

# Contingency Filtering Techniques for Preventive Security-Constrained Optimal Power Flow

Florin Capitanescu, Mevludin Glavic, *Senior Member, IEEE*, Damien Ernst, *Member, IEEE*, and Louis Wehenkel, *Member, IEEE*

**Abstract**—This paper focuses on contingency filtering to accelerate the iterative solution of preventive security-constrained optimal power flow (PSCOPF) problems. To this end, we propose two novel filtering techniques relying on the comparison at an intermediate PSCOPF solution of post-contingency constraint violations among postulated contingencies. We assess these techniques by comparing them with severity index-based filtering schemes, on a 60- and a 118-bus system. Our results show that the proposed contingency filtering techniques lead to faster solution of the PSCOPF, while being more robust and meaningful, than severity-index based ones.

**Index Terms**—Contingency filtering, optimal power flow, security-constrained optimal power flow, static security control.

## I. INTRODUCTION

THE Optimal Power Flow (OPF) problem was introduced in the 1960s [1], [2]. The main limitation of the OPF is that it focuses on the optimization of a single system configuration at the time, while the system operator needs to know 1) how robust the system is with respect to various credible contingencies and 2) how (much it costs) to meet also post-contingency constraints. While item 1) can be tackled by a mere security analysis performed at the OPF solution, item 2) led to the extension of the OPF formulation into the Security-Constrained Optimal Power Flow (SCOPF) formulation [3], which takes into account pre-contingency constraints as well as post-contingency ones. Modern SCOPF software handle also dynamic security constraints, most often expressed as surrogate power flow limits for some interfaces [4]–[6].

The SCOPF is in its general form a nonlinear, nonconvex, static, large-scale optimization problem with both continuous and discrete variables. It aims at optimizing some objective by acting on available control means while satisfying some equality constraints (e.g., power flow equations) and inequality constraints (e.g., physical and operational limits). In this paper, we focus on the “preventive” [3] SCOPF (called hereafter PSCOPF) which, contrary to the “corrective” SCOPF [7], does not consider the possibility of re-scheduling controls in post-contingency states, except of those with automatic response to contingencies (e.g., active power of generators

participating in frequency control, automatic tap-changers, etc.).

The major difficulty of the PSCOPF problem is its high dimensionality, especially for large systems and/or when many contingencies are considered [8]. Trying to solve this problem directly for a large power system, by imposing simultaneously all post-contingency constraints, would lead to prohibitive memory requirements and CPU times. Moreover, and although in real life applications most contingencies do not constrain the optimum, including them all into the PSCOPF problem increases the complexity of the computations by shrinking the feasible region, and can lead to algorithmic/numerical problems. This is especially true under stressed operating conditions, i.e., when the PSCOPF solution is most useful.

A widely used approach to mitigate these drawbacks combines three modules: a PSCOPF which considers a subset of potentially active contingencies, a (steady-state) security analysis (SSSA), and a contingency filtering (CF) technique [3], [9]–[11]. This approach requires iteration between these modules until all post-contingency constraints are met. The PSCOPF can also be simplified by adding to the base case constraints only relevant post-contingency inequalities, linearized around the optimized base case, while dropping all post-contingency equality constraints (which are checked at the optimal solution) [4], [6], [9], [12]. Many approaches use also a linear approximation of the problem [3], [9]–[11]; this, however, is questionable under highly loaded conditions. Finally, to speed-up computations, the parallelization of computations has been proposed [11], [13], where the PSCOPF problem is decomposed and distributed among several processors, each one solving independently a limited subset of post-contingency states. As regards the CF techniques, many of them rank contingencies by a severity index (SI) and select those yielding a SI above some threshold [3], [9]–[12], [14]. These SIs are typically computed from post-contingency quantities obtained with various load flow models, except [14] which exploits Lagrange multipliers of a relaxed PSCOPF solution.

In this paper, we adopt an iterative PSCOPF algorithm and propose two CF techniques based on the comparison of post-contingency violations aiming to identify efficiently a minimal subset of contingencies which inclusion in the PSCOPF problem provides the same solution as the use of the complete set of contingencies. We compare these techniques with several SI-based ranking techniques on two test systems.

The rest of the paper is organized as follows. Section II presents the PSCOPF problem and its solution strategy. Then, we describe in Section III several approaches for contingency filtering. Section IV provides numerical results for two PSCOPF problems. The conclusions are given in Section V.

Manuscript received August 29, 2006; revised March 13, 2007. The work of D. Ernst and M. Glavic was supported by the Belgian National Fund of Scientific Research (FNRS). Paper no. TPWRS-00554-2006.

The authors are with the Department of Electrical Engineering and Computer Science, University of Liège, B28, B-4000 Liège, Belgium (e-mail: capitane@montefiore.ulg.ac.be; glavic@montefiore.ulg.ac.be; dernst@ulg.ac.be; l.wehenkel@ulg.ac.be).

Digital Object Identifier 10.1109/TPWRS.2007.907528

## II. PREVENTIVE SECURITY-CONSTRAINED OPTIMAL POWER FLOW

### A. Statement of the Problem

The benchmark PSCOPF problem can be compactly formulated as follows:

$$\min_{\mathbf{x}_0, \dots, \mathbf{x}_c, \mathbf{u}_0} f_0(\mathbf{x}_0, \mathbf{u}_0) \quad (1)$$

$$\text{s.t.} \quad \mathbf{g}_k(\mathbf{x}_k, \mathbf{u}_0) = \mathbf{0} \quad k = 0, \dots, c \quad (2)$$

$$\mathbf{h}_k(\mathbf{x}_k, \mathbf{u}_0) \leq \mathbf{0} \quad k = 0, \dots, c \quad (3)$$

where  $f_0$  is the (real-valued) objective function,  $\mathbf{g}_k$  (resp.  $\mathbf{h}_k$ ) is the set of equality (resp. inequality) constraints for the  $k$ th system configuration ( $k = 0$  corresponds to the base case, while  $k = 1, \dots, c$  corresponds to the  $k$ th post-contingency state,  $c$  being the number of contingencies considered),  $\mathbf{x}_k$  is the vector of state variables (i.e., complex voltages) for the  $k$ th configuration and  $\mathbf{u}_0$  is the vector of base case control variables (e.g., active and reactive generator powers, transformer ratios, shunt element reactances, loads, etc.).

Equation (2) mainly refers to base case and post-contingency network power flow equations. Inequality constraint (3) concerns physical limits (e.g., bounds on: active and reactive generator powers, controllable transformer ratios, shunt element reactances, etc.), steady-state operational limits (e.g., on branch currents and voltage magnitudes), and possibly dynamic security limits [6]. Notice that the OPF problem corresponds to equations (1)–(3) when  $c = 0$ . Thus, the PSCOPF problem size is  $(c + 1)$  times larger than the OPF problem size.

In our simulations, we use the Interior-Point Method (IPM) together with a rectangular formulation of the state variables in order to solve the PSCOPF problem. For more details on our implementation, the interested reader may refer to [15].

### B. Solving the PSCOPF Problem Iteratively

Our aim is to quickly identify in an iterative fashion the set of *binding* contingencies, i.e., those that in the full PSCOPF lead to active post-contingency constraints other than those active in the base case OPF. To this purpose, we use an iterative algorithm to solve the PSCOPF problem, where at each step a subset (denoted by  $\mathcal{C}_b$ ) of potentially binding contingencies is identified and included in the optimization problem:

- 1) Let  $\mathbf{P}_0$  be the operating point to be optimized and  $\mathcal{C} = \{1, \dots, c\}$  the contingency set with respect to which we want the system to be secure; let  $\mathcal{C}_b \leftarrow \emptyset$ .
- 2) Solve the PSCOPF by including, besides base case constraints, only the post-contingency constraints for the subset  $\mathcal{C}_b$ . Let  $\mathbf{P}_{\mathcal{C}_b}^*$  denote the optimal operating point.
- 3) Simulate each contingency in  $\mathcal{C} \setminus \mathcal{C}_b$  at  $\mathbf{P}_{\mathcal{C}_b}^*$ . If none of them leads to violations,  $\mathbf{P}_{\mathcal{C}_b}^*$  is a secure and (locally) optimal solution; the computation terminates. Otherwise, let  $\mathcal{C}_v \subset \mathcal{C} \setminus \mathcal{C}_b$  be the subset of *critical* contingencies, i.e., those leading to some post-contingency violations.
- 4) Identify a set  $\mathcal{C}_b^+ \subset \mathcal{C}_v$  of additional potentially binding contingencies. Let  $\mathcal{C}_b \leftarrow \mathcal{C}_b \cup \mathcal{C}_b^+$ ; go to step 2.

Observe that, since initially  $\mathcal{C}_b$  is empty, the step 2 of the first iteration is in fact an OPF computation with only base case constraints. Alternatively, this OPF computation could be skipped and replaced by a first security analysis at  $\mathbf{P}_0$ . The main advantage of using the first OPF computation is that, if the resulting operating point turns out to be secure with respect to the contingency set  $\mathcal{C}$ , there is no need to iterate on PSCOPF computations. Additionally, binding contingencies can be sooner revealed by the SSSA, especially if the OPF outcome is close to the sought security-constrained optimum.

Step 3 of the algorithm is a reduced SSSA. The simulation of the system response to contingencies at this step is performed by a classical full ac load flow computation software. Thus, if a post-contingency load flow solution is found one reports post-contingency (branch current and voltage magnitude) limits violation. We call a contingency which leads to such violations (or to load flow divergence) at an optimal operating point  $\mathbf{P}_{\mathcal{C}_b}^*$  at some stage, a *critical* one (it belongs to the subset  $\mathcal{C}_v$ ). Note that the SSSA acts also as a contingency pre-filter, since at each iteration noncritical contingencies (the subset  $\mathcal{C} \setminus (\mathcal{C}_b \cup \mathcal{C}_v)$ ) are filtered-out. Obviously, a non-critical contingency at an iteration may become critical during subsequent iterations, and eventually (potentially) binding.

The subsequent sections propose and compare performances of various heuristic CF techniques, to be used at step 4 of the algorithm, in order to accelerate the PSCOPF solution.

## III. CONTINGENCY FILTERING APPROACHES

In this section, we propose two novel techniques to identify binding contingencies based on the concept of constraint violation domination. We also describe several SI based ranking techniques inspired by the literature, to which we will compare our methods in Section IV.

### A. Nondominated Contingency Filtering Approaches

Intuitively, a critical contingency is *dominated* at a given step of the PSCOPF procedure, if the constraint violations of other ones are larger. Our conjecture is that keeping only nondominated contingencies may lead to a solution that is very close to the PSCOPF solution obtained by keeping them all.

Let us denote by  $h_{kr}(\mathbf{P}_{\mathcal{C}_b}^*)$  the left-hand value of constraint  $r$ ,  $\forall r \in \{1, \dots, q\}$ , relative to any contingency  $k \in \mathcal{C} \setminus \mathcal{C}_b$ , where  $q$  is the size of any vector  $\mathbf{h}_k$  in (3). This quantity is computed at step 3 of the iterative PSCOPF procedure after simulating contingency  $k$  at  $\mathbf{P}_{\mathcal{C}_b}^*$ . Let us further define by

$$h_{kr}^+(\mathbf{P}_{\mathcal{C}_b}^*) = \max(0, h_{kr}(\mathbf{P}_{\mathcal{C}_b}^*)) \quad (4)$$

the constraint violation (by extension, we set this quantity to  $+\infty$ ,  $\forall r \in \{1, \dots, q\}$ , for contingencies leading to load flow divergence).

1) *Individually Nondominated Contingency (INDC) Technique*: We say that contingency  $k$  *individually dominates* contingency  $j$  if

$$h_{kr}^+(\mathbf{P}_{\mathcal{C}_b}^*) \geq h_{jr}^+(\mathbf{P}_{\mathcal{C}_b}^*), \quad \forall r \in \{1, \dots, q\} \quad (5)$$

and the inequality is *strict* for at least one  $r \in \{1, \dots, q\}$ ,  $h_{kr}^+$ , and  $h_{jr}^+$  referring to the same type of constraint.

We say that contingency  $k$  is *individually dominated* in  $\mathcal{C}_v$  if there is a contingency  $j \in \mathcal{C}_v \setminus \{k\}$  that dominates it. The set of these contingencies and its complement are respectively denoted by  $\mathcal{IDC}$  and  $\mathcal{INDC}$  ( $\mathcal{INDC} = \mathcal{C}_v \setminus \mathcal{IDC}$ ).

When filtering contingencies according to this technique, we identify at step 4 of the iterative PSCOPF the members of  $\mathcal{C}_b^+$  in the following way:

- 1) Initialization: let  $\mathcal{IDC} = \emptyset$ .
- 2) For each contingency  $j \in \mathcal{C}_v$  do:
  - a) For each contingency  $k \in \mathcal{C}_v \setminus (\{j\} \cup \mathcal{IDC})$  do:

If  $k$  dominates  $j$  let  $\mathcal{IDC} \leftarrow \mathcal{IDC} \cup \{j\}$ .
- 3) Let  $\mathcal{C}_b^+ = \mathcal{C}_v \setminus \mathcal{IDC}$ .

Note that among the nondominated contingencies, two or more could in principle lead to exactly the same violations. However, this is extremely unlikely to happen unless the contingency list contains several contingencies consisting in the loss of identical network elements. The latter situation can be avoided by removing from the contingency list the loss of identical network elements before running the PSCOPF.

2) *Nondominated Contingency Group (NDCG) Technique:* We define the nondominated contingency group as a minimal subset of  $\mathcal{C}_v$  containing for each constraint a contingency among those that lead to the largest violation for this constraint. We denote such a set by  $\mathcal{NDCG}$  and by  $\mathcal{DCG}$  its complement in  $\mathcal{C}_v$ . Their relationship is as follows:

$$\begin{aligned} \forall j \in \mathcal{DCG}, \forall r \in \{1, \dots, q\} \\ \exists k \in \mathcal{NDCG} : h_{kr}^+(\mathbf{P}_{\mathcal{C}_b}^*) \geq h_{jr}^+(\mathbf{P}_{\mathcal{C}_b}^*). \end{aligned} \quad (6)$$

With this technique, we construct  $\mathcal{C}_b^+$  in the following way:

- 1) Initialization: let  $\mathcal{NDCG} = \emptyset$ .
- 2) For each constraint  $r \in \{1, \dots, q\}$  do:
  - a) Let  $\overline{dh} = 0$  and  $i(\overline{dh}) = 0$ .
  - b) For each contingency  $k \in \mathcal{C}_v$  do:

If  $h_{kr}^+(\mathbf{P}_{\mathcal{C}_b}^*) > \overline{dh}$ , let  $\overline{dh} \leftarrow h_{kr}^+(\mathbf{P}_{\mathcal{C}_b}^*)$  and  $i(\overline{dh}) = k$ .
  - c) If  $i(\overline{dh}) \neq 0$ , set  $\mathcal{NDCG} \leftarrow \mathcal{NDCG} \cup \{i(\overline{dh})\}$ .
- 3) Let  $\mathcal{C}_b^+ = \mathcal{NDCG}$ .

We show in the Appendix that at any iteration of the PSCOPF we have  $\mathcal{NDCG} \subseteq \mathcal{INDC}$ .

### B. Severity indexes for Contingency Ranking

In this section we present a contingency ranking scheme together with four SIs. The first three indexes are obtained from a relaxed PSCOPF problem, where a single contingency is incorporated at the time [14]. The last one is based on post-contingency violations, derived from SSSA applied to the PSCOPF

solution. Notice that within the iterative PSCOPF procedure, the value of these SIs will be refreshed at every iteration in order to rank the contingencies and to select among the top ranked ones a pre-specified number.

Let the PSCOPF problem reduced to the  $k$ th contingency be stated as follows:

$$F_k = \min_{\mathbf{x}_0, \mathbf{x}_k, \mathbf{u}_0} f_0(\mathbf{x}_0, \mathbf{u}_0) \quad (7)$$

$$\text{s.t.} \quad \mathbf{g}_0(\mathbf{x}_0, \mathbf{u}_0) = \mathbf{0} \quad (8)$$

$$\mathbf{h}_0(\mathbf{x}_0, \mathbf{u}_0) \leq \mathbf{0} \quad (9)$$

$$\mathbf{g}_k(\mathbf{x}_k, \mathbf{u}_0) = \mathbf{0} \quad (10)$$

$$\mathbf{h}_k(\mathbf{x}_k, \mathbf{u}_0) \leq \mathbf{0}. \quad (11)$$

The Lagrangian associated to this problem is defined, in the IPM, as

$$\begin{aligned} L_\mu(\mathbf{z}) = f_0(\mathbf{y}_0) - \mu \sum_{j=1}^q \ln s_{0j} - \mu \sum_{j=1}^q \ln s_{kj} + \boldsymbol{\lambda}_0^T \mathbf{g}_0(\mathbf{y}_0) \\ + \boldsymbol{\lambda}_k^T \mathbf{g}_k(\mathbf{y}_k) + \boldsymbol{\pi}_0^T [\mathbf{h}_0(\mathbf{y}_0) + \mathbf{s}_0] + \boldsymbol{\pi}_k^T [\mathbf{h}_k(\mathbf{y}_k) + \mathbf{s}_k] \end{aligned}$$

with  $\mathbf{z} = [\mathbf{y}_0, \mathbf{y}_k, \mathbf{s}_0, \mathbf{s}_k, \boldsymbol{\lambda}_0, \boldsymbol{\lambda}_k, \boldsymbol{\pi}_0, \boldsymbol{\pi}_k]^T$ ,  $\mathbf{y}_0 = [\mathbf{x}_0, \mathbf{u}_0]^T$ , and  $\mathbf{y}_k = [\mathbf{x}_k, \mathbf{u}_0]^T$ . The primal variables are  $\mathbf{y}_0, \mathbf{y}_k, \mathbf{s}_0, \mathbf{s}_k$ , the dual ones are  $\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_k, \boldsymbol{\pi}_0, \boldsymbol{\pi}_k$ , and  $\mu$  is the barrier parameter. The slack variables satisfy  $\mathbf{s}_0, \mathbf{s}_k > \mathbf{0}$ .

Note that by extension, (7)–(11) with  $k = 0$  actually correspond to the OPF problem without any post-contingency constraint.

The first three indexes are defined as follows:

$$SI_1(k) = \|\mathbf{W}_{1k} \boldsymbol{\lambda}_k^*\|_2 \quad (12)$$

$$SI_2(k) = 100 \left( \frac{F_k^*}{F_0^*} - 1 \right) \quad (13)$$

$$SI_3(k) = \|\mathbf{W}_{3k} \boldsymbol{\pi}_k^*\|_2 \quad (14)$$

where the superscript  $*$  stresses the fact that the quantities are relative to the final solution of the reduced PSCOPF (7)–(11) computed by the IPM.  $F_0^*$  denotes the value of the objective function at the solution of the OPF without contingency constraints and  $\mathbf{W}_{1i}$  denotes a diagonal matrix of weights, which unless explicitly specified will be the identity matrix of appropriate dimension;  $\|\cdot\|_2$  denotes the Euclidean norm.

$SI_1$  is adapted from [14] to the ac model. The components of  $\boldsymbol{\lambda}_k^*$  are the sensitivities of the objective function to a small active or reactive load increase at a bus in the  $k$ th post-contingency situation. The rationale of ranking contingencies according to this index relies on the conjecture that the higher the sensitivity of the objective function of the  $k$ th relaxed PSCOPF problem, the higher the chances that the  $k$ th contingency is binding at the optimum.

$SI_2$  defines the severity of contingency  $k$  by its impact on the objective function, in relative terms with respect to the optimized base case.

$SI_3$  is similar to  $SI_1$ , except that instead of the Lagrange multipliers of the equality constraint (10), it uses those of the inequality constraint (11), i.e.,  $\boldsymbol{\pi}_k^*$ . The contingency ranking according to  $SI_3$  can thus be interpreted as follows: the more the

TABLE I  
TEST SYSTEMS SUMMARY

system	$n$	$g$	$d$	$b$	$l$	$t$	$o$	$s$
Nordic32	60	23	22	81	57	31	4	12
IEEE118	118	54	91	186	175	11	9	14

objective is altered by post-contingency constraint limit variations, the more constraining the contingency.

The fourth index that we will consider is defined as follows (see [3], [9], and [10]):

$$SI_4(k) = \|\mathbf{W}_{4k} \mathbf{h}_k^+(\mathbf{P}_{C_b}^*)\|_2 \quad (15)$$

where  $\mathbf{P}_{C_b}^*$  denotes the solution at the current iteration of the PSCOPF procedure, and where  $\mathbf{h}_k^+$  is the vector of inequality constraint violations obtained when simulating contingency  $k$  at this point, or  $+\infty$  if the post-contingency load flow diverges.

In addition to these four SI-based ranking methods inspired from the literature, we also compare our method to the *direct* PSCOPF method, which consists of including all contingencies at once, and to an iterative PSCOPF *without filtering*, that includes at each iteration all the contingencies of  $C_v$ .

#### IV. NUMERICAL RESULTS

In this section we present representative numerical results obtained with the proposed approaches on two test systems: a 60 bus system, which is a modified variant of the Nordic32 system [16], and the popular IEEE118 system. A summary of their characteristics is given in Table I, where  $n$ ,  $g$ ,  $d$ ,  $b$ ,  $l$ ,  $t$ ,  $o$ , and  $s$  denote the number of: buses, generators, loads, branches, lines, all transformers, transformers with controllable ratio and shunt elements, respectively. All tests have been performed on a PC Pentium IV, 1.7-GHz, 512-Mb RAM.

We consider a list of contingencies containing the outage of any single transmission line that does not create islands. In the case of identical lines, we keep only one of them.

##### A. Results Using the IEEE118 Test System

We focus on the problem of minimizing the overall generation cost. Control variables are the generator active/reactive powers. Equality constraints are the bus active/reactive power balance equations, and inequality constraints are bounds on generator active/reactive powers and limits on branch currents. They apply both in pre- and in post-contingency states. We use the same branch thermal limit in all states.

1) *Application of the Proposed Approaches:* We first use the algorithm of Section II-B, while filtering contingencies with the approaches of Section III-A. Since for this example, both INDC and NDCG approaches provide exactly the same results, we will discuss only those obtained with INDC.

We first run the base case OPF (no contingency constraints) and observe that 3 branch currents are at their maximum value at the optimal solution. A SSSA at this point results in 15 contingencies (out of 166) leading to violations. Among them, there are 4 nondominated ones which are thus included in the PSCOPF computation. Its solution yields 18 active constraints (2 pre-contingency and 16 post-contingency ones) and the subsequent SSSA yields 5 critical contingencies, of which 3 are nondominated and hence added to the potentially binding ones.

TABLE II  
BINDING CONSTRAINTS AT SUCCESSIVE PSCOPF SOLUTIONS

iter	$P_g$	$Q_g$	$I$	total
0	3	0	3	6
1	4	1	13	18
2	3	1	21	25

TABLE III  
SSSA REPORT AT SUCCESSIVE PSCOPF SOLUTIONS

iter	$ncc$	$ncv$	$\Delta I_m(\%)$	$\Delta I_a(\%)$
0	15	19	81.6	10.6
1	5	5	28.7	14.1
2	0	0	-	-

TABLE IV  
BINDING CONTINGENCIES AT SUCCESSIVE PSCOPF SOLUTIONS

iter	binding contingencies
1	31, 84, 96
2	31, 84, 96, 32, 69

TABLE V  
NONDOMINATED AND DOMINATED CONTINGENCIES

iter	non-dominated contingencies	dominated contingencies
1	31, 46, 84, 96	19, 26, 27, 35, 36, 93, 103, 104, 105, 113, 163
2	32, 35, 69	29, 33

Running the PSCOPF with the augmented set of contingencies yields 21 active branch current constraints (2 in pre-contingency state and 19 in the 7 post-contingency states), but SSSA performed at its optimum results in no further violations; thus the sought optimum has been reached. Note that, at this optimum the 2 active branch current constraints in the base case are also binding in all considered post-contingency states. As a matter of fact, only 5 contingencies (leading to different active branch current constraints than in the base case) are actually binding at the final optimum.

These results are gathered in Tables II–V. Table II provides the number and type of binding constraints at the optimum of the OPF and subsequent PSCOPFs. The columns labeled with  $P_g$ ,  $Q_g$ , and  $I$  refer to constraints relative to generator active power, generator reactive power and branch current, respectively. Table III displays the SSSA report at the optimal solution provided by the OPF and PSCOPFs, where  $ncc$ ,  $ncv$ , and  $\Delta I_m$  (resp.  $\Delta I_a$ ) represent the number of critical contingencies, the number of violated constraints, and the maximum (resp. the average) branch overload among all critical contingencies. Table IV shows the binding contingencies at the PSCOPF solution for the 2 iterations, while Table V reports INDC contingency filtering results.

By comparing Tables IV and V, one can observe the good accuracy of the INDC technique. Indeed, only one contingency (number 35) among the total of 13 dominated ones becomes critical (but nonbinding) at the next iteration.

2) *Comparison With Alternative Approaches:* To assess the computational efficiency of the proposed filtering technique, we compare it to the other methods discussed in Section III-B. To this end, we provide in Table VI the overall CPU times (in seconds) of all tasks during the PSCOPF sequential solution for

TABLE VI

approach	iter	PSCOPF (card( $C_b$ ))	SSSA	CF	overall time
DA	-	82.9 (166)	-	-	82.9
WF	0	0.5 (0)	2.4	-	46.3
	1	18.7 (15)	2.1	-	
	2	20.5 (18)	2.1	-	
INDC	0	0.5 (0)	2.4	0.0	20.9
	1	4.8 (4)	2.3	0.0	
	2	8.7 (7)	2.2	-	
$SI_1$	0	0.5 (0)	2.4	11.4	50.1, ..., 61.3
	1	4.6 (4)	2.3	3.6	
	2	10.3 (8)	2.2	0.0	
	3	11.6 (9)	2.2	-	
$SI_2$	0	0.5 (0)	2.4	11.4	50.1, ..., 75.4
	1	4.6 (4)	2.3	3.6	
	2	10.3 (8)	2.2	0.0	
	3	11.6 (9)	2.2	-	
$SI_3$	0	0.5 (0)	2.4	11.4	44.3, ..., 61.3
	1	3.2 (3)	2.3	3.1	
	2	6.9 (6)	2.2	0.0	
	3	10.1 (8)	2.2	-	
$SI_4$	0	0.5 (0)	2.4	0.0	29.2, ..., 61.8
	1	9.7 (8)	2.2	0.0	
	2	12.2 (10)	2.2	-	

the different methods, including as baselines the direct approach (denoted DA) and the sequential approach without filtering (denoted WF).

In Table VI,  $\text{card}(C_b)$  denotes the number of potentially binding contingencies handled in the PSCOPF iterative approach at every iteration. As regards the filtering schemes relying on SIs, we provide the range of the overall CPU time, when the maximal number of top ranked contingencies (which we denote by  $M$ ) allowed to be additionally included into the PSCOPF at every iteration, varies between 1 and 10, while the times associated to the steps of the iterative PSCOPF solution correspond to their fastest PSCOPF solution only.

Note first that all iterative PSCOPF approaches, with or without filtering, are (significantly) faster than the direct approach. We also observe that the INDC filtering technique clearly outperforms all other filtering schemes as well as the WF scheme. On the other hand, among the SI-based schemes  $SI_4$  is the most efficient one, while among those relying on relaxed PSCOPF formulations it is  $SI_3$ .

We derive also from Table VI that the computing times of the CF task are negligible (order of milliseconds) for the INDC (and NDCG) and  $SI_4$  techniques, while for  $SI_1$  to  $SI_3$  they are not. This is due to the fact that these latter require to solve a relaxed PSCOPF problem for each critical contingency detected at a given iteration. We note however that these calculations could be done in parallel, and that by doing so it would have been possible to reduce their computing times to similar values to those of the  $SI_4$ -based scheme.

The superior performances of the INDC technique are due to its higher accuracy and its very low computational requirements. The accuracy of a CF technique is its ability to identify binding contingencies while leading to a small number of false alarms. Ideally, the CF scheme should filter out from the list of critical contingencies all those that are not binding at the final optimum. Thus, an accurate filtering scheme includes a small number of nonbinding contingencies and a large number of binding ones. Normally, the more accurate the CF scheme is, the smaller is the number of iterations until convergence to the final optimum, and the faster is the overall procedure.

TABLE VII

VALUES OF VARIOUS SIs FOR THE CRITICAL CONTINGENCIES

nb cont	INDC NDCG	$SI_1$ (pu)	$SI_2$ (%)	$SI_3$ (pu)	$SI_4$ (%)
<b>31</b>	yes	6.34	4.03	2.23	83.3
<b>35</b>	no	3.98	0.22	1.56	4.6
<b>84</b>	yes	2.00	0.06	0.73	4.8
<b>96</b>	yes	11.3	0.53	3.08	7.2
19	no	3.28	0.04	1.29	0.8
26	no	3.06	0.18	1.43	5.4
27	no	6.00	1.88	2.33	32.9
36	no	3.25	0.04	1.25	0.8
46	yes	4.37	0.27	1.28	7.3
93	no	8.39	1.19	2.28	21.3
103	no	4.09	0.14	1.57	3.0
104	no	3.28	0.07	1.31	1.5
105	no	3.24	0.06	1.26	1.3
113	no	3.60	0.07	1.38	1.5
163	no	3.35	0.06	1.32	1.4

Table VII provides more details about the filtering at the first iteration of the algorithm by the INDC technique and the four SIs. To identify the 4 binding contingencies (shown in bold in the first column of this table) at the first PSCOPF solution (by considering all 15 critical contingencies) by means of the four SIs one needs to include in the PSCOPF, under best circumstances,<sup>1</sup> 15, 11, 15, and 8 top ranked contingencies, if one uses the  $SI_1$ ,  $SI_2$ ,  $SI_3$ , and  $SI_4$ , respectively. Thus, except  $SI_4$  to some extent, none of these filtering schemes is accurate enough. When using the INDC approach one keeps only 4 contingencies (see Table VII), missing one binding contingency (number 35) while correctly identifying the 3 others. A fair comparison of the INDC technique with SI-based ones requires identification of only 3 binding contingencies. To achieve this objective, one needs to include in the PSCOPF 8, 8, 6, and 7 top ranked contingencies, respectively, if one uses the  $SI_1$ ,  $SI_2$ ,  $SI_3$ , and  $SI_4$ , respectively. Furthermore, in order to identify the 3 contingencies among the 15 critical ones which are binding at the final PSCOPF optimum (see Table IV) one has to include in the PSCOPF: 4, 15, 11, 15, and 7 contingencies, according to the filtering techniques based on INDC,  $SI_1$ ,  $SI_2$ ,  $SI_3$ , and  $SI_4$ , respectively. Therefore, the INDC approach is much more accurate than the SI-based CF schemes.

Note that the better identification of binding contingencies with the INDC technique with respect to the popular  $SI_4$  is due to the fact that the latter filters contingencies somewhat "blindly," looking only at (a certain norm of) the total amount of constraints violation, but disregarding whether various violations are relative to the same power system element or not. For instance, contingencies 27 and 93 are high ranked according to  $SI_4$ , while being discarded according to INDC because both are dominated by contingency 31.

3) *Discussion of the Effect of Parameters:* Since the severity based ranking schemes depend on the number  $M$  of contingencies selected at each iteration and the weight matrices, we made some side experiments in order to assess the impact of these parameters on their efficiency.

Table VIII shows the performances of the four methods with different values of  $M$ , in terms of overall CPU time and number of potentially binding contingencies at each step. Note that in the

<sup>1</sup>Depending on the predefined maximum number of contingencies allowed to be kept for the inclusion into the PSCOPF

TABLE VIII  
SIS PERFORMANCE DEPENDING ON THE MAXIMAL NUMBER OF  
CONTINGENCIES ADDITIONALLY INCLUDED IN PSCOPF AT AN ITERATION

$M$	$SI_1$		$SI_2$		$SI_3$		$SI_4$	
	time	card( $\mathcal{C}_b$ )	time	card( $\mathcal{C}_b$ )	time	card( $\mathcal{C}_b$ )	time	card( $\mathcal{C}_b$ )
1	57.1	1,2,3,4,5,6	51.0	1,2,3,4,5	58.7	1,2,3,4,5,6	38.2	1,2,3,4,5,6
2	58.2	2,4,6,7,8	59.1	2,4,6,7,8	49.9	2,4,5,6	43.9	2,4,6,7,8
3	51.3	3,6,7,8	75.4	3,6,9,10,11	44.3	3,6,8	61.8	3,6,9,10,11
4	50.1	4,8,9	50.1	4,8,9	50.1	4,8,9	54.1	4,8,12,13
5	56.5	5,10,11	56.5	5,10,11	56.5	5,10,11	40.2	5,10,11
6	52.4	6,8,9	53.7	6,9,10	52.4	6,8,9	42.7	6,11,12
7	53.9	7,9,10	54.5	7,10,11	53.9	7,9,10	30.0	7,12
8	55.2	8,10,11	56.9	8,11,12	55.2	8,10,11	29.2	8,10
9	58.4	9,11,12	59.3	9,12,13	58.4	9,11,12	31.3	9,11
10	61.3	10,12,13	62.8	10,13,14	61.3	10,12,13	32.7	10,12

SI-based techniques the filter is activated (and the SI computed), only if the number of critical contingencies exceeds the value of  $M$ .

We first observe that changing the value of  $M$  results in rather large CPU time variations. Although, this effect is more pronounced for  $SI_4$ , one can nevertheless maintain the conclusion that using  $SI_4$  is significantly more efficient than using any of the other three indices. On the other hand,  $SI_4$  and  $SI_3$  appear to be (slightly) more accurate (yielding a smaller number of iterations and/or resulting in a smaller number of contingencies handled at each iteration) than  $SI_1$  and  $SI_2$ . Although for specific values of  $M$  (e.g.,  $M = 2$ ,  $M = 6$ , etc.) some of  $SI_1$  to  $SI_3$  are (much) more accurate, the scheme based on  $SI_4$  remains nevertheless faster. This is essentially due to the computational burden of solving additional relaxed PSCOPF at each iteration, induced by the other methods. As already mentioned, we note that by computing in parallel relaxed PSCOPF solutions, the three SIs, and especially  $SI_3$ , could equally compete with  $SI_4$ .

To evaluate the impact of the weight matrices on the performances of the SI-based methods, we made an experiment with  $SI_1$  keeping the weights of the dual variables corresponding to post-contingency real power balance equations constant (and equal to 1) while varying between 0 and 100 those related to reactive power balance equations. Under these conditions, the CPU time of the whole PSCOPF solution varies between 46.8 s and 64.1 s. These results seem to indicate that the values of the weighting matrices do not influence too much the conclusions that we have drawn up to now. We will check this further in the Section IV-B3.

### B. Results Using the Nordic32 Test System

We now concentrate on the minimization of generation cost by means of a “full” PSCOPF. For this problem, we allow the following control variables: generator active/reactive powers, controllable transformer ratios and shunt reactances. Equality constraints are again the bus active/reactive power balance equations, while inequality constraints include bounds on all control variables and branch currents, as well as limits on bus voltage magnitudes. Both apply in pre- and post-contingency states. The bus voltage magnitudes are allowed to vary between 0.95 pu (resp. 0.92 pu) and 1.05 pu (resp. 1.08 pu) in pre-contingency (resp. post-contingency) state.

1) *Filtering by the INDC Technique:* When using the INDC filtering technique, the PSCOPF algorithm of Section II-B converges after 2 iterations. The main results are gathered in Ta-

TABLE IX  
BINDING CONSTRAINTS AT SUCCESSIVE PSCOPF SOLUTIONS

iter	$P_g$	$Q_g$	$I$	$V$	$r$	$x$	total
0	20	0	1	23	0	2	46
1	14	1	4	23	0	6	48
2	13	2	5	26	0	7	53

TABLE X  
SSSA REPORT AT SUCCESSIVE PSCOPF SOLUTIONS

iter	$ncc$	$ncvI$	$ncvV$	$\Delta I_m(\%)$	$\Delta I_a(\%)$	$\Delta V_m(\%)$	$\Delta V_a(\%)$
0	17	30	2	74.9	24.9	3.5	2.7
1	3	2	1	17.0	16.4	1.5	1.5
2	0	0	0	-	-	-	-

TABLE XI  
BINDING CONTINGENCIES AT SUCCESSIVE PSCOPF SOLUTIONS

iter	binding contingencies
1	5, 12, 13, 18, 31
2	5, 12, 13, 18, 21, 30, 31

TABLE XII  
NONDOMINATED AND DOMINATED CONTINGENCIES

iter	non-dominated contingencies	dominated contingencies
1	5, 12, 13, 15, 18, 19, 20, 22, 23, 31	11, 14, 16, 17, 21, 29, 33
2	21, 26, 30	-

bles IX–XII. Table IX yields the number and type of binding constraints at the PSCOPF optimum for successive iterations. The columns labeled with  $V$ ,  $r$ , and  $x$  refer to constraints relative to bus voltage magnitudes, controllable transformer ratios, and shunt reactances, respectively.

Table X reports the results of the SSSA at successive iterations, where  $ncvI$  (resp.  $ncvV$ ) and  $\Delta V_m$  (resp.  $\Delta V_a$ ) represent the total number of branch current (resp. voltage magnitude) limit violations and the maximum (resp. the average) voltage magnitude limit violation among all critical contingencies.

Table XI provides the binding contingencies at the PSCOPF optimum obtained at 2 successive iterations. Recall that we consider a contingency to be binding from the voltage magnitude viewpoint if a voltage magnitude constraint is active in post-contingency situation, but not in the base case.

Table XII reports the filtering results at successive iterations. One can observe that 7 critical contingencies are filtered out after the first run of the SSSA. Among these 7 contingencies, only one (#21) reveals itself to be binding at the final optimum (see Table XI). Note however, that this contingency leads only to branch current violations at the first iteration (where it is dominated), while it leads voltage constraint violations only at the second iteration.

2) *Comparison of Different Schemes:* Table XIII reports, for various contingency filtering options, CPU times of all tasks during the PSCOPF sequential solution. As regards the filtering schemes relying on SIs, we provide the range of the overall CPU time, when  $M \in [1, 10]$ , while the times associated to the steps of the iterative PSCOPF solution correspond to their fastest PSCOPF solution only.

Once again, the filtering techniques INDC, NDCG and, to a lesser extent,  $SI_4$  lead to the highest overall time gain for the PSCOPF solution with respect to the direct approach. This

TABLE XIII  
CPU TIMES OF VARIOUS APPROACHES TO THE PSCOPF SOLUTION

approach	iter	PSCOPF (card( $C_b$ ))	SSSA	CF	overall time
DA	-	22.6 (33)	-	-	22.6
WF	0	0.5 (0)	0.4	-	21.8
	1	9.6 (17)	0.2	-	
	2	11.0 (19)	0.1	-	
INDC	0	0.5 (0)	0.4	0.0	15.4
	1	6.0 (10)	0.3	0.0	
	2	8.0 (13)	0.2	-	
NDCG	0	0.5 (0)	0.4	0.0	14.9
	1	5.6 (8)	0.3	0.0	
	2	7.9 (12)	0.2	-	
$SI_1$	0	0.5 (0)	0.4	13.6	29.6,.....50.1
	1	6.2 (10)	0.3	0.0	
	2	8.4 (14)	0.2	-	
$SI_2$	0	0.5 (0)	0.4	13.6	35.0,.....48.4
	1	2.1 (3)	0.4	3.2	
	2	4.0 (6)	0.3	0.0	
	3	4.7 (7)	0.3	0.0	
$SI_3$	0	0.5 (0)	0.4	13.6	24.4,.....46.9
	1	3.6 (5)	0.3	0.0	
	2	5.8 (10)	0.2	-	
$SI_4$	0	0.5 (0)	0.4	0.0	16.1,.....24.2
	1	2.9 (4)	0.3	0.0	
	2	5.5 (8)	0.3	0.0	
	3	5.9 (9)	0.3	-	

TABLE XIV  
VALUES OF VARIOUS SIS FOR THE CRITICAL CONTINGENCIES

nb cont	INDC	NDCG	$SI_1$ (pu)	$SI_2$ (%)	$SI_3$ (pu)	$SI_4$ (%)
5	yes	yes	2.06	1.85	1.00	30.2
12	yes	yes	3.76	5.22	0.06	123.0
13	yes	no	2.83	3.85	0.08	56.1
18	yes	yes	3.60	5.78	0.09	80.6
21	no	no	2.49	0.20	0.10	2.0
31	yes	yes	0.32	0.03	1.48	1.9
11	no	no	0.38	0.23	0.02	20.9
14	no	no	1.37	0.03	0.06	0.8
15	yes	yes	2.92	3.65	0.09	68.8
16	no	no	1.88	0.02	0.08	0.4
17	no	no	2.94	0.52	0.08	10.8
19	yes	yes	2.84	2.70	0.09	30.6
20	yes	no	3.83	0.59	0.07	10.9
22	yes	yes	3.29	2.80	0.10	28.9
23	yes	yes	3.59	2.83	0.09	30.3
29	no	no	1.80	0.03	0.07	0.30
33	no	no	1.46	0.02	0.06	0.27

gain is lower in this example than for the previous one, due to the higher proportion of binding contingencies at the final optimum (7 out of 33, compared to 5 out of 166 in the previous example) and the difficulty to reveal all these binding contingencies at the first iteration. For these reasons, the DA is faster than the iterative PSCOPF using any among the three SIs stemming from relaxed PSCOPF formulations. We recall that these latter suffer from the time required by the filtering task and could benefit from parallel computations. Among them,  $SI_3$  is again the fastest because of its very good accuracy, even better here than that of INDC and NDCG techniques.

Let us now compare the filtering accuracy of the 17 critical contingencies at the first iteration of the algorithm by the INDC and NDCG techniques with those of the four SIs. The values of these SIs are given in Table XIV.

When running the PSCOPF with the 17 critical contingencies we found that 6 of them (shown in bold in Table XIV) are binding at the first PSCOPF solution as well as at the final optimum (see Table XI). To identify these 6 binding contingencies by means of  $SI_1$ ,  $SI_2$ ,  $SI_3$ , or  $SI_4$ , one needs to include in the PSCOPF, in ideal conditions, 17, 13, 14, or 13 top ranked ones. Once more none SI-based filtering scheme is sufficiently accurate. To fairly compare INDC/NDCG techniques with SI-based ones it is first required to identify only 5 binding contingencies. To this end, one needs to include in the PSCOPF, 10, 11, 12, 9 or 12 top ranked contingencies, by means of INDC,  $SI_1$ ,  $SI_2$ ,  $SI_3$  or  $SI_4$ , respectively. Therefore, the  $SI_3$  and INDC techniques are again (slightly) more accurate. Finally, to identify only 4 binding contingencies by means of NDCG,  $SI_1$ ,  $SI_2$ ,  $SI_3$ , or  $SI_4$ , one needs to include in the PSCOPF, 8, 10, 8, 6, or 7 top ranked ones, respectively. Now schemes based on  $SI_3$  and  $SI_4$  are (slightly) more accurate.

Note that the type of constraints violated by the critical contingencies can be classified as i) only thermal, ii) only voltage, or iii) thermal and voltage (see the SSSA report in Table X). This classification arises a delicate question regarding  $SI_4$ , since its computation formula mixes constraint violations relative to (per unit of) ampères and volts. For this kind of PSCOPF problem the use of  $SI_4$  has little physical sense. Admittedly, this effect can be alleviated by using separate SI formulas for contingencies which violate limits on either branch currents or voltage magnitudes.

3) *Effect of Weighting Factors on the Performance of SI-Based Techniques:* To assess the impact of weighting factors on CPU times, we considered in the computation of  $SI_3$  (resp.  $SI_4$ ) constant weights (equal to 1) for dual variables corresponding to branch current constraints (resp. for branch current limit violation), while varying weights of dual variables related voltage magnitude constraints (resp. for voltage limits violation) between 0 and 100. This resulted in PSCOPF CPU times varying between 22.3 s and 48.7 s (resp. 14.3 s and 26.4 s) with  $SI_3$  (resp.  $SI_4$ ). On the other hand, we made the same experiment as in Section IV-A3, in order to assess the impact of the weight matrix on the performances of  $SI_1$ , and recorded CPU times varying between 25.9 s and 55.5 s.

From these experiments and those reported in Section IV-A3, we conclude that the performances of the severity index based methods are indeed sensitive with respect to the parameters (weights  $\mathbf{W}$ , number of kept contingencies  $M$ ). We also note that the optimal values of these parameters are strongly dependent on the particular case considered.

In this context, we remind that our nondominated contingency filtering techniques are parameter free, and we observe that in all our simulations they remained competitive with the methods from the literature to which we have compared them.

## V. CONCLUSION

In order to speed up the sequential solution of the PSCOPF problem, this paper has introduced two variants of a new contingency filtering scheme based on the concept of "constraint violation domination." Both methods rely on the comparison of post-contingency constraint violations at an intermediate optimal base case of an iterative PSCOPF procedure. The first one consists in discarding those contingencies for which there exists another contingency which leads to larger violations for all constraints. The second one consists in selecting for each

constraint the contingency which leads to the largest violation. Both methods are free of any parameter tuning, which is in contrast with most other methods proposed in the literature. These methods can be coupled with any PSCOPF software package without any adaptation.

The experiments carried out on two power systems of 60 and 118 buses showed that the proposed methods allow to accelerate the sequential solution of a PSCOPF problem in manner which is at least as effective than what could be obtained with various existing severity index based methods, even when the parameters of the latter (such as weight matrices and thresholds) are optimized in a problem specific way.

The excellent robustness and efficiency of the proposed methods suggest to apply them in real-life conditions on very large scale problems. Another direction of further research is the extension of these ideas to security-constrained optimal power flow problems where *corrective* controls are allowed.

#### APPENDIX

It is straightforward to show that  $\mathcal{NDCG} \subseteq \mathcal{INDC}$ , under the assumption that there are no ties in constraint violations.

Indeed, assuming that there are no ties in constraint violations, the following property holds according to (6),  $\forall k \in \mathcal{NDCG}$

$$\exists r \in \{1, \dots, q\} | \forall j \in \mathcal{C}_v \setminus \{k\} : h_{kr}^+(\mathbf{P}_{\mathcal{C}_b}^*) > h_{jr}^+(\mathbf{P}_{\mathcal{C}_b}^*).$$

Therefore, it is impossible that some contingency dominates  $k$  in the sense of definition (5), and hence  $k \in \mathcal{INDC}$ .

While it is in theory possible that  $\mathcal{NDCG} = \mathcal{INDC}$ , quite often the inclusion is strict. We illustrate this by a simple example. Let us consider that  $\mathcal{C}_v$  is composed of three contingencies:  $j$ ,  $k$ , and  $\ell$ . Assume that contingency  $j$  violates only constraint  $r_1$ , i.e.,  $h_{jr_1}^+(\mathbf{P}_{\mathcal{C}_b}^*) > 0$ ; contingency  $k$  violates only constraint  $r_2$ , i.e.,  $h_{kr_2}^+(\mathbf{P}_{\mathcal{C}_b}^*) > 0$ ; and contingency  $\ell$  violates both constraints  $r_1$  and  $r_2$ , i.e.,  $h_{\ell r_1}^+(\mathbf{P}_{\mathcal{C}_b}^*) > 0$  and  $h_{\ell r_2}^+(\mathbf{P}_{\mathcal{C}_b}^*) > 0$ . Assume moreover that:  $h_{jr_1}^+(\mathbf{P}_{\mathcal{C}_b}^*) > h_{\ell r_1}^+(\mathbf{P}_{\mathcal{C}_b}^*)$  and  $h_{kr_2}^+(\mathbf{P}_{\mathcal{C}_b}^*) > h_{\ell r_2}^+(\mathbf{P}_{\mathcal{C}_b}^*)$ .

Thus, although contingency  $\ell$  does not lead to the highest violation for any constraint  $r_1$  or  $r_2$ , it is individually nondominated in the sense of definition (5), since it is neither dominated by contingency  $j$  nor by  $k$ . Thus, according to (5) all three contingencies:  $j$ ,  $k$ , and  $\ell$  are individually nondominated ones and  $\mathcal{INDC} = \{j, k, \ell\}$ . On the other hand, according to the nondominated contingency group definition (6),  $\mathcal{NDCG} = \{j, k\}$ , the highest constraints violations of contingencies  $j$  and  $k$  “cover” together those of contingency  $\ell$ . Thus, in this example  $\mathcal{NDCG} \subset \mathcal{INDC}$ .

#### REFERENCES

- [1] J. Carpentier, “Contribution à l’étude du dispatching économique,” *Bull. Soc. Française D’Electricité*, vol. 3, pp. 431–447, 1962.
- [2] H. W. Dommel and W. F. Tinney, “Optimal power flow solutions,” *IEEE Trans. Power App. Syst.*, vol. PAS-87, no. 10, pp. 1866–1876, Oct. 1968.
- [3] O. Alsac and B. Stott, “Optimal load flow with steady-state security,” *IEEE Trans. Power App. Syst.*, vol. PAS-93, no. 3, pp. 745–751, May 1974.

- [4] S. M. Shahidepour *et al.*, “Nonlinear programming algorithms and decomposition strategies for optimal power flow,” *IEEE Tutorial, Optimal Power Flow: Solution Techniques, Requirements and Challenges*, 1996.
- [5] D. Gan, R. J. Thomas, and R. D. Zimmerman, “A transient stability constrained optimal power flow,” in *Proc. Bulk Power Systems Dynamics and Control IV*, Santorini, Greece, 1998.
- [6] R. A. Schlueter, S. Liu, and N. Alemadi, “Preventive and corrective open access system dispatch based on the voltage stability security assessment and diagnosis,” *Electric Power Syst. Res.*, vol. 60, pp. 17–28, 2001.
- [7] A. J. Monticelli, M. V. P. Pereira, and S. Granville, “Security-constrained optimal power flow with post-contingency corrective rescheduling,” *IEEE Trans. Power Syst.*, vol. 2, no. 1, pp. 175–182, Feb. 1987.
- [8] J. A. Momoh *et al.*, “Challenges to optimal power flow,” *IEEE Trans. Power Syst.*, vol. 12, no. 1, pp. 444–455, Feb. 1997.
- [9] O. Alsac, J. Bright, M. Prais, and B. Stott, “Further developments in LP-based optimal power flow,” *IEEE Trans. Power Syst.*, vol. 5, no. 3, pp. 697–711, Aug. 1990.
- [10] T. J. Bertram, K. D. Demaree, and L. C. Dangelmaier, “An integrated package for real-time security enhancement,” *IEEE Trans. Power Syst.*, vol. 5, no. 2, pp. 592–600, May 1990.
- [11] M. Rodrigues, O. R. Saavedra, and A. Monticelli, “Asynchronous programming model for the concurrent solution of the security constrained optimal power flow problem,” *IEEE Trans. Power Syst.*, vol. 9, no. 4, pp. 2021–2027, Nov. 1994.
- [12] B. Stott, O. Alsac, and A. J. Monticelli, “Security analysis and optimization,” *Proc. IEEE*, vol. 75, no. 12, pp. 1623–1644, Dec. 1987.
- [13] W. Qiu, A. J. Flueck, and F. Tu, “A new parallel algorithm for security constrained optimal power flow with a nonlinear interior-point method,” in *Proc. IEEE Power Eng. Soc. General Meeting*, Jun. 2005, pp. 447–453.
- [14] F. Bouffard, F. D. Galiana, and J. M. Arroyo, “Umbrella contingencies in security-constrained optimal power flow,” in *Proc. 15th Power Systems Computation Conf. (PSCC)*, Liège, Belgium, Aug. 2005.
- [15] F. Capitanescu, M. Glavic, D. Ernst, and L. Wehenkel, “Interior-point based algorithms for the solution of optimal power flow problems,” *Elect. Power Syst. Res.*, vol. 77, no. 5–6, pp. 508–517, Apr. 2007.
- [16] Long-Term Dynamics, Phase II, 1995, CIGRE Task Force 38.02.08.

**Florin Capitanescu** received the electrical power engineering degree and the M.Sc. degree from the University “Politehnica” of Bucharest, Bucharest, Romania, in 1997 and 1998, respectively, and the DEA and Ph.D. degrees from the University of Liège, Liège, Belgium, in 2000 and 2003, respectively.

His main research interests include optimization methods and voltage stability.

**Mevludin Glavic** (M’04–SM’07) received the Ph.D. degree from the University of Tuzla, Tuzla, Bosnia, in 1997. He spent one academic year (1999–2000) with the University of Wisconsin-Madison, as a Fulbright postdoctoral scholar.

He is currently an Invited Professor with the University of Liège, Liège, Belgium, supported by the Belgian National Fund of Scientific Research (FNRS). His research interests include power system control and optimization.

**Damien Ernst** (M’98) received the M.Sc. and Ph.D. degrees from the University of Liège, Liège, Belgium, in 1998 and 2003, respectively.

He is currently a Postdoctoral Researcher at the University of Liège, with Belgian National Fund of Scientific Research (FNRS) support. His main research interests are power system dynamics, reinforcement learning, and multi-agent systems.

**Louis Wehenkel** (M’93) received the Electrical Engineer (electronics) and Ph.D. degrees from the University of Liège, Liège, Belgium, in 1986 and 1990, respectively.

He is a full Professor of electrical engineering and computer science with the University of Liège. His research interests lie in the fields of stochastic methods for systems and modeling, machine learning, and data mining, with applications in power systems planning, operation and control, and bioinformatics.