Electrical Power and Energy Systems 98 (2018) 269–278

Contents lists available at ScienceDirect

Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

Scalable corrective security-constrained economic dispatch considering conflicting contingencies

Yaowen Yu[a,*], Peter Luh[b]

[a] ABB Enterprise Software, San Jose, CA, USA
[b] Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA

ARTICLE INFO

Keywords:
Contingency analysis
Large scale optimization
Power system reliability
Real-time operations
Security-constrained economic dispatch

ABSTRACT

Reliability is an overriding factor in power system operations. Corrective security-constrained economic dispatch (SCED) satisfying the "$N-1$" criterion is difficult because of a large number of contingencies and the strict time limits for real-time operations. The existence of conflicting contingencies further complicates the problem. To overcome these difficulties, this paper develops a new iterative contingency filtering approach to manage "$N-1$" transmission and generator contingencies via decomposition and coordination. Instead of always removing conflicting contingencies as in existing papers, we offer system operators an important option to keep them for increased reliability, enabled by identifying multiple conflicting contingencies simultaneously. To satisfy the strict time requirements in real-time operations, the computational performance of our approach is significantly enhanced by novel warm-start of subproblem models and by parallel computing. Numerical results demonstrate that our new approach is computationally efficient and scalable, and increases the system reliability. In particular, the Polish 2383-bus system with all transmission contingencies is solved within two minutes.

1. Introduction

Reliability is an overriding factor in power system operations. Power engineers make great efforts to "keep the lights on" under normal operation conditions and contingencies. A contingency is an unexpected outage of a component (a transmission line or a generator). To protect power systems against cascading failures and even blackouts, the North American Electric Reliability Corporation (NERC) set, among other reliability standards, the "$N-1$" criterion: in a system that has N components, no single contingency will lead to violations of other components [1]. In real-time wholesale electricity markets, this criterion is considered in economic dispatch (ED), a central operational process. ED is conducted every five minutes to decide how much MW of power each online generator (or unit) should produce to minimize the total generation cost. The version of ED considering the "$N-1$" criterion is known as "security-constrained economic dispatch" (SCED).

1.1. Motivations of corrective SCED

There are two categories of SCED models: preventive and corrective. Preventive SCED is currently practiced to manage transmission contingencies, and requires one set of ED decisions feasible against the base case (under which no contingency happens) and all "$N-1$" transmission contingencies [2]. Such a model restricts ED decisions to remain unchanged from the base-case values right after a contingency occurs. In corrective SCED [3], after a contingency happens, corrective actions can be taken to address the contingency. Corrective SCED models one set of base-case ED decisions and multiple sets of post-contingency ED decisions, one set per contingency. Post-contingency flows that are required to be within corresponding Long-Time Emergency (LTE) ratings in 15 min after a contingency [4,5]. It is ideal to include both preventive decisions to capture the system status right after a contingency happens, and corrective decisions to model the adjustment of post-contingency flows as improved corrective SCED [6]. Moreover, generator contingencies are currently managed by pre-defined reserve requirements based on capacities of particular generators [1]. Since these requirements do not explicitly consider each generator contingency, results may be conservative or infeasible for certain contingencies. In corrective SCED, the output of the tripped generator can be picked up by corrective actions of others. In addition, distributed battery energy storage is used to provide fast corrective actions to quickly alleviate short-term violations [7].

This paper focuses on corrective SCED considering "$N-1$" transmission and generator contingencies. Corrective SCED involves large numbers of post-contingency ED decisions and constraints, and has traditionally been very hard to solve within the timeframe of the real-

* Corresponding author.
E-mail address: yaowen.yu@engineer.uconn.edu (Y. Yu).

https://doi.org/10.1016/j.ijepes.2017.12.011
Received 18 July 2017; Received in revised form 31 October 2017; Accepted 11 December 2017

Nomenclature

c (or c')	index of contingencies, $0 \le c \le L + K$. When $c = 0$, the system is under the base case; when $c = 1, \ldots, L$, the system is under a transmission contingency where line c is tripped; when $c = L + 1, \ldots, L + K$, the system is under a generator contingency where unit $(c - L)$ is tripped
i	index of buses, $1 \le i \le I$
k	index of online units, $1 \le k \le K$
l	index of lines, $1 \le l \le L$
$\alpha(l)$, $\beta(l)$	from and to buses of line l, respectively
$\Phi(i)$	set of units at bus i
$C_k(\cdot)$	increasing continuous piecewise linear generation cost function of unit k (\$)
D_i	demand at bus i (MW)
$f_{l,c}$	power flow along line l under contingency c (MW)
$f_{l,c}{}^{\max}$	rating of line l under contingency c (MW)
M	Penalty factor (\$)
$p_{k,c}$	dispatch decision of unit k under contingency c (MW)
p_k^{min}, p_k^{max}	minimum and maximum generation limits of unit k, respectively (MW)
R_k	Ramp rate of unit k (MW/minute)
$s_{k,c}{}^U$, $s_{k,c}{}^D$	slack variables to relax the ramp-up and ramp-down inequalities of redispatch constraints, respectively (MW)
t_c	time allowed for corrective actions under contingency c (minute)
X_l	reactance of line l (Ω)
y_c	penalty term of contingency c in the master SCED problem (\$)
$\Delta_{k,c}$	maximal allowed variation of unit k under contingency c (MW)
$\theta_{i,c}$	voltage phase angle at bus i under contingency c
ν_c	objective value of contingency subproblem c (\$)
S_A	set of (possibly) active contingencies that have been identified by the contingency filtering approach. Each active contingency has a positive optimal objective value of its corresponding contingency subproblem as formulated in Section 3.3 at the current or any previous iteration
S_C	set of candidate contingencies considered by the contingency filtering approach. It may start with either a predefined contingency set approved by the operator, or the set that contains all possible "$N-1$" contingencies. During the contingency filtering process, contingencies that are included in S_A, S_1 or S_2 are removed from S_C
S_1	set of Type 1 contingencies that have been identified by the contingency filtering approach
S_2	set of Type 2 contingencies that have been identified by the contingency filtering approach

time dispatch [8]. Furthermore, different types of infeasible contingencies, especially conflicting ones, often exist in practical systems and further complicate the solution process [9,10]. It is thus important to identify, differentiate, and manage them.

1.2. Literature review

To solve the corrective SCED problem, there are three typical approaches: the direct approach, contingency filtering, and Benders decomposition. The direct approach considers all possible contingencies and solves the corrective SCED problem as a large linear programming (LP) problem or a large nonlinear programming problem depending on whether the DC or AC power flow model is assumed. Since there are large numbers of decision variables and constraints corresponding to contingencies, the direct approach requires large computer memory and long solution time [11]. In addition, although a pre-screening step that solves the base-case problem together with each contingency sequentially can be used to identify some of the infeasible contingencies, that step can take considerable time and is blind to those contingencies that are conflicting with each other [10].

To reduce the problem size, contingency filtering methods (often considering AC power flow) start with solving the base-case model, and then iteratively add selected active contingencies to update the solution [9,12–14]. The base-case and selected active contingencies were solved in a master problem, while candidate contingencies were checked or ranked in subproblems. The active contingencies were selected by ranking all contingencies based on the severity index (the 2-norm of weighted constraint violations) [12], the rescheduling index (the minimum of the maximal controllable redispatch value) [9], or by using the non-dominated contingency technique (comparing constraint violations) [13]. The non-dominated contingency technique was used together with a network compression method in [14], where a general SCED formulation with both preventive and corrective actions were modeled. In addition, [14] managed discrete variables, including transformer ratios, phase shifter angles and the shunt reactive power, by a progressive round-off method.

Alternatively, Benders decomposition was used to divide the corrective SCED problem into a base-case master problem and multiple contingency subproblems [8,10,11,15,16]. For a given base-case ED solution, feasibility cuts were derived from subproblems and were added to the master problem to update the base-case ED solutions. In [8,11,15,16], AC power flow was considered, and the generalized Benders decomposition was used. In [15], a linear feasibility cut was shifted adaptively according to the constraint violation to alleviate the infeasibility caused by the nonconvexity. Moreover, [15] also investigated a global optimization method based on Lagrangian duality as well as the alternating direction method of multipliers. In [16], semidefinite programming (SDP) was used as convex relaxations of subproblems, and Benders cuts were developed on top of the relaxations. In a recent work [10], DC power flow was considered, and multi-stage redispatch was modeled for transmission contingencies.

Infeasible contingencies were first discussed in [9] where only transmission contingencies were considered. All islanding contingencies, identified in a primary contingency filtering step, were directly removed. Conflicting contingencies were identified and removed one at a time by relaxing the redispatch constraints with penalty terms. In [10], all infeasible contingencies were removed. Removing conflicting contingencies and all islanding ones may decrease system reliability as will be discussed in Section 2.2.

To further improve the performance, the authors of [9] developed a decomposed parallel interior point method to accelerate the solution process, and tested parallel computing by using from 3 to 8 processes. Performance enhancements in [10] included reducing the number of subproblems in iterations, solving subproblems by using the barrier method without crossover, including difficult contingencies within the master problem, and using parallel computing. The overall approach in [10] was able to solve the Polish 2383-bus system with all transmission contingencies within 10 min, using GAMS on a server that had two 3.46 G X5690 Xeon chips with 12 Cores, and 288 GB Memory. A faster approach is still desired to satisfy the strict time requirements in real-time operations.

1.3. Contributions and organization of this paper

This paper focuses on developing a novel contingency filtering approach to solve large-scale corrective SCED problems within the timeframe of real-time dispatch considering conflicting contingencies. DC power flow is used. We acknowledge that DC power flow approximates the underlying AC power flow, and does not model voltage or VAR. Nevertheless, DC power flow fits in the current practice of majority of the real-time markets [10,17,18], especially its linearity suits the economic theories that are used in market design and calculations [17]. Moreover, since corrective SCED contains a large number of post-contingency ED decisions and corresponding constraints for each contingency, it is difficult to solve problems that have thousands of buses and thousands of contingencies in every five minutes. For these cases, a tradeoff needs to be made between modeling accuracy and computational efficiency, and an ED solution with DC power flow and corrective actions is desired [10]. Furthermore, our idea of managing conflicting contingencies, to be presented in Section 3, can be demonstrated with DC power flow. In addition, results from the corrective SCED with DC power flow can be used to provide the initial set of active contingencies to accelerate the solution process of the corrective SCED with AC power flow [19].

The major contributions of this paper are twofold:

(1) Instead of always removing conflicting contingencies as in existing papers [9,10], the new contingency filtering approach provides system operators an important option to keep conflicting contingencies for increased reliability. To enable this option, the method that identified and removed one conflicting contingency at a time in [9] is improved to identify multiple conflicting ones simultaneously.
(2) To satisfy the strict time requirements in real-time operations, the computational performance is significantly enhanced by novel warm-start of subproblem models and by parallel computing. As a result, the Polish 2383-bus system with all transmission contingencies is solved within two minutes.

Section 2 formulates the problem considering "$N-1$" transmission and generator contingencies. With DC power flow, the overall model is a large LP problem. Based on the formulation, formal definitions for different types of infeasible contingencies are provided, and impacts of infeasible contingencies are analyzed.

Section 3 develops the new contingency filtering approach. The problem is decomposed into a master SCED problem and multiple contingency subproblems. Our approach includes all active contingencies into the master SCED problem to accelerate the convergence, instead of ranking active contingencies and only selecting top-ranked ones into the master problem as in existing literature [9,12–14]. In the solution process, infeasible contingencies, especially conflicting ones, are identified and managed. Instead of always removing conflicting contingencies as in existing papers [9,10], we offer system operators an important option to keep them for increased reliability. To enable this option, the method that identified and removed one conflicting contingency at a time in [9] is improved to identify multiple conflicting ones simultaneously within the same master SCED problem.

Section 4 significantly enhances the computational performance by novel warm-start of subproblem models between different contingencies and by parallel computing. The warm-start method is developed to significantly reduce the overhead time of generating large numbers of subproblem models over multiple iterations, and can be applied to both serial and parallel computing.

Numerical results in Section 5 demonstrate that our approach is computationally efficient and scalable for practical use in real-time operations. The Polish 2383-bus system with all transmission contingencies is solved within two minutes. Optimization and simulation results demonstrate a tradeoff between system reliability and the base-case cost when keeping or removing conflicting contingencies.

2. Problem formulation

Section 2.1 formulates the corrective SCED problem and Section 2.2 analyzes infeasible contingencies.

2.1. Real-time corrective SCED formulation

The problem is to minimize the total base-case ED cost by selecting one set of base-case ED decisions $\{p_{k,0}\}$ and multiple sets of post-contingency ED decisions $\{p_{k,c}\}$ $(c \neq 0)$ for online units. The following assumptions are made in this paper:

(1) DC power flow is used as discussed in Section 1.3.
(2) A single time period is considered based on the current practice of the majority of ISOs (e.g., ISO New England [20] and PJM [21]).
(3) Reserves are not included, given that the output of the tripped generator can be picked up by corrective actions of others.
(4) This paper focuses on the corrective actions of generators taken 10 to 15 mins after a contingency happens, so the frequency deviation right after a generator contingency is ignored for simplicity. Frequency-related constraints, such as the limit of the Rate-of-Change-of-Frequency and the Frequency Nadir Limit, can be considered based on [22].

Building on [3], constraints and the objective function are presented as follows.

(1) Transmission constraints: The power flow along line l under contingency c, modeled by DC power flow with voltage phase angles, should be within the corresponding line rating for the positive and negative directions, i.e.,

$$-f_{l,c}^{\max} \leqslant f_{l,c} = \frac{\theta_{\alpha(l),c}-\theta_{\beta(l),c}}{X_l} \leqslant f_{l,c}^{\max},\forall\ l,\forall\ c = 0,1,\ldots,l-1,l+1,\ldots,L+K. \tag{1}$$

Transmission capacity $f_{l,0}{}^{\max}$ is the normal rating under the base case $(c = 0)$, and $f_{l,c}{}^{\max}$ $(c \neq 0)$ is the Long-Time Emergency rating under the contingency case [4,5]. When l is tripped $(c = l)$, its power flow is zero, i.e.,

$$f_{l,l} = 0,\forall\ l. \tag{2}$$

(2) Generator capacity constraints: The dispatch level of unit k under contingency c should be within its minimum and maximum generation limits, i.e.,

$$p_k^{\min} \leqslant p_{k,c} \leqslant p_k^{\max},\forall\ k,\forall\ c = 0,1,\ldots,L+k-1,L+k+1,\ldots,L+K, \tag{3}$$

When unit k is tripped $(c = L+k)$, its dispatch level is zero:

$$p_{k,L+k} = 0,\forall\ k. \tag{4}$$

(3) Nodal flow balance constraints: The net nodal injection (i.e., generation minus demand) at node i equals the total outflow minus the total inflow. The base-case constraints are:

$$\sum_{k\in\Phi(i)} p_{k,0}-D_i = \sum_{l:\alpha(l)=i} f_{l,0}- \sum_{l:\beta(l)=i} f_{l,0},\forall\ i. \tag{5}$$

Under transmission contingency c, the power flow at line c is not included, i.e.,

$$\sum_{k\in\Phi(i)} p_{k,c}-D_i = \sum_{l:\alpha(l)=i,l\neq c} f_{l,c}- \sum_{l:\beta(l)=i,l\neq c} f_{l,c},\forall\ i,\forall\ c = 1,\ldots,L. \tag{6}$$

Similarly, under generator contingency c, the generation of unit $(c-L)$ is not included, i.e.,

$$\sum_{k\in\Phi(i):k\neq c-L} p_{k,c}-D_i = \sum_{l:\alpha(l)=i} f_{l,c}-\sum_{l:\beta(l)=i} f_{l,c}, \forall i, \forall c = L+1,...,L+K. \tag{7}$$

(4) Post-contingency redispatch constraints: Under contingency c, the deviation between the post-contingency dispatch decision and the base-case one for each unit should be within the maximal allowed variation, i.e.,

$$p_{k,0}-\Delta_{k,c} \leqslant p_{k,c} \leqslant p_{k,0}+\Delta_{k,c}, \forall k, \forall c = 1,...,L+k-1, L+k+1,... ,L+K, \tag{8}$$

where the maximal allowed variation is the ramp rate multiplied by the time allowed for corrective actions, i.e.,

$$\Delta_{k,c} = R_k t_c, \forall k, \forall c = 1,...,L+k-1, L+k+1,...,L+K. \tag{9}$$

For a transmission contingency, $t_c = 15$ (minute) [4,5]; for a generator contingency, $t_c = 10$ (minute) (NERC requires the area control error to be recovered within 15 min after a generator contingency [23], and ISO New England uses 10-min reserves to provide a buffer for this requirement [24]).

(5) Objective function: The objective is to minimize the total base-case ED cost based on [3,9–16,19], i.e.,

$$\min_{p_{k,0},p_{k,c}} \sum_k C_k(p_{k,0}). \tag{10}$$

The above corrective SCED model is an LP problem with a large number of contingency ED decisions and corresponding constraints. These decisions are loosely coupled with the base case through (8). Given that constraints (6) exclude the power flow at the tripped line, constraints (2) are redundant. Likewise, constraints (4) are redundant given (7). In addition, (9) can be computed before optimization. The SCED model only needs to include (1), (3), (5)–(8), (10).

2.2. Infeasible contingencies

In practical problems, there does not always exist a feasible solution that satisfies all contingencies. Some of them may incur infeasibility, and load shedding may be necessary. However, even under these infeasible cases, system operators still want to “keep the lights on” as much as possible. It is thus important to understand causes of infeasible contingencies (in this subsection), and to identify and manage them in the solution process (in the next section).

Enlightened by [9,10], we categorize infeasible contingencies under our SCED model into two types.

Definition 1. A contingency is Type 1 if there is no solution to satisfy its contingency-level constraints (1), (3), (6), and (7) simultaneously.

Type 1 contingencies remain infeasible whatever the values of the base-case ED decisions are. Since keeping Type 1 contingencies will not help revise the base-case ED solution to reduce the overall violation, they should be removed from the problem [9,10]. Furthermore, islanding contingencies are not necessarily Type 1 contingencies. As long as the island is not a load bus, it may still be possible to balance both the main grid and the island.

Type 2 contingencies are conflicting contingencies. The reason of conflicting contingencies is that each contingency has its own set of constraints (1), (3), (5)–(7), and all contingencies are coupled with the base case through post-contingency redispatch constraints (8). Consequently, there may not exist a solution to satisfy all constraints simultaneously even if each contingency is feasible. Type 2 contingencies can be further divided into two categories: Type 2a that only conflicts with the base case, and Type 2b that conflicts with other contingencies.

Definition 2. One contingency c is Type 2a if it is not Type 1, and there are insufficient ramp rates to redispatch from the base case to c, i.e., there is no solution to satisfy constraints (1), (3), (5), (6), (7), and (8) for c and the base case simultaneously.

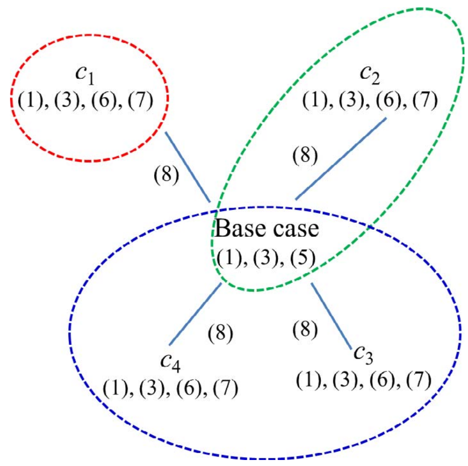

Fig. 1. Illustration of different types of infeasible contingencies. Circles indicate the constraints that cannot be simultaneously satisfied. Contingency c_1 is Type 1, c_2 is Type 2a, and c_3 and c_4 are Type 2b.

Definition 3. Multiple contingencies in a group g are Type 2b if

(1) None of them is Type 1 or Type 2a;
(2) There are insufficient ramp rates to redispatch from the base case to all contingencies in g, i.e., there is no solution to satisfy constraints (1), (3), (5), (6), (7), and (8) for them and the base case simultaneously;
(3) There will be at least one feasible solution if any one contingency from g is removed.

In addition, the above group g is a minimal set of Type 2b contingencies.

Fig. 1 illustrates different types of infeasible contingencies:

Existing methods [9,10] remove Type 2 contingencies, which is questionable because system operators may still want to keep them, so that the base-case ED decisions are pre-positioned to an operating point where the total violation is minimized for increased reliability.

3. Solution methodology

To solve the above problem, Section 3.1 presents key points and the control flow of our new contingency filtering approach. Our approach decomposes the problem into a linear master SCED problem in Section 3.2 and multiple linear contingency subproblems in Section 3.3. The coordination between the master problem and subproblems is described in Section 3.4.

3.1. Key points and the control flow of our approach

Inspired by [9,13], our approach starts with the base-case model, and then fixes the base-case ED decisions and solves subproblems to detect active contingencies that incur violations of redispatch constraints (8). Active contingencies are added to the master problem to revise the base-case ED decisions iteratively. In the process, Type 1 contingences are identified and removed in subproblems, while feasible islanding contingencies are kept. As for Type 2 contingencies, our approach is able to keep them in the master problem for increased reliability, or to remove them for the reduced base-case cost based on the operator’s option.[1] Moreover, the method in [9] that identified and

[1] Quantification of the risk of each choice will involve probabilities that are not

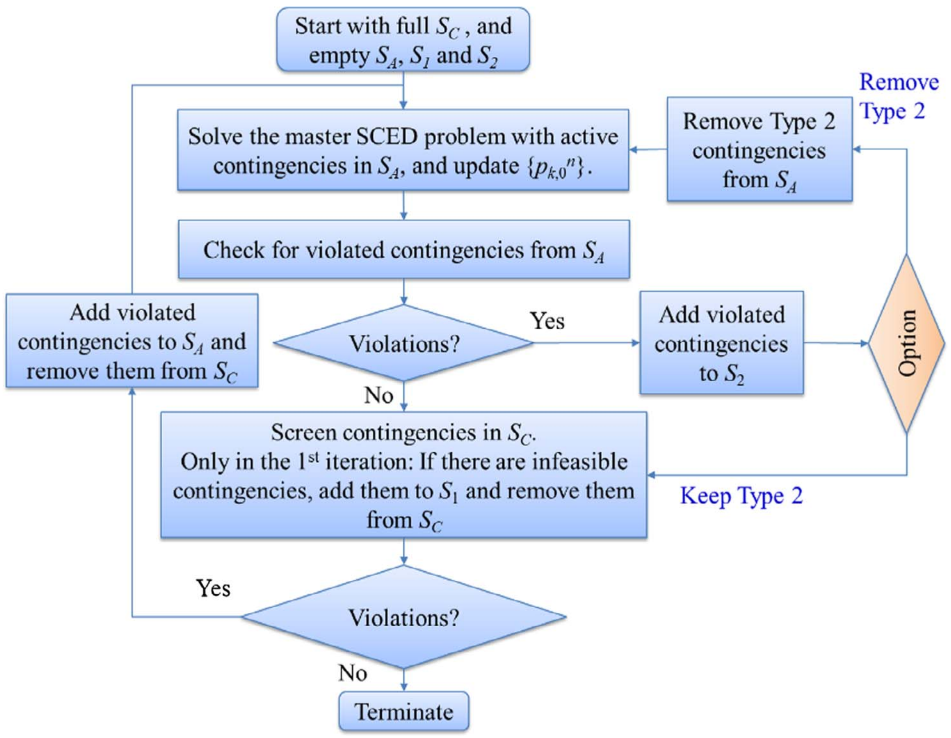
Start with full S_C, and empty S_A, S_1 and S_2
Solve the master SCED problem with active contingencies in S_A, and update $\{p_{k,0}{}^n\}$.
Remove Type 2 contingencies from S_A
Remove Type 2
Check for violated contingencies from S_A
Add violated contingencies to S_A and remove them from S_C
Violations?
Yes
Add violated contingencies to S_2
Option
No
Screen contingencies in S_C. Only in the 1st iteration: If there are infeasible contingencies, add them to S_1 and remove them from S_C
Keep Type 2
Yes
Violations?
No
Terminate

Fig. 2. Flowchart of our contingency filtering approach.

removed conflicting contingencies one at a time cannot be used when Type 2 contingencies are kept, because the second Type 2 contingency could not be identified when the first one is kept. That method is thus improved in our approach to identify simultaneously multiple Type 2 contingencies. The flowchart of our algorithm is in Fig. 2.

The main steps of the solution process are as follows:

1. Initialize S_C to contain all "$N-1$" contingencies, and initialize empty S_A, S_1, and S_2.
2. Solve the master SCED problem with all active contingencies in S_A, including corresponding ED decisions and constraints. Obtain an updated operating point that is delineated by the set of base-case decisions at the n^{th} iteration $\{p_{k,0}{}^n\}$. Check for violated contingencies (from S_A) that incur positive penalty costs in the master SCED problem as described in Section 3.2. Once detected, add such contingencies to the Type 2 contingency set, S_2. Otherwise, directly go to Step 4. Note that at the 1st iteration, the master SCED problem only contains the base case.
3. Proceed to Step 3a or 3b, depending on the option chosen by the operator.

   3a. If keeping Type 2 contingencies, go to Step 4;

   3b. If removing Type 2 contingencies, remove them from S_A and go back to Step 2.
4. Screen all contingencies in S_C to minimize violations by solving contingency subproblems as in Section 3.3.

   4a. (only used in the 1st iteration): If there are infeasible contingencies, add them to the Type 1 contingency set, S_1, and remove them from S_C.

   4b. If there are violated contingencies that have positive objective values in subproblems, add them to S_A and remove them from S_C, and then go back to Step 2. Otherwise, terminate the algorithm as it converges.

(*footnote continued*)
modeled by the standard corrective SCED formulation [3], and is out of the scope of this paper.

3.2. The master SCED problem

In the decomposition, the master problem is formulated as the following linear programing problem:

$$\min_{p_{k,0},p_{k,c},s_{k,c}^U,s_{k,c}^D}\left[\sum_k C_k(p_{k,0})+\sum_{c\in S_A} y_c\right], \tag{11}$$

where

$$y_c = M\sum_{k\neq c-L}(s_{k,c}^U+s_{k,c}^D). \tag{12}$$

$$\text{s.t. } p_{k,c}-s_{k,c}^U \leqslant p_{k,0}^n+\Delta_{k,c}\ ,\forall\ c\in S_A\ ,\forall\ k\neq c-L, s_{k,c}^U\geqslant 0, \tag{13}$$

$$p_{k,0}^n-\Delta_{k,c}-s_{k,c}^D\leqslant p_{k,c}\ ,\forall\ c\in S_A\ ,\forall\ k\neq c-L, s_{k,c}^D\geqslant 0, \tag{14}$$

Constraints (1), (3), and (5)–(7) for $c\in\{0\}\cup S_A$.

Constraints (13) are relaxed versions of right inequalities (ramp-up) of redispatch constraints (8), with non-negative slack variables $s_{k,c}{}^U$. Symmetrically, constraints (14) are relaxed versions of left inequalities (ramp-down) of (8), with non-negative slack variables $s_{k,c}{}^D$. These slack variables are penalized by a penalty factor M (as in (12)) in the objective function (11) to minimize the violation. The contingencies with positive penalty costs y_c are identified as Type 2 contingencies. The value of M should be large (e.g., \$5000/MWh); otherwise, feasible active contingencies may tend to "violate" the relaxed redispatch constraints (13) and (14), and will thus be misidentified as Type 2. Because penalty terms y_c have resolutions on each contingency (with index c), we are able to identify multiple Type 2 contingencies that appear in the master problem simultaneously. Among multiple Type 2b contingencies conflicting with each other, those can be violated with the lowest overall cost will be identified through optimizing (11).[2]

By identifying the Type 2 contingencies, operators have the flexibility of either keeping or removing these conflicting contingencies. If

[2] To further identify the groups of all Type 2b contingencies, different groups (combinations) of contingencies in S_A need to be solved. Such a combinatorial identification problem may not be solvable in real-time and is thus out of the scope of this paper.

Type 2 contingencies are removed, the resulting base case ED solutions are similar to the ones in [9,10]; otherwise, our approach leads to an ED solution that requires smaller violations, or possibly lower penalty cost against Type 2 contingencies than existing approaches.[3] This is because the goal of the management of Type 2 contingencies is to provide a base-case ED solution that minimizes the overall or base-case cost as indicated in the objective function (11) of the master problem.

3.3. Contingency subproblems

Subproblems are formulated to check for violations in contingencies to identify possibly active ones as well as Type 1 contingencies. The subproblem of transmission contingency c given $\{p_{k,0}{}^{n}\}$ is:

$$v_c = \min_{p_{k,c}, s_{k,c}^U, s_{k,c}^D} \left[\sum_k (s_{k,c}^U + s_{k,c}^D) \right], \tag{15}$$

$$\text{s.t. } p_{k,c} \leqslant p_{k,0}^n + \Delta_{k,c} + s_{k,c}^U, \forall k, s_{k,c}^U \geqslant 0, \tag{16}$$

$$p_{k,0}^n - \Delta_{k,c} - s_{k,c}^D \leqslant p_{k,c}, \forall k, s_{k,c}^D \geqslant 0, \tag{17}$$

Constraints (1), (3), and (6) for contingency c.

The feasibility of each contingency depends on the values of $p_{k,0}{}^{n}$. However, in the subproblem, the post-contingency redispatch constraints (8) that couple post-contingency ED decisions $p_{k,c}$ ($c \neq 0$) and $p_{k,0}{}^{n}$ are relaxed by the slack variables $s_{k,c}{}^{U}$ and $s_{k,c}{}^{D}$. In (16), if $p_{k,0}{}^{n}$ reduces by 1 MW to "tighten" the right-hand side, $s_{k,c}{}^{U}$ can increase by 1 MW to offset the impact. In (17), if $p_{k,0}{}^{n}$ increases by 1 MW to "tighten" the left-hand side, $s_{k,c}{}^{D}$ can increase by 1 MW to offset the impact. Consequently, the feasibility of each subproblem does not depend on the values of $p_{k,0}{}^{n}$. If this subproblem with the relaxed constraints is infeasible, then contingency c is Type 1. Moreover, a positive optimal objective value, $v_c{}^{*}$, indicates that contingency c is active.[4] When $v_c{}^{*} = 0$, contingency c is feasible and inactive.

Similarly, the subproblem of generator contingency c is:

$$v_c = \min_{p_{k,c}, s_{k,c}^U, s_{k,c}^D} \left[\sum_{k \neq c-L} (s_{k,c}^U + s_{k,c}^D) \right], \tag{18}$$

$$\text{s.t. } p_{k,c} \leqslant p_{k,0}^n + \Delta_{k,c} + s_{k,c}^U, \forall k \neq c-L, s_{k,c}^U \geqslant 0, \tag{19}$$

$$p_{k,0}^n - \Delta_{k,c} - s_{k,c}^D \leqslant p_{k,c}, \forall k \neq c-L, s_{k,c}^D \geqslant 0, \tag{20}$$

Constraints (1), (3), and (7) for contingency c.

3.4. Coordination between the master problem and subproblems

The coordination between the master problem and subproblems are through the iterative process as in Fig. 2. When all contingencies are feasible, our approach can be proven to have global convergence. When there are infeasible contingencies, all Type 1 contingencies can be identified and removed at the 1st iteration, since the feasibility of each subproblem does not depend on the values of $p_{k,0}{}^{n}$. Type 2 contingencies are identified in the master SCED problem.[5]

[3] In the practice of ISOs or TSOs, there does not always exist a feasible solution, and constraints relaxations (violations) with penalty prices are widely used [25]. Moreover, as will be demonstrated in Case 1 of Example 2, keeping Type 2 contingencies (with non-zero slack variables in the master problem) in optimization will reduce the violation in simulation and thus improve the overall reliability, as compared to removing Type 2 contingencies.

[4] The definition of active contingency in this paper is specific for the contingency filtering approach, and is different from the meaning of an active constraint where the equality holds.

[5] Type 2 contingencies are conflicting contingencies and can only be identified in the master problem where more than one contingencies are considered. Although $p_{k,0}{}^{n}$ are included in contingency subproblems, the values are fixed from the previous master problem solution and may change in the next iteration. Consequently, contingency

Based on our observations, most of the solution time is on screening contingencies at Step 4 in Section 3.1, while the solution time of solving the master problem is relatively small. Therefore, the number of iterations (and the resulting number of contingencies to be screened) has a higher impact on the overall solution time than the size of the master problem does. Consequently, our approach includes all active contingencies identified in subproblems in S_A. In this way, all constraints of identified active contingencies are included in the master problem, so the algorithm converges fast (within 2 to 3 iterations for examples tested in Section 5). In contract, Benders decomposition methods [8,10,11,15,16] included the feasibility cuts derived from active contingencies into the master problem at an iteration instead of all corresponding constraints, so each active contingency may need to be checked over multiple iterations to generate multiple cuts. Consequently, it may take more iterations to converge, and an additional remedy was included in [10] to reduce the number of subproblems in iterations.

In our approach, only a minimum amount of information needs to be communicated. From the master problem to a subproblem, the base-case ED decisions are passed. From a subproblem to the master, the solution status and the objective value are passed.

4. Performance enhancements

To enhance the performance of our approach, Section 4.1 significantly reduces the overhead through our new warm-start method of subproblem models, and Section 4.2 discusses parallel computing implementations.

4.1. Warm-start method of subproblem models between different contingencies

The overhead issue can be a "performance killer" when applying optimization methods to practical use. For the corrective SCED problem, the overhead mainly occurs when creating models for all subproblems in software. The overhead time of generating a new LP model for each subproblem may be comparable to or even more than the CPU time of solving it, and there can be thousands of subproblems at each iteration. However, this issue has not been discussed in existing papers related to the corrective SCED problem to the best of our knowledge. Although modeling languages (e.g., AIMMS [26]) and APIs (such as CPLEX C+ + API [27] and Gurobi C+ + API [28]) support functions to modify created optimization models, its direct application is to modify the subproblem model of the same contingency among different iterations. Since there are a large number of contingencies, a large number of subproblems still have to be created at the first iteration.

To overcome this difficulty, we explore the control flow of our algorithm and structures of subproblem models, and develop a new warm-start method of subproblem models between different contingencies. This method creates, over all iterations, only two subproblem models for the first transmission contingency and the first generator contingency. Created models are then reused with the fewest number of modifications between subproblems.

The contingency screening procedure (in Step 4) of our approach in serial computing is detailed as in Fig. 3. The procedure starts with subproblem 1 in S_C, and then checks subproblem 2, and so forth. Our method only requires creating a model for subproblem 1 and can then modify this model to represent subproblem 2, and so forth.

To make the fewest modifications from one subproblem to another, we analyze structures of two subproblems corresponding to transmission contingencies c and c'. Software does not have to treat decision variables corresponding to these two contingencies differently, and the

(*footnote continued*)
subproblems will not be able to identity Type 2 contingencies.

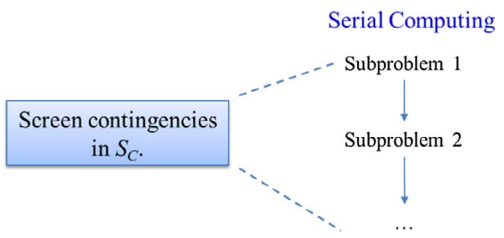

Fig. 3. Contingency screening procedure of our contingency filtering approach in serial computing.

Table 1
Comparison between our warm-start of subproblem models and creating all subproblem models in serial computing.

Constraints	Creating all models	Our novel warm-start method	
	# of constraints created	# of constraints created	# of constraints modified
(1)	$2(L-1)\times L$	$2(L-1)$	$4\times(L-1)$
(3)	$2K\times L$	$2K$	0
(6)	$I\times L$	I	$4\times(L-1)$
(8)	$2K\times L$	$2K$	0

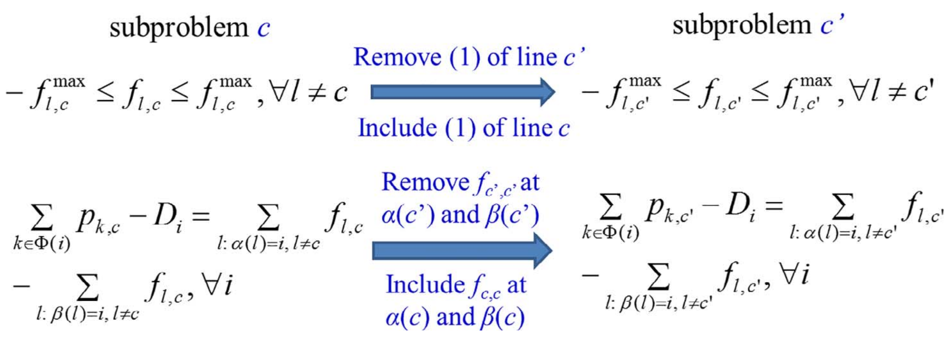
subproblem c
subproblem c'
$-f_{l,c}^{\max}\le f_{l,c}\le f_{l,c}^{\max},\forall l\ne c$
Remove (1) of line c'
Include (1) of line c
$-f_{l,c'}^{\max}\le f_{l,c'}\le f_{l,c'}^{\max},\forall l\ne c'$
$\sum_{k\in\Phi(i)}p_{k,c}-D_i=\sum_{l:\alpha(l)=i,l\ne c}f_{l,c}-\sum_{l:\beta(l)=i,l\ne c}f_{l,c},\forall i$
Remove $f_{c',c'}$ at $\alpha(c')$ and $\beta(c')$
Include $f_{c,c}$ at $\alpha(c)$ and $\beta(c)$
$\sum_{k\in\Phi(i)}p_{k,c'}-D_i=\sum_{l:\alpha(l)=i,l\ne c'}f_{l,c'}-\sum_{l:\beta(l)=i,l\ne c'}f_{l,c'},\forall i$

Fig. 4. Warm-start between two subproblems of transmission contingencies.

objective functions and most of the corresponding constraints are essentially the same between c and c'. The only places that differentiate c and c' are transmission constraints (1) and nodal flow balance constraints (6) as illustrated in Fig. 4. Under contingency c, the power flow at the tripped line $f_{c,c}$ is excluded from these constraints. A similar exclusion holds for contingency c'. Based on this observation, our method only removes two transmission constraints (for positive and negative directions) corresponding to line c' and then includes two corresponding to line c. Similarly, at most four nodal flow balance constraints are modified. This process starts in the first iteration and continues for the remaining ones. As a result, we only need to create one model for all transmission contingency subproblems.

The numbers of operations of our warm-start are compared to those from creating models for all subproblems in Table 1. The total number of operations to create and modify constraints is significantly reduced. Our warm-start method is similar for generator contingencies and is not presented for conciseness.

4.2. Parallel computing

Even after the reduction of overhead, it can still be time-consuming to solve a large number of contingency subproblems. Solving them in parallel can reduce the time. Commercial solvers, such as CPLEX and Gurobi, directly provide the functionality of multithreaded parallelization [27,28], where an optimization problem is solved in parallel on multiple threads of a local computer. When applied to the corrective SCED problem, multithreaded parallelization solves each subproblem on multiple threads, while different subproblems are still solved in serial (as illustrated in Fig. 3).

A different parallelism solves multiple subproblems on different threads in parallel. In implementation, we adopt the "remote object for distributed parallel optimization" [29] (referred as the "remote object" for the rest of the paper) provided by CPLEX. One master process is used to solve the master SCED problem and control the algorithm flow, and multiple worker processes are used to solve subproblems. Multithreaded parallelization can also be applied within the remote object, so that each subproblem is solved in multiple threads at a lower level and multiple subproblems are parallelized at an upper level. Supported communication protocols between the master and each worker include Secure Shell (SSH), Message Passing Interface (MPI), and Transmission Control Protocol/Internet Protocol (TCP/IP).

The new warm-start method can also be applied in such a parallelism. The resulting contingency screening procedure is illustrated in Fig. 5, assuming W workers. For each worker, the new warm start method of subproblem models is applied (see Table 2). W subproblems will be created at the beginning, and there will be $L-W$ remaining subproblems. Take transmission constraints (1) for example. As illustrated in Fig. 4, four transmission constraints (1) need to be modified from one subproblem to another, so there will be $4\times(L-W)$ transmission constraints (1) to be modified. The new warm-start method can still significantly reduce the overhead in the usual situation where W is much smaller than L. For example, when there are thousands of contingencies, there may be 100 cores available for parallel computing.

5. Numerical results

Two problems are tested to demonstrate properties of our contingency filtering approach. In Example 1, the IEEE RTS with three areas is tested to illustrate that our approach is able to manage feasible islanding contingencies. In Example 2, the Polish 2383-bus system is tested to validate that the option to keep Type 2 contingencies offered by our approach can increase the system reliability. It also demonstrates the computational efficiency of our approach with enhancements developed in Section 4. Testing is conducted on a laptop with an Intel i7-6920HQ 2.90 GHz CPU (4 cores and 8 threads), 32 GB memory, and Windows 10, where CPLEX 12.6.1 is called by using OPL. In addition, Case 3 of Example 2 also utilizes the Storrs HPC cluster [30] with CPLEX 12.6.1 called by its C++ API.[6]

5.1. Example 1

The IEEE RTS with three areas [31] is tested. There are 120 transmission lines, 72 conventional units, and 192 resulting "$N-1$" contingencies. The penalty factor M in (12) is set at \$5000/MWh in our approach. For benchmarking, the direct approach is tested by solving the full-size LP problem. The active contingencies in the direct

[6] Testing data and results are available at http://www.engr.uconn.edu/msl/J1_IEEE.htm.6.

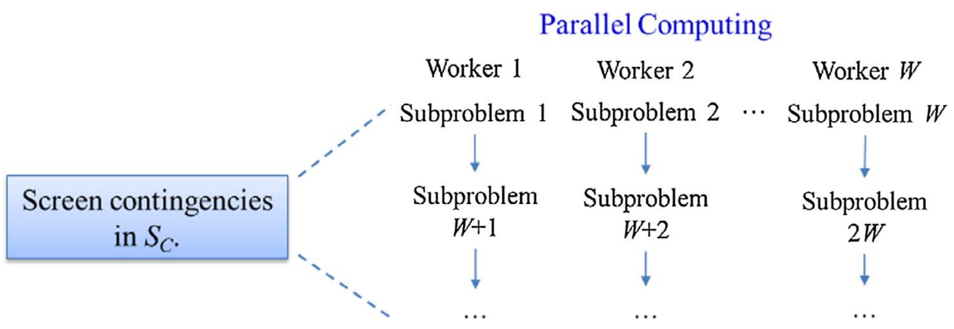

Fig. 5. Contingency screening procedure of our contingency filtering approach in parallel computing.

Table 2
Comparison between our warm-start of subproblem models and creating all subproblem models in parallel computing.

Constraints	Creating all models	Our novel warm-start method	
	# of constraints created	# of constraints created	# of constraints modified
(1)	$2(L-1)\times L$	$2(L-1)\times W$	$4\times(L-W)$
(3)	$2K\times L$	$2K\times W$	0
(6)	$I\times L$	$I\times W$	$4\times(L-W)$
(8)	$2K\times L$	$2K\times W$	0

Table 3
Optimization and Simulation Results of Case 1 in Example 2.

		Keep Type 2	Remove Type 2
Optimization	Wall clock time (s)	36	39
	Total cost (k $)	4244.24	1855.99
	Penalty cost (k$)	2326.11	0
Base-case ED cost (k$)		1918.12	1855.99
# of active contingencies in S_A		3	2
Simulation 1: Penalty factor = $5000/MWh	Total cost (k $)	4244.24	6917.39
	Penalty cost (k$)	2326.11	5061.40
Simulation 2: Penalty factor = $500/MWh	Total cost (k $)	2150.73	2362.13
	Penalty cost (k$)	232.61	506.14

Table 4
Computational performance on Laptop of Case 3 in Example 2.

Configuration	a	b	c	d
Language	OPL	C++	C++	C++
Subproblem models	Creating all	Creating all	Warm-start	Warm-start
Parallelization	Multi-threaded	Multi-threaded	Multi-threaded	Remote object
Wall clock time	40 min 08 s	2 h 11 min 30 s	8 min 34 s	3 min 20 s
CPU time	5 min 30 s	18 min 17 s	8 min 25 s	3 min 03 s [a]
Overhead time	34 min 38 s	1 h 53 min 13 s	9 s	17 s
Overhead/CPU time ratio	629.70%	619.23%	1.78%	9.29%
Speedup ratio of wall clock time	3.28	1	15.35	39.45

[a] This CPU time is the sum of the CPU time of the slowest subproblem in each group.

Table 5
Computational Performance on HPC of Case 3 in Example 2.

Configuration	b	c	d
Wall clock time	21 min 42 s	7 min 53 s	1 min 51 s
CPU time	16 min 07 s	7 min 52 s	1 min 45 s
Overhead time	5 min 35 s	1 s	6 s
Overhead/CPU time ratio	34.64%	0.21%	5.71%
Speedup ratio of wall clock time	1	2.75	11.73

approach only include those with active post-contingency redispatch constraints (8) at the optimal solution. The active contingencies in our approach, however, include all contingencies that have positive optimal objective values of the corresponding contingency subproblems as formulated in Section 3.3 at the last or any previous iteration (if not identified and removed as Type 2). As a result, our approach may identify more active contingencies than the direct approach.

It turns out that there is no active or infeasible contingency. The total costs of two approaches are the same, $74,441. The direct approach takes 5.26 s of wall clock time, while our approach converges at the 1st iteration in 4.09 s.

To better illustrate our approach, ramp rates of all units are reduced to 7.7% of their original values to make some of the contingencies active. In this case, both approaches have the same cost $78,229. The direct approach takes 8.41 s of wall clock time and identifies three active contingencies (feasible with binding constraints (8)): 144, 168, and 192, corresponding to Units 24, 48, and 72, respectively. Our approach takes 8.80 ss and converges in two iterations. The set of active contingencies, S_A, is empty before the 1st iteration. Two transmission contingencies (49 and 87) and 17 generator contingencies (140–144, 157, 158, 164–168, and 188–192) are identified as active in the 1st iteration, and are included in S_A. No further active contingency is identified in the 2nd iteration.

One important finding of this case is that Contingencies 49 and 87 have islanding issues based on the topology. Since there is sufficient generation on each islanding bus, the demand can still be met. For example, Contingency 49 trips Line 49 that connects Buses 31 and 32 (Buses 207 and 208, respectively, in Fig. 4 of [31]), and islands Bus 31. However, there are three 100-MW units located at Bus 31 with demand of 125 MW, and the power on this islanding bus can be balanced. Contingency 89 that islands Bus 55 (Bus 307 in [31]) is similar. Since these two islanding contingencies are feasible, they may need to be kept in the contingency filtering process. Otherwise, the base-case solution could be positioned such that unnecessary infeasibilities would occur under these two contingencies. This illustrates that our approach can increase the reliability by managing feasible islanding contingencies.

5.2. Example 2

The Polish 2383-bus system at winter peak [32] is tested. There are 327 units and 2896 transmission lines. Three cases are provided to demonstrate different aspects of our approach. Case 1 compares the options between keeping and removing Type 2 contingencies in our approach. Case 2 illustrates that our approach is able to identify multiple Type 2 contingencies simultaneously. Case 3 demonstrates the computational performance of our approach with the performance enhancements as developed in Section 4.

Case 1. Keeping or removing Type 2 contingencies.

Contingencies 2801–2900 are tested, including 96 transmission contingencies and four generator contingencies. In our approach, the barrier method without crossover, as the fastest LP algorithm provided by CPLEX according to our testing results and consistent with [10], is used to solve the master problem and subproblems. The threshold of the total violation in Step 5 to terminate our approach is 0.001 MW.

After optimization, simulation is conducted to evaluate the consequences of keeping or removing Type 2 contingencies within the algorithm. In this simulation process, base-case ED decisions are fixed at the solution corresponding to each option and one more contingency screening procedure is conducted to solve subproblems of all contingencies that are not Type 1. Optimal objective values of the subproblems in (15) and (18) are multiplied by a penalty factor. In practice, constraint violations are usually penalized by high penalty factors to improve reliability [32]. To limit the impact of the choice of the penalty factor on results, two simulation runs are conducted with different penalty factors: a high penalty factor of $5000/MWh in Simulation 1 and a moderate penalty factor of $500/MWh in Simulation 2. Results are summarized in Table 3.

In optimization, our approach with the option of keeping Type 2 contingencies takes 36 s to converge in two iterations. S_A is {2898, 2899, 2900}, S_1 contains 15 transmission contingencies, and S_2 is {2900}. Our approach with Type 2 contingencies removed takes 39 s to converge in three iterations, among which the 2nd iteration is a small one that only solves the master problem and removes Contingency 2900, as illustrated by the smaller loop in Fig. 2. S_A is {2898, 2899} since Contingency 2900 has been removed, while S_1 and S_2 are the same as corresponding ones when keeping Type 2 contingencies. The total cost corresponds to the optimal objective value of (11) and is much higher when keeping Type 2 contingencies than when removing them. This outcome is caused by Contingency 2900, which has a high penalty cost and raises the base-case ED cost. Comparing the base-case ED costs of $1,918.12 and $1,855.99, the benefit of removing Type 2 contingencies is that the base-case ED cost is reduced.

In both Simulations 1 and 2, keeping Type 2 contingencies incurs a lower penalty cost (and resulting total cost) than removing them. This validates that keeping Type 2 contingencies can increase the system reliability when contingency happens. There is a tradeoff between reliability and the base-case cost when making an option on how to treat Type 2 contingencies. Based on experience, when a Type 2 contingency is likely to happen or has high impacts, the operator tends to keep it. Otherwise, the operator tends to remove it.

Case 2. Identifying multiple Type 2 contingencies simultaneously.

The same contingencies are tested as in the previous case, while ramp rates of all units are reduced by half to create more conflicting contingencies. Our approach converges in two iterations at 36 s when keeping Type 2 contingencies, and converges in three iterations at 37 s when removing them. At the 2nd iteration of either mode, two Type 2 contingencies, 2899 and 2900, are identified at the same time.

Case 3. Performance enhancements.

All 2896 "$N-1$" transmission contingencies are considered in this case. The direct approach is tested by using OPL for benchmarking. The pre-screening step as in [10] is used to identify and remove Type 1 and Type 2a contingencies. The SCED problem considering the base case and the remaining contingencies are then solved as a large LP problem.

The optimal objective value is $1,859.85k with a total solution time of 43 min and 30 s. The pre-screening step takes 26 min and 30 s, and identifies 539 Type 1 contingencies. The large LP problem has a peak memory usage of 22 GB, and takes the wall clock time of 17 min, where the CPU time is 9 min and 32 s, and the overhead time 7 min and 28 s. The pre-screening step is time-consuming, and since there is no decomposition, it is impossible to apply performance enhancements such as warm-start of subproblem models and the remote object to the direct approach.

Our approach is tested in four configurations with their specifications and computational performance summarized in Table 4. There is no Type 2 contingency, so there is no difference between whether Type 2 ones are kept or removed for all configurations. Configuration a is the same as in previous testing. Configurations b, c and d are implemented in C+ +; c and d adopt the warm-start of subproblems; d exploits the remote object with one thread as the master process, seven as worker processes, and communication via SSH.

For all configurations, the algorithm converges in three iterations. The optimal objective value and the number of Type 1 contingencies are the same as corresponding ones obtained by the direct approach. There are 6 active contingencies in S_A: 474, 544, 1075, and 1798 are identified in the 1st iteration, and 20 and 396 in the 2nd iteration.

The overhead/CPU time ratio of Configurations a and b are more than 600%, indicating that creating models for all subproblems is a big burden for the entire solution process. In contrast, by using our warm-start method, Configuration c only has 9 s of overhead, which is 1.78% of the CPU time and is negligible in the wall clock time. This leads to a speedup ratio of 15.35 and demonstrates the benefit of our new warm-start method.

When the remote object is used for parallelization, the speedup ratio is increased to 39.45, demonstrating that the remote object is more efficient than multi-threaded parallelization for this case. Meanwhile, the overhead time is longer than that of Configuration c, since seven subproblems are created for workers in Configuration d.

To further accelerate the solution process, 24 cores (2.60 GHz CPU) at one compute node with 128 GB memory and the Linux operating system in the Storrs HPC Cluster [30] is utilized. Results are summarized in Table 5, where Configuration d uses one core as the master process, 23 as worker processes, and communication via MPI. Configuration a is not tested, because the graphical interface to use OPL is not allowed in our HPC system.

The optimal objective values of all configurations are $1,859.85k, and there are 6 active contingencies in S_A, same as those obtained on the laptop (corresponding to Table 4). Although the overhead times at the HPC environment are shorter than corresponding ones at the PC environment, our warm-start method can still significantly reduce the overhead time from Configuration b to c. In c, the one second overhead time is only 0.21% of the related CPU time. Furthermore, Configuration d exploiting the remote object solves the problem in one minute and 51 s, and the average time of each iteration is 37 s. This performance demonstrates the computational efficiency of our approach for practical use in real-time operations.

Our approach is able to solve the above cases of the IEEE RTS and the Polish 2383-bus system within 2 to 3 iterations in short amounts of time, demonstrating its scalability.

6. Conclusion

This paper develops a new contingency filtering approach to manage "$N-1$" transmission and generator contingencies in real-time corrective SCED via decomposition and coordination. Our approach provides system operators an important option to keep conflicting contingencies for increased reliability, or remove them for reduced base-case costs. The performance is enhanced by novel warm-start of subproblem models and by parallel computing. Our approach solves the Polish 2383-bus system with all transmission contingencies within two minutes, demonstrating its computational efficiency for practical use in real-time operations.

Acknowledgement

This work was supported in part by the National Science Foundation under Grant ECCS-1509666, and by a project funded by ISO New England. The authors would like to thank the contributions of E. Litvinov, T. Zheng, F. Zhao, J. Zhao, D. A. Schiro, and I. Lelic at ISO New England for their work on the original version of this paper.

References

[1] Wood AJ, Wollenberg BF, Sheble GB. Power generation, operation and control. 3rd ed. Hoboken, NJ, USA: Wiley; 2013.
[2] Alsac O, Stott B. Optimal load flow with steady-state security. IEEE Trans Power Ap Syst 1974;PAS-93(3):745–51.
[3] Monticelli A, Pereira MVF, Granville S. Security-constrained optimal power flow with post-contingency corrective rescheduling. IEEE Trans Power Syst 1987;2(1):175–80.
[4] NERC, System Operating Limit Definition and Exceedance Clarification, 2015. [Online]. Available: http://www.nerc.com/pa/Stand/Prjct201403RvsnstoTOPandIROStndrds/2014_03_fifth_posting_white_paper_sol_exceedance_20150108_clean.pdf.
[5] ISO New England Operating Procedure No. 19 – Transmission Operations, 2016. [Online]. Available: http://www.iso-ne.com/rules_proceds/operating/isone/op19/op19_rto_final.pdf.
[6] Capitanescu F, Wehenkel L. Improving the statement of the corrective security-constrained optimal power-flow problem. IEEE Trans Power Syst 2007;22(2):887–9.
[7] Wen Y, Guo C, Kirschen DS, Dong S. Enhanced security-constrained OPF with distributed battery energy storage. IEEE Trans Power Syst Jan. 2015;30(1):98–108.
[8] Stott B, Alsac O. Optimal power flow—basic requirements for real-life problems and their solutions (White Paper). In: Proc. SEPOPE XII Symp., Rio de Janeiro, Brazil, May 2012.
[9] Jiang Q, Xu K. A novel iterative contingency filtering approach to corrective security-constrained optimal power flow. IEEE Trans Power Syst 2014;29(3):1099–109.
[10] Liu Y, Ferris MC, Zhao F. Computational study of security constrained economic dispatch with multi-stage rescheduling. IEEE Trans Power Syst 2015;30(2):920–9.
[11] Capitanescu F, Ramos JLM, Panciatici P, Kirschen D, Marcolini AM, Platbrood L, et al. State-of-the-art, challenges, and future trends in security constrained optimal power flow. Electr Power Syst Res 2011;81(8):1731–41.
[12] Stott B, Alsac O, Monticelli AJ. Security analysis and optimization. Proc IEEE Dec. 1987;75(12):1623–44.
[13] Capitanescu F, Wehenkel L. A new iterative approach to the corrective security-constrained optimal power flow problem. IEEE Trans Power Syst Nov. 2008;23(4):1533–41.
[14] Platbrood L, Capitanescu F, Merckx C, Crisciu H, Wehenkel L. A generic approach for solving nonlinear-discrete security-constrained optimal power flow problems in large-scale systems. IEEE Trans Power Syst 2014;29(3):1194–203.
[15] Phan D, Kalagnanam J. Some efficient optimization methods for solving the security-constrained optimal power flow problem. IEEE Trans Power Syst 2014;29(2):863–72.
[16] Liu Y, Ferris M. Security-constrained economic dispatch using semidefinite programming. In: Proceedings of the 2015 IEEE Power and Energy Society General Meeting, Denver, CO, 2015.
[17] Stott B, Jardim J, Alsac O. DC power flow revisited. IEEE Trans Power Syst Aug. 2009;24(3):1290–300.
[18] Molzahn DK, Dvijotham K. Error bounds on power flow linearizations: a convex relaxation approach. In: FERC Technical Conference: increasing real-time and day-ahead market efficiency through improved software, Washington DC, USA, 2017. [Online]. https://www.ferc.gov/EventCalendar/EventDetails.aspx?ID=8672&CalType=%20&CalendarID=116&Date=06/26/2017&View=Listview.
[19] Marano-Marcolini A, Capitanescu F, Martinez-Ramos JL, Wehenkel L. Exploiting the use of DC SCOPF approximation to improve iterative AC SCOPF algorithms. IEEE Trans Power Syst Aug. 2012;27(3):1459–66.
[20] Lelic I. Introduction to wholesale electricity markets (WEM 101): unit commitment and dispatch, ISO New England, 2014, [Online]. Available: http://www.iso-ne.com/static-assets/documents/2016/04/20160404-06-wem101-unit-commitment-dispatch.pdf.
[21] PJM Energy Market, http://www.pjm.com/markets-and-operations/energy.aspx.
[22] Wen Y, Li W, Huang G, Liu X. Frequency dynamics constrained unit commitment with battery energy storage. IEEE Trans Power Syst Feb. 2016;31(6):5115–25.
[23] NERC, Standard BAL-002-0 – Disturbance Control Performance, 2015. [Online]. Available: http://www.nerc.com/files/bal-002-0.pdf.
[24] ISO New England Operating Procedure No. 8 Operating Reserve and Regulation, 2015. [Online]. Available: http://www.iso-ne.com/rules_proceds/operating/isone/op8/op8_rto_final.pdf.
[25] Al-Abdullah YM, Salloum A, Hedman KW, Vittal V. Analyzing the impacts of constraint relaxation practices in electric energy markets. IEEE Trans Power Syst 2016;31(4):2566–77.
[26] AIMMS. [Online]. Available: http://aimms.com/.
[27] IBM ILOG CPLEX Optimization Studio V12.6.1 documentation. [Online]. Available: http://www.ibm.com/support/knowledgecenter/SSSA5P_12.6.1/ilog.odms.studio.help/Optimization_Studio/topics/COS_home.html.
[28] Gurobi Optimization. [Online]. Available: http://www.gurobi.com/.
[29] Rux S. Applications and Use Cases of the CPLEX Remote API. IBM Software Group, 2014. [Online]. Available: http://www-01.ibm.com/support/docview.wss?uid=swg27044403.
[30] Storrs HPC Cluster, http://hpc.uconn.edu/.
[31] IEEE RTS Task Force of APM Subcommittee. The IEEE reliability test system-1996. IEEE Trans Power Syst 1999;14(3):1010–20.
[32] Polish 2383-bus system at winter peak (case2383wp). [Online]. Available: http://www.neos-guide.org/content/optimal-power-flow.