list. That is, we *do not* account for *all* possible contingencies but rather just a certain set, generally all N-1 contingencies.

5.0 Fully corrective SCOPF

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

$$
P_{c1} \quad
\begin{aligned}
Min \quad & f(\underline{x}_0, \underline{u}_0) \\
s.t. \quad & \underline{g}_0(\underline{x}_0, \underline{u}_0) = \underline{0} \\
& \underline{h}_0(\underline{x}_0, \underline{u}_0) \le \underline{h}_0^{\max} \\
& \underline{g}_k(\underline{x}_k, \underline{u}_k) = \underline{0} \qquad k = 1,2,\ldots,c \\
& \underline{h}_k(\underline{x}_k, \underline{u}_k) \le \underline{h}_k^{\max} \qquad k = 1,2,\ldots,c \\
& \left| \underline{u}_k - \underline{u}_0 \right| \le \underline{K} \qquad k = 1,2,\ldots,c
\end{aligned}
$$

Here, $\underline{K}$ is a vector of very large positive numbers. Observations:

- This problem is considered "corrective" because post-contingency (k=1,2,…c) controls $\underline{u}_k$ 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., $\underline{g}_k$ and $\underline{h}_k$, k>0, include $\underline{u}_k$ as an argument instead of $\underline{u}_0$ (compare to prob P_p). Two implications:
    - Post-contingency controls **<u>may</u>** move as much as needed, within bounds of the control capabilities (usually the max & min gen values) to satisfy post-contingency constraints. This is made possible by setting the elements of $\underline{K}$ to +∞, i.e., the difference between post-contingency control levels $\underline{u}_k$ and pre-contingency control levels $\underline{u}_0$ are unrestricted.
    - Post-contingency controls **<u>must</u>** move as much as needed, within bounds of the control capabilities (usually the max & min gen values) to satisfy post-contingency constraints. In other words, we do not allow preventive control in this problem, i.e., we do not allow setting pre-contingency controls $\underline{u}_0$ to satisfy post-contingency constraints. So post-contingency constraints must be satisfied entirely by post-contingency control.

Because the post-contingency conditions are independent of pre-contingency variables, Problem P_{c1} is really Problem P_0 (the OPF), *unless*…one of the contingency problems k>1 is infeasible. In this case, the entire problem is considered infeasible because there is a contingency which leads to an infeasible condition and cannot be made feasible no matter what post-contingency control we take.

Question: What is the order of these problems, P_0, P_p, P_{c1}, in terms of decreasing production cost?

Answer: $\text{Cost}(P_p) \geq \text{Cost}(P_{c1}) = \text{Cost}(P_0)$.

Question: What is the order of these problems, P_0, P_p, P_{c1}, in terms of increasing system risk? Here, we must imagine that we have an acceptable measure of system risk.

Answer: $\text{Risk}(P_p) \leq \text{Risk}(P_{c1}) < \text{Risk}(P_0)$.

Observe: Production costs decrease as risk increases.

6.0 Preventive-corrective SCOPF

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

$$P_{c2} \qquad \begin{aligned} &Min && f(\underline{x}_0, \underline{u}_0) \\ &s.t. && \underline{g}_0(\underline{x}_0, \underline{u}_0) = \underline{0} \\ &&& \underline{h}_0(\underline{x}_0, \underline{u}_0) \leq \underline{h}_0^{\max} \\ &&& \underline{g}_k(\underline{x}_k, \underline{u}_k) = \underline{0} && k = 1,2,\ldots,c \\ &&& \underline{h}_k(\underline{x}_k, \underline{u}_k) \leq \underline{h}_k^{\max} && k = 1,2,\ldots,c \\ &&& \left|\underline{u}_k - \underline{u}_0\right| \leq \Delta\underline{u}_k^{\max} && k = 1,2,\ldots,c \end{aligned}$$

Here, the amount of corrective control that can be expended is limited by an amount $\Delta\underline{u}_k^{\max}$ and the pre-contingency control setting $\underline{u}_0$ through the last constraint. The following observations are made:

1. The right-hand side of the last constraint, $\Delta\underline{u}_k^{max}$, 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, $d\underline{u}_k/dt$, i.e.,

$$\Delta\underline{u}_k^{\max} = T_k \left.\frac{d\underline{u}_k}{dt}\right|_{\max} \qquad k = 1,2,\ldots,c$$

2. The post-contingency control levels $\underline{u}_k$ do not appear in the objective function, i.e., the only values that affect the objective function are $\underline{u}_0$.
3. If there are no violated post-contingency constraints, then $\underline{u}_0$ 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 $\underline{u}_k$, 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 $\underline{u}_k$, then the algorithm uses pre-contingency control levels $\underline{u}_0$ 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 $\underline{u}_0$, 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 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 can add post-contingency control costs to the objective function, but since the contingencies might or might not happen, one must condition those post-contingency control costs for each contingency on the contingency probability. This problem is provided below as Problem P_{c3}.

$$P_{c3} \quad \begin{aligned} Min \quad & p_0 f(\underline{x}_0, \underline{u}_0) + p_k \sum_{k=1}^{c} f(\underline{x}_k, \underline{u}_k) \\ s.t. \quad & \underline{g}_0(\underline{x}_0, \underline{u}_0) = \underline{0} \\ & \underline{h}_0(\underline{x}_0, \underline{u}_0) \le \underline{h}_0^{\max} \\ & \underline{g}_k(\underline{x}_k, \underline{u}_k) = \underline{0} \qquad k = 1,2,\ldots,c \\ & \underline{h}_k(\underline{x}_k, \underline{u}_k) \le \underline{h}_k^{\max} \qquad k = 1,2,\ldots,c \\ & \left|\underline{u}_k - \underline{u}_0\right| \le \Delta\underline{u}_k^{\max} \qquad k = 1,2,\ldots,c \end{aligned}$$

---

[1] Y. Li, “Decision making under uncertainty in power system using Benders decomposition.” PhD Dissertation, Iowa State University, 2008.