

An exact and scalable problem decomposition for security-constrained optimal power flow[☆]

Alexandre Velloso^{*,a}, Pascal Van Hentenryck^b, Emma S. Johnson^b

^a Electrical Engineering Department, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil

^b H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, U.S.A.

ARTICLE INFO

Keywords:

Column and constraint generation
Decomposition methods
Primary response
Security-constrained optimal power flow

ABSTRACT

In this paper, we present decomposition techniques for solving large-scale instances of the security-constrained optimal power flow (SCOPF) problem with primary response. Specifically, under each contingency state, we require that the nodal demands are met and that the synchronized units generating below their limits follow a linear model for primary response. The resulting formulation is a mixed-integer linear program since the primary response model introduces disjunctions to the SCOPF problem. Unfortunately, exact methods relying on traditional Benders decomposition do not scale well. As an alternative, we propose a decomposition scheme based on the column-and-constraint-generation algorithm where we iteratively add disjunctions and cuts. We provide procedures for preprocessing dedicated cuts and for numerically determining the post-contingency responses based on the master problem solutions. We also discuss heuristics to generate high-quality primal solutions and upper bounds for the method. Finally, we demonstrate the efficiency of the proposed method on large-scale systems.

1. Introduction

System reliability under contingencies has been widely discussed in the literature. In this context, the goal of the well-known security constrained optimal power flow (SCOPF) problem [1–7] is to produce a pre-contingency (or nominal) schedule for generators at minimal cost, such that it allows for feasible steady-state points of operation for a predefined set of *credible contingencies*. A review of the SCOPF problem, its challenges and trends is available in [4].

The specification of the set of credible contingencies varies in academic works. Generally, a loss of up to one or two elements (generators and/or transmission lines) is considered. Interesting discussions about credible contingencies and reserve requirements can be found in [2,6] and the references therein. Security criteria and regulation for reserves also vary across independent system operators. A survey about the requirements for reserves across U.S. ancillary services can be found in [8]. Without loss of generality, we consider the $N - 1$ criterion for generators in this paper; that is, the system must operate under the loss of any individual generator.

Variants of the SCOPF problem include the *corrective* case [5] where

re-scheduling is possible, and the *preventive* case where no re-dispatch occurs [3,6]. In this work, we consider preventive SCOPF with *primary response* [6]. In this framework, the synchronized generators must be able to automatically respond to contingencies to restore the balance between loads and generation.

Even though SCOPF is a nonlinear and nonconvex problem [9,10], for computational purposes, several authors adopt dc approximations [2,3,6,7]. Some authors have dedicated works to address the quality of approximations or relaxations to practical situations [11–14] or to discuss how to exploit dc SCOPF solutions on more accurate ac SCOPF approaches [15]. Notwithstanding these relevant discussions, the focus of this work is to improve current industry approaches still based on dc SCOPF algorithms. In such models, stability constraints for the system are generally expressed as power flow limits. In practical applications, the solution provided by the dc model can be checked for ac power flow feasibility. Then, iterative and/or heuristic procedures can be applied to further restrict the dc power flow constraints until feasibility is reached.

The primary response of generators is explicitly modeled in [16] for a unit commitment application and in [6,7], and [17] for SCOPF problems. In [6,7], and [16] variables that represent the frequency drop in each contingency state were used to generate linear approximations of

[☆] This research is partly supported by NSF Grant 1912244. The work of Alexandre Velloso is partially supported by Finep/PIPG – Financiadora de Estudos e Projetos / Programa de Incentivo à Pós-Graduação.

^{*} Corresponding author.

E-mail addresses: alevelloso@gmail.com (A. Velloso), pascal.vanhentenryck@isye.gatech.edu (P. Van Hentenryck), ejohnson335@gatech.edu (E.S. Johnson).

<https://doi.org/10.1016/j.epsr.2020.106677>

Received 4 October 2019; Received in revised form 17 April 2020; Accepted 1 August 2020

Available online 18 March 2021

0378-7796/© 2020 Published by Elsevier B.V.

Nomenclature

This section introduces our notation. We use bold symbols for matrices (uppercase) and vectors (lowercase). Additional symbols can be interpreted by the following general rules: Symbols with superscript “(j)” denote new variables, parameters or sets corresponding to the j -th iteration of the solution method. The symbols with superscript “(*)” denote the optimal value of the associated (iterating) variable.

Sets

$\mathcal{E}, \mathcal{E}_s$	Feasibility sets for the nominal power flow constraints and for the power flow constraints under contingency state s , respectively.
$\mathcal{F}_s$	Feasibility set for primary response constraints under contingency state s .
$\mathcal{G}, \mathcal{L}, \mathcal{N}$	Sets of generators, transmission lines and buses, respectively.
$\mathcal{H}$	Subset of $\mathcal{G}$ for devising primal solutions.
$\mathcal{S}$	Set of contingencies.
$\mathbb{S}$	Subset of $\mathcal{S}$ with disjunctive constraints, used in the column-and-constraint-generation algorithm.
$\mathcal{X}, \mathcal{X}_s$	Sets of power flow-related decision variables for nominal state and for contingency state s .
$\mathcal{Y}_s$	Set of decision variables associated with primary response under contingency state s .

Parameters

$\alpha, \alpha_{s,l}$	Largest transmission line capacity violation and violation for transmission line l , under contingency state s .
β, β_1, β_2	Parameters for selecting preprocessed cuts.
γ	Vector of parameters for primary response.
γ_i	Parameter for primary response of generator i .
ϵ	Tolerance for transmission line violation.
ε	Tolerance for the binary search procedure.
A, B	Line-bus and Generator-bus incidence matrices.
c	Vector of generation costs.
c_i	Generation cost of generator i .
d	Vector of nodal net loads.
e	Vector of ones with appropriate dimension.

e_s	Total load imbalance for contingency state s .
$\bar{\mathbf{f}}$	Vector of line capacities.
$\mathbf{g}, \bar{\mathbf{g}}$	Vectors of lower and upper limits for generators.
$\hat{\mathbf{g}}$	Vector of capacities for generators.
$\bar{g}_i$	Upper limit for generator i .
$\hat{g}_i$	Capacity of generator i .
$\mathbf{K}_0$	Matrix of power transfer distribution factors.
$\mathbf{K}_1, \mathbf{k}_2$	Preprocessed structures for positive flow limits.
$\mathbf{K}_3, \mathbf{k}_4$	Preprocessed structures for negative flow limits.
M	Big-M.
lb, ub	Lower/upper bound for the decomposition method.
p	Parameter for primal solution approach.
$\bar{\mathbf{r}}$	Vector of primary response limits of generators.
$\bar{r}_i$	Primary response limit of generator i , given by $\gamma_i \hat{g}_i$.
S	Angle-to-flow matrix.
z	Objective value of the master problem for the column-and-constraint-generation algorithm.
z_p	Objective value z when using the parameter p .

Nominal-state-related decision variables and vectors

$\theta, \mathbf{f}, \mathbf{g}$	Phase angles, line flows, and nominal generation.
g_i	Generation of generator i in nominal state.

Contingency-state-related decision variables and vectors

θ_s	Vector of phase angles under contingency state s .
μ_s	Vector of dual variables associated with nodal load balance constraint under contingency state s .
$\mathbf{f}_s$	Vector for line flows under contingency state s .
$\mathbf{g}_s$	Vector for generation under contingency state s .
$\mathbf{g}_s'$	Provisional vector for $\mathbf{g}_s$.
$g_{s,i}$	Generation of generator i under contingency state s .
$g_{s,i}'$	Provisional variable for $g_{s,i}$.
n_s	Global signal under contingency state s .
$\mathbf{u}_s^+, \mathbf{u}_s^-$	Vectors of slack variables for line capacities.
$\mathbf{v}_s^+, \mathbf{v}_s^-$	Vectors of slack variables for nodal demand balance.
$\mathbf{x}_s$	Binary vector indicating whether generators reached $\bar{\mathbf{g}}$ under contingency state s .
$x_{s,i}$	Binary variable indicating whether generator i reached $\bar{g}_i$ under contingency state s .

primary response. These variables are multiplied by parameters that represent the sensitivity of generators to frequency changes. Such frequency regulation parameters are related to *droop coefficients*. We refer the interested reader to the discussions about droop coefficients in [6] (and references therein) where, particularly, the authors argue that the co-optimization of the droop coefficient and the SCOPF might save on costs. In this work, we have also opted for the dc power flow approximation with a linear model for primary response. In summary, for each contingency state, we have substituted the single variable representing frequency drop adopted in [6] and [7] with a single global signal to generators (also a variable).

The SCOPF problem featuring automatic primary response of generators is a mixed-integer linear program (MILP) even under the dc relaxation. This is because the constraints for the automatic response of generators may lead to power outputs above generator limits [16]. To remedy this, we require binary variables for each generator and for each

contingency state to determine whether a generator is producing according to the constraints for automatic response or at its limit. The size of the problem is generally large. It is proportional to the number of contingencies since we are required to represent the network and the power flow variables for each post-contingency state.

The Benders' decomposition approach, which has often been applied to solve energy planning problems [3,5,18,19], is a natural candidate to tackle the preventive SCOPF problem. Generally, in Benders' approaches, the extensive formulation is recast into a master problem and subproblems. The master problem for power systems applications usually solves the nominal dispatch, and the subproblems represent the redispatch or corrective actions under contingencies and/or uncertain scenarios. An iterative procedure that involves solving the master problem and subproblems is performed. During this process, Benders' cuts for the violated subproblems are added into the master problem. The process continues until all subproblems are feasible. A valuable

review on the Benders' decomposition method can be found in [20].

Unfortunately, preventive SCOPF imposes challenges for the application of traditional Benders' decomposition since the subproblems are nonconvex. The constraints that enforce the primary response of synchronized generators contain binary variables. Despite such challenges, a solution method inspired by [19] considering nonconvex subproblems was provided in [6]. However, the optimality for this method is not guaranteed. An alternative that ensures optimality is to recast the master problem to include the constraints for the primary response. This modification, however, does not scale well since the number of binary variables increases quadratically with the number of synchronized generators (assuming the $N - 1$ security criteria for generators).

In order to remedy the aforementioned limitations, we have devised an exact and scalable algorithm to tackle the preventive SCOPF problem with primary response. The focus of this work is on the computational and practical aspects of the solution methodology. The proposed decomposition scheme differs significantly from previously proposed solution methods. The outline of the method is as follows.

In the master problem we consider a nominal optimal power flow problem that accounts for valid constraints for each contingency state. We initially disregard the network for contingency states and the disjunctions (binary variables) in the master problem. This approach alleviates the computational burden required for solving the master problem. During the iterative process only a small subset of the disjunctions and network constraints are introduced to the master problem by a column-and-constraint generation algorithm (CCGA) [21]. In the proposed decomposition approach, the only optimization problem that is solved is the master problem. This is possible since we use: i) preprocessed structures based on the power transfer distribution factors (PTDF) that are useful both as feasibility checkers and as dedicated cuts in the post-contingency states, and ii) a numerical procedure that determines the post-contingency variables based solely on the nominal generation. The aforementioned preprocessed structures allow us to monitor the critical congested areas of the system. As it is necessary, these structures are transformed into constraints (that differ from Benders' cuts) that are added to the master problem. These cuts represent the network for the post-contingency states. Likewise, as it is necessary, we introduce the disjunctions (binary variables) representing the primary response model for a few contingency states into the master problem. We also propose a method to find high-quality primal solutions and a procedure that monitors the upper and lower bounds for the method.

In summary, the main contribution of this work is a decomposition approach that combines a column-and-constraint-generation algorithm with numerical methods to determine exact solutions to the SCOPF problem in such a way that the nonconvex subproblems do not need to be solved directly. We demonstrate that this approach is possible due to the presence of valid post-contingency constraints in the master problem and due to the preprocessed structures derived from a PTDF-based formulation.

The rest of this paper is organized as follows: In Section II, the SCOPF model is introduced. The solution methodology is presented in Section III. Numerical experiments are reported in Section IV. Finally, this paper is concluded in Section V.

2. SCOPF with primary response formulation

We assume a generic framework where a bid-based market for energy and reserve and/or unit commitment (UC) procedures have taken place hours before (or in the day before). We assume that, at the time the SCOPF is solved, the operator has precise forecasts for the few-hours-ahead non-dispatchable generation, and nodal net loads. For notational conciseness we assume that all generators are synchronized.

2.1. Power flow constraints

We use the following dc power flow constraints:

$$\mathbf{A}\mathbf{f} + \mathbf{B}\mathbf{g} = \mathbf{d} \quad (1)$$

$$\mathbf{f} = \mathbf{S}\boldsymbol{\theta} \quad (2)$$

$$-\bar{\mathbf{f}} \leq \mathbf{f} \leq \bar{\mathbf{f}} \quad (3)$$

$$\underline{\mathbf{g}} \leq \mathbf{g} \leq \bar{\mathbf{g}} \quad (4)$$

$$\mathbf{A}\mathbf{f}_s + \mathbf{B}\mathbf{g}_s = \mathbf{d}, \quad \forall s \in \mathcal{S} \quad (5)$$

$$\mathbf{f}_s = \mathbf{S}\boldsymbol{\theta}_s, \quad \forall s \in \mathcal{S} \quad (6)$$

$$-\bar{\mathbf{f}} \leq \mathbf{f}_s \leq \bar{\mathbf{f}}, \quad \forall s \in \mathcal{S} \quad (7)$$

$$\mathbf{g}_s \leq \bar{\mathbf{g}}, \quad \forall s \in \mathcal{S}. \quad (8)$$

Constraints (1)–(4) model the power flow in the nominal state. Expression (1) represents nodal power balance under a dc power flow model, while Kirchhoff's second law is accounted for in (2). Transmission line limits and generator limits are enforced by (3) and (4), respectively. In (4), we allow generation bounds $\underline{\mathbf{g}}$ and $\bar{\mathbf{g}}$ to be different from the minimum and maximum ($\bar{\mathbf{g}}$) set points for the generators due to commitment and/or operational constraints. Analogously to block (1)–(4), the set of constraints (5)–(8) model the power flow under each contingency state s .

2.2. Primary response model

The primary response under contingency state s is modeled by a global signal¹ n_s sent to all synchronized generators. This approach differs from those of [6] and [7], where variables representing frequency drops under contingency states are considered. We assume that the response of generator i is proportional to its capacity $\hat{g}_i$ and also to a predefined coefficient γ_i that is associated with the droop coefficient. Hence, under s , the automatic response of synchronized generator i is given by $g_{s,i} - g_i = n_s \gamma_i \hat{g}_i$, with the additional constraints that $g_{s,i} \leq \bar{g}_i$. Mathematically, we have

$$g_{s,i} = \min \left\{ g_i + n_s \gamma_i \hat{g}_i, \bar{g}_i \right\}, \quad \forall i \in \mathcal{G}, \forall s \in \mathcal{S}, i \neq s \quad (9)$$

$$g_{s,s} = 0, \quad \forall s \in \mathcal{S}. \quad (10)$$

By using traditional MILP modeling techniques to rewrite (9)–(10), we obtain

$$\left| g_{s,i} - g_i - n_s \gamma_i \hat{g}_i \right| \leq \bar{g}_i (1 - x_{s,i}), \quad \forall i \in \mathcal{G}, \forall s \in \mathcal{S}, i \neq s \quad (11)$$

$$g_i + n_s \gamma_i \hat{g}_i \geq \bar{g}_i (1 - x_{s,i}), \quad \forall i \in \mathcal{G}, \forall s \in \mathcal{S}, i \neq s \quad (12)$$

$$g_{s,i} \geq \bar{g}_i (1 - x_{s,i}), \quad \forall i \in \mathcal{G}, \quad \forall s \in \mathcal{S}, i \neq s \quad (13)$$

$$n_s \in [0, 1], \quad \forall s \in \mathcal{S} \quad (14)$$

$$x_{s,i} \in \{0, 1\}, \quad \forall i \in \mathcal{G}, \forall s \in \mathcal{S} \quad (15)$$

$$g_{s,s} = 0, \quad \forall s \in \mathcal{S}. \quad (16)$$

¹ An alternative interpretation is that n_s mimics the systems response which is required for adjusting the power (or frequency) imbalance. The physical interpretation for the limitation of 1 to n_s is that γ_i multiplied by capacity is the maximum deliverable reserve or maximum ramp for a given time frame.

2.3. Extensive formulation for the SCOPF problem

The SCOPF problem is modeled as a MILP, where we minimize the cost of nominal generation in the objective function subject to constraints (1)–(8) and (11)–(16). For conciseness, we define $\mathcal{X} = [\mathbf{g}, \mathbf{f}, \boldsymbol{\theta}]$, $\mathcal{X}_s = [\mathbf{g}_s, \mathbf{f}_s, \boldsymbol{\theta}_s]$, and $\mathcal{Y}_s = [\mathbf{g}_s, \mathbf{x}_s, \mathbf{n}_s]$. Let $\mathcal{Y}_s \in \mathcal{F}_s$ denote the disjunctions related to (11)–(16) while $\mathcal{X} \in \mathcal{E}$ and $\mathcal{X}_s \in \mathcal{E}_s$ denote the power flow constraints in nominal (1)–(4) and contingency states (5)–(8) respectively. The extensive formulation for the SCOPF problem, labeled as EF method, is as follows

$$\min_{\mathcal{X}, [\mathcal{X}_s, \mathbf{x}_s, \mathbf{n}_s]_{s \in \mathcal{S}}} \mathbf{c}^\top \mathbf{g} \quad (17)$$

$$\text{s.t.: } \mathcal{X} \in \mathcal{E} \quad (18)$$

$$\mathcal{X}_s \in \mathcal{E}_s, \quad \forall s \in \mathcal{S} \quad (19)$$

$$\mathcal{Y}_s \in \mathcal{F}_s, \quad \forall s \in \mathcal{S}. \quad (20)$$

3. Solution methodology

In this work we focus on methods that guarantee optimality for the SCOPF problem. A Benders' decomposition with valid post-contingency constraints in the master problem is presented in Section 3.1. Preprocessed structures for feasibility checking and cut generation are presented in Section 3.2, while a useful binary search is introduced in Section 3.3. The CCGA is described in Section 3.4. A method for finding high-quality primal solutions is presented in Section 3.5.

3.1. Modified Benders' decomposition

The intuitive Benders' decomposition approach for (17)–(20) is to define the master problem as the nominal schedule, associated with (17)–(18), and the subproblems as the separable feasibility recourse problems enforcing (19)–(20) for each $s \in \mathcal{S}$. Unfortunately, this approach introduces nonconvexities to the subproblems [6], and thus, does not guarantee optimality.

In order to ensure the convexity of the subproblems, and thus optimality for the method, we define the subproblems as feasibility-like problems for the constraints in (19). As part of the modification, we also add the following valid post-contingency constraints to the master problem:

$$\mathbf{e}^\top \mathbf{g}_s = \mathbf{e}^\top \mathbf{d}, \forall s \in \mathcal{S}. \quad (21)$$

The purpose of (21) is to strengthen the master problem with the necessary post-contingency condition that the total generation and the total load are equal. By also enforcing (8) we guarantee that post-contingency generation is within bounds. We define the master problem² as

$$\min_{\mathcal{X}, [\mathbf{g}_s, \mathbf{x}_s, \mathbf{n}_s]_{s \in \mathcal{S}}} \mathbf{c}^\top \mathbf{g} \quad (22)$$

$$\text{s.t.: } (18) \quad (23)$$

$$(8), (20), (21), \quad \forall s \in \mathcal{S}. \quad (24)$$

The subproblem for each $s \in \mathcal{S}$, where $\mathbf{g}_s^{(*)}$ is the solution determined in (22)–(24), is then defined as

$$\min_{\mathbf{v}_s^+, \mathbf{v}_s^-, \mathbf{f}_s, \mathbf{g}_s, \boldsymbol{\theta}} \mathbf{e}^\top (\mathbf{v}_s^+ + \mathbf{v}_s^-) \quad (25)$$

$$\text{s.t.: } (6), (7) \quad (26)$$

$$\mathbf{g}_s = \mathbf{g}_s^{(*)} : \boldsymbol{\mu}_s \quad (27)$$

$$\mathbf{A}\mathbf{f}_s + \mathbf{B}\mathbf{g}_s = \mathbf{d} + \mathbf{v}_s^+ - \mathbf{v}_s^-. \quad (28)$$

A feasibility Benders' cut is then added to the master problem at each iteration, for each s that is not feasible; that is, $\forall s \in \mathcal{S}$ such that $\mathbf{e}^\top (\mathbf{v}_s^{+(*)} + \mathbf{v}_s^{-(*)}) > \epsilon$, where ϵ is a tolerance level for the net load imbalance. The Benders' cut for s is as follows: $\mathbf{e}^\top (\mathbf{v}_s^{+(*)} + \mathbf{v}_s^{-(*)}) + \boldsymbol{\mu}^\top (\mathbf{g}_s - \mathbf{g}_s^{(*)}) \leq 0$. We label this approach as the BD method.

3.2. Precomputation of dedicated cuts

In this subsection, an alternative method named BDDC is introduced. Unlike the BD method that involves subproblems that generate Benders' cuts, the BDDC method uses preprocessed structures as feasibility checkers and to generate cuts. These structures, that are also applied in the CCGA of Section 3.4, are based on the PTDF formulation for dc power flow.

In the BDCC method, we have the same master problem (22)–(24) as the BD method. Thus, $\mathbf{g}_s^{(*)}$ is determined in (22)–(24), where the primary response constraints (20) and the post-contingency generation constraints (8) and (21) are enforced.

The aforementioned preprocessed structures are constructed directly from the PTDF-based formulation for the dc power flow. This formulation, for contingency state s , is as follows:

$$\min_{\mathbf{u}_s^-, \mathbf{u}_s^+} 0 \quad (29)$$

$$\text{s.t.: } -\bar{\mathbf{f}} + \mathbf{u}_s^- = \mathbf{K}_0 (\mathbf{d} - \mathbf{B}\mathbf{g}_s^{(*)}) = \bar{\mathbf{f}} - \mathbf{u}_s^+ \quad (30)$$

$$\mathbf{u}_s^-, \mathbf{u}_s^+ \geq 0. \quad (31)$$

In (30), $\mathbf{g}_s^{(*)}$ is a solution determined by (22)–(24) and $\mathbf{K}_0$ is the PTDF matrix. A similar description for (30) is presented in [22]. We highlight that (8) and (21) are enforced in the master problem and therefore are not necessary in (29)–(31). As opposed to the subproblems of the BD method we do not allow nodal imbalance in (29)–(31). Thus, a given master problem solution $\mathbf{g}_s^{(*)}$ is feasible under contingency state s if there is a feasible solution $\mathbf{u}_s^-, \mathbf{u}_s^+$ for (29)–(31).

In this work we assume that we have more lines $|\mathcal{L}|$ than buses $|\mathcal{N}|$ and that there are no isolated buses. Under these assumptions and because we enforce (8) and (21) in the master problem, we do not need to solve (29)–(31) to check for feasibility in post-contingency states or to obtain cuts. Manipulating (30), we derive the preprocessed structures:

$$(\mathbf{u}_s^+) : \mathbf{K}_1 \mathbf{g}_s + \mathbf{k}_2 \geq 0 \quad (32)$$

$$(\mathbf{u}_s^-) : \mathbf{K}_3 \mathbf{g}_s + \mathbf{k}_4 \geq 0. \quad (33)$$

Interestingly, the matrices $\mathbf{K}_1$, $\mathbf{K}_3$ and the vectors $\mathbf{k}_2$, $\mathbf{k}_4$ can be efficiently precomputed³ and are the same for all $s \in \mathcal{S}$. Another feature is that (32) and (33) are directly associated with the transmission lines of the system, relating either to the positive (32) or negative (33) limits. By inspecting solutions $\mathbf{g}_s$ on (32)–(33) it is possible to verify the existence of violated lines and the intensity of violations (in MW) under each contingency state.

The algorithm for the BDDC method involves adding rows of (32)

² There are two main differences with respect to the EF: Line constraints (6)–(7) are not included, and constraint (5) is substituted by (21).

³ The incorporation of line outages as part of the set of credible contingencies would require the precomputation of different structures for $\mathbf{K}_1$, $\mathbf{K}_3$, $\mathbf{k}_2$, and $\mathbf{k}_4$. This would lead to a linear increase in the size of the precomputed data for the problem (as many as the line contingencies). This would not present a significant computational obstacle.

and (33) as lazy constraints to problem (22)–(24). We do not require that an optimal solution $\mathbf{g}_s$ is found. Whenever a feasible (suboptimal) integer solution is determined by the solver, as a subroutine, we check the feasibility of $\mathbf{g}_s$ using (32) and (33). We define $\alpha_{s,i}$ as the violation of line l , for contingency state s and α as the largest violation among all transmission lines for all s . We then add to (22)–(24) as lazy constraints the rows of both (32) and (33) corresponding to violated lines such that $\alpha_{s,l} > \alpha/\beta_1$, where β_1 is a parameter. We stop the algorithm when α is smaller than a defined tolerance ϵ . This procedure converges in finite steps since adding all rows of (32) and (33), for every $s \in \mathcal{S}$, to (22)–(24) results in a problem which is equivalent to (17)–(20).

Unfortunately, the application of the BDDC method alone is not scalable since its master problem contains all the binary variables. Before proceeding to the proposed CCGA (Section 3.4), we introduce next a useful binary search procedure.

3.3. Numerical procedure for calculating n_s and $\mathbf{g}_s$

Under the primary response model, the post-contingency generation for s ; that is, $\mathbf{g}_s$, is defined by the combination of the nominal schedule $\mathbf{g}^{(*)}$ and the global signal $n_s^{(*)}$. Namely, given $\mathbf{g}^{(*)}$ and $n_s^{(*)}$, it is straightforward to compute $\mathbf{g}_s^{(*)}$ by applying the relations in (11)–(16).

Interestingly, $n_s^{(*)}$ can also be calculated from $\mathbf{g}^{(*)}$. This is achieved by a binary search for n_s for each $s \in \mathcal{S}$. The binary search is possible in this case since, for a fixed $\mathbf{g}^{(*)}$, each component of $\mathbf{g}_s$ is monotone with respect to n_s . Thus, despite the presence of the disjunctions, we only need to find the correct $n_s^{(*)}$ that results in a vector $\mathbf{g}_s^{(*)}$ that satisfies the total net demand. Given the fast convergence of the binary search, the tolerance ϵ can be set to very small values. This procedure is described in Algorithm 1.

3.4. Column-and-Constraint-Generation algorithm

We define the master problem for the CCGA as follows

$$z = \min_{\mathcal{S}, \{n_s\}_{s \in \mathcal{S}}, \{\mathbf{g}_s\}_{s \in \mathcal{S}}} \mathbf{c}^\top \mathbf{g} \quad (34)$$

$$\text{s.t.:} \quad (18) \quad (35)$$

$$\mathbf{g}'_s - \mathbf{g} \leq \bar{\mathbf{r}}, \quad \forall s \in \mathcal{S} \quad (36)$$

$$(8), (16), (21), \quad \forall s \in \mathcal{S} \quad (37)$$

$$(20), \quad \forall s \in \mathcal{S}. \quad (38)$$

Note that, as opposed to the BDDC method, in (38) we define a different set of contingency states $\mathcal{S}$ for the disjunctive constraints (starting with $\mathcal{S} = \emptyset$). We also abuse notation by using $\mathbf{g}'_s$ as a provisional variable for the post-contingency generation replacing $\mathbf{g}_s$ in (36)

Algorithm 1 Binary Search($s, \varepsilon, \mathbf{g}^{(*)}$)

```

1: Initialization:  $j \leftarrow 0$  and  $n_s^{(0)} \leftarrow 0.5$ .
2: for  $\forall i \in \mathcal{G}, i \neq s$  do
3:   if  $g_i^{(*)} + n_s^{(j)} \gamma_i \hat{g}_i \geq \bar{g}_i$  then:  $g_{s,i}^{(j)} \leftarrow \bar{g}_i$ 
4:   else:  $g_{s,i}^{(j)} \leftarrow g_i^{(*)} + n_s^{(j)} \gamma_i \hat{g}_i$ 
5:   end if.
6: end for
7:  $g_{s,s}^{(j)} \leftarrow 0$ ;  $e_s \leftarrow (\mathbf{e}^\top \mathbf{g}_s^{(j)} - \mathbf{e}^\top \mathbf{d})$ .
8: if  $|e_s| \leq \varepsilon$  then:  $n_s^{(*)} \leftarrow n_s^{(j)}$ ,  $g_{s,i}^{(*)} \leftarrow g_{s,i}^{(j)}, \forall i \in \mathcal{G}$  and BREAK
9: else if  $e_s > 0$  then:  $n_s^{(j+1)} \leftarrow n_s^{(j)}/2$ 
10: else:  $n_s^{(j+1)} \leftarrow (1 + n_s^{(j)})/2$ 
11: end if.
12:  $j \leftarrow j + 1$ ; Go to step 2.

```

and (37). We performed this substitution to make explicit that $\mathbf{g}_s$ is not determined in (34)–(38) for the entire iterative process. The determination of $\mathbf{g}_s$ in (34)–(38) would only be possible in the presence of the disjunctive constraints (20) for s . These disjunctive constraints are not initially present in (34)–(38) for computational purposes. In fact, the determination of $\mathbf{g}_s$ is performed by Algorithm 1, which requires only $\mathbf{g}$ as an input. The purpose of $\mathbf{g}'_s$ in (36) and (37) is to guarantee that $\mathbf{g}$ is determined in such a way that Algorithm 1 is capable of enforcing the primary response compatibility to $\mathbf{g}_s$, while meeting the global demand. That is, for each s , $|\mathbf{e}^\top \mathbf{g}_s - \mathbf{e}^\top \mathbf{d}| \leq \epsilon$.

In order to verify the above claim, note that by (36) and (8), $g'_{s,i} \leq \min\{\bar{g}_i, g_i + \gamma_i \hat{g}_i\}$ for each i and s , where $g'_{s,i}$ is the i -th element of $\mathbf{g}'_s$. Defining $n_s = 0$ in Algorithm 1 implies $\mathbf{g}_s = \mathbf{g}$, except for $g_{s,s} = 0$. If instead we set $n_s = 1$ then $g_{s,i} = \min\{\bar{g}_i, g_i + \gamma_i \hat{g}_i\} \geq g'_{s,i}$ for each i and s , with $i \neq s$. For $i = s$, we have that $g'_{s,s} = g_{s,s} = 0$. Thus, since $\mathbf{g}'_s$ meets the global demand, it is always the case that $\mathbf{e}^\top \mathbf{g}_s \geq \mathbf{e}^\top \mathbf{g}'_s = \mathbf{e}^\top \mathbf{d}$ by choosing $n_s = 1$. By the monotonicity and continuity of $g_{s,i}$ with respect to n_s for a given $\mathbf{g}_s$, there is a value $n_s^{(*)}$ that results in $\mathbf{g}_s^{(*)}$ that satisfies the global demand and preserves the primary response model.

At each iteration j of the CCGA, we solve the master problem (34)–(38) to obtain $\mathbf{g}^{(j)}$ and $z^{(j)}$. Then, for each $s \in \mathcal{S}$, we perform the binary search (Algorithm 1) to define $\mathbf{g}_s^{(j)}$ according to the primary response model. Next, for all s , we check feasibility of the solutions $\mathbf{g}_s^{(j)}$ using (32) and (33). We use $\alpha_{s,i}^{(j)}$ to define the violation of each line l , for each contingency state s and we use $\alpha^{(j)}$ as the largest violation among all transmission lines for all s . We identify the contingency state $s^{(j)}$ that contains the most violated line. If $s^{(j)} \in \mathcal{S}$ we skip the rest of this step. Otherwise we set $\mathcal{S} = \mathcal{S} \cup \{s^{(j)}\}$ which means including the disjunctions (20) for $s^{(j)}$ into (34)–(38).

We also add to (34)–(38) the rows of both (32) and (33) corresponding to violated lines using two criteria: i) For the post-contingency states $s \in \mathcal{S}$ we include the lines where $\alpha_{s,l}^{(j)} > \alpha^{(j)}/\beta_1$. ii) For the post-contingency states $s \notin \mathcal{S}$ we include the lines where $\alpha_{s,l}^{(j)} > \alpha^{(j)}/\beta_2$. The objective of this step is to enforce the network constraints for critical lines in post-contingency states. Typically, $\beta_1 > \beta_2$. We are stricter with

Algorithm 2 CCGA($\varepsilon, \epsilon, \beta_1, \beta_2$)

```

1: Initialization:  $j \leftarrow 0, \mathcal{S} \leftarrow \emptyset$ .
2: Solve: (34)–(38) to obtain  $\mathbf{g}^{(j)}$ 
3: for  $\forall s \in \mathcal{S}$  do
4:   Use Algorithm 1 to obtain:  $z^{(j)}, \mathbf{g}_s^{(j)}, n_s^{(j)}$ 
5:   for  $\forall l \in \mathcal{L}$  do
6:     Compute  $\alpha_{s,l}^{(j)}$  as the maximum violation of (32) and (33)
7:   end for
8: end for
9: Compute  $\alpha^{(j)}$  and identify the state  $s^{(j)}$  of the most violated line
10: if  $s^{(j)} \notin \mathcal{S}$  then set  $\mathcal{S} = \mathcal{S} \cup \{s^{(j)}\}$  (add (20) for  $s^{(j)}$  to (34)–(38))
11: end if
12: for  $\forall s \in \mathcal{S}$  do
13:   if  $s \in \mathcal{S}$  then  $\beta \leftarrow \beta_1$ 
14:   else  $\beta \leftarrow \beta_2$ 
15:   end if
16:   for  $\forall l \in \mathcal{L}$  do
17:     if  $\alpha_{s,l}^{(j)} > \alpha^{(j)}/\beta$  AND if cuts (32)–(33) for the pair  $(l,s)$  are not yet included then
18:       Add (32)–(33) for contingency state  $s$ , line  $l$ 
19:     end if
20:   end for
21: end for
22: if  $\alpha^{(j)} < \epsilon$  then  $z^{(*)} \leftarrow z^{(j)}, \mathbf{g}^{(*)} \leftarrow \mathbf{g}^{(j)}$ ; BREAK
23: else  $j \leftarrow j + 1$ ; Go to step 2.
24: end if

```

the states $s \in \mathbb{S}$ since defining tight parameters for contingency states without corresponding disjunctions may lead to the inclusion of many constraints at a time. A user defined tolerance ϵ (in MW) for maximum line violation is used to stop the iterative process. The CCGA is described in [Algorithm 2](#).

3.5. Finding high-Quality primal solutions and monitoring the optimality gap using the CCGA

Because very large cases might still impose computational challenges, we propose a procedure for finding feasible primal solutions. This procedure restricts the disjunctions in (20) to a subset of synchronized generators $\mathcal{H} \subset \mathcal{G}$. The generators in $\mathcal{G} \setminus \mathcal{H}$ respond with $g_{s,i} - g_i = n_s \gamma_i \hat{g}_i$; that is, we define $x_{s,i} = 1, \forall s \in \mathcal{S}, \forall i \in \mathcal{G} \setminus \mathcal{H}, i \neq s$.

The following criterion is used for defining $\mathcal{H}$. We rank the synchronized generators according to a “cost/limit” index ($c_i / \bar{g}_i$) and define $\mathcal{H}$ as the $p\%$ of generators with lowest ranks, where p is a parameter. The objective value of the problem using this approach is denoted as $z_p^{(*)}$. Note that $z_{100}^{(*)} = z^{(*)}$.

This approach reduces the number of binary variables and thus the complexity of the problem. It is then a tool for finding upper bounds for (17)–(20). If we apply [Algorithm 2](#) using the proposed primal method; i.e., setting $x_{s,i} = 1, \forall s \in \mathcal{S}, \forall i \in \mathcal{G} \setminus \mathcal{H}$ in (34)–(38) we obtain $ub = z_p^{(*)}$ as a valid upper bound. If the problem is infeasible for p then $z_p^{(*)} \leftarrow \text{inf}$.

Note, however, that a procedure that monitors the optimality gap is still required. A lower bound for (17)–(20) is not obtained for $p < 100$. Conversely, solving for $p = 100$ generates upper bounds only after a feasible solution is found. This typically occurs in the later iterations when the tolerance ϵ for all lines in every contingency state is achieved.

We propose a simple strategy that monitors the bounds. Note that solving the SCOPF with the CCGA for low values of p tends to be faster than for high values of p . Thus, we start $p = 0$ and increase it sequentially. The solution for each $p < 100$ provides an upper bound for (17)–(20). As a parallel procedure, we solve for $p = 100$ to obtain valid lower bounds. Namely, for each iteration j of [Algorithm 2](#) for $p = 100$, a valid lower bound is defined as the best bound provided by the solver. This procedure monitors the gap efficiently.

4. Computational experiments

We compared the proposed CCGA with two solution methods: EF and BDDC described in [Sections 2.3](#) and [3.2](#).

We performed simulations for various values of γ , β_1 , and β_2 . Our results indicate that varying the parameters may impact the performance of the CCGA. However, the dominance of the CCGA over other methods (EF and BDDC) was a constant, despite the parameterization. We have reported results for $\beta_1 = 5$, $\beta_2 = 1.2$, and $\gamma_i = 0.05$ for all $i \in \mathcal{G}$.

We used Gurobi 8.1.1 under the modeling package JuMP 0.18.5 for Julia Language 0.6.4 on a Xeon E5-2680 processor at 2.5 GHz and 128 GB of RAM. We set the optimality gap of Gurobi to 0.5% for the EF method and BDDC method as well as for the master problem of the CCGA. The maximum line violation was set to $\epsilon = 0.05$ MW and the precision of [Algorithm 1](#) to $\epsilon = 10^{-10}$ MW.

The data are based on modified versions of the benchmark systems presented in [\[23\]](#). The size of the instances for the EF method, after Gurobi’s presolve, are reported in [Table 1](#).

4.1. Solution method comparison

[Table 2](#) provides the computational times for selected methods and the number of iterations for the CCGA method in parentheses. The CCGA dominates the other methods, which were only able to solve the 118 IEEE case within a reasonable time limit. For this instance, the CCGA required less than one third of the time of the BDDC method and less

Table 1

Instance size for the EF method after Gurobi’s presolve

System	Continuous Variables	Binary Variables	Linear Constraints
118 IEEE	10,604	2,862	19,137
1354 PEGASE	323,571	63,455	513,677
1888 RTE	387,979	79,032	624,780
1951 RTE	563,273	149,370	1,010,994
2848 RTE	1,104,192	276,822	1,934,115
2868 RTE	1,284,568	348,036	2,328,081
6468 RTE	5,067,009	1,563,640	9,756,668

Table 2

Comparative CPU times (s) and number of iterations.

System	Solution Method		
	EF	BDDC	CCGA
118 IEEE	46.2	27.6	8.0 (4)
1354 PEGASE	T	T	138.0 (12)
1888 RTE	T	2899.0	14.0 (4)
1951 RTE	T	T	16.2 (4)
2848 RTE	T	T	26.6 (4)
2868 RTE	T	T	31.0 (3)
6468 RTE	T	T	7881.6 (18)

T - Time limit of 4 hours exceeded.

than one fifth of the time of the EF method. The only other instance that the BDDC method was able to solve in less than 4 hours was the 1888 RTE. The CCGA was more than 200 times faster for this instance. Interestingly, the number of iterations required by the CCGA is generally small, implying that the CCGA solved far less complex MILPs than the other methods. The only instance that posed difficulties for the CCGA was the 6468 RTE that contains more than 6000 buses, 1200 generators, and 9000 transmission lines. Nevertheless, a solution for the optimality gap of 0.5% was achieved in less than 3 hours.

Interestingly, as reported next, it is possible to determine high-quality solutions in competitive computational times for large systems by applying the primal method of [Section 3.5](#).

4.2. Finding primal solutions and determining bounds

The method of [Section 3.5](#) for defining primal solutions was applied for the 6468 RTE and 1354 PEGASE systems. CCGA was used to solve the SCOPF problem for different values of p to an optimality gap of 0.5%. The results are summarized in [Tables 3](#) and [4](#). Columns 1 and 2 present the cost and the relative cost gap for each p with respect to the cost

Table 3

Primal Approach for the 6468 RTE System.

p	Cost (10^3 \$)	Cost Gap (%)	Time (s)	Iter. (#)
100	1624.8	0.00	7881.6	18
10	1625.8	0.02	813.5	17
0	1628.7	0.25	481.2	16

Table 4

Primal Approach for the 1354 PEGASE System.

p	Cost (10^3 \$)	Cost Gap (%)	Time (s)	Iter. (#)
100	1190.6	0.00	138.0	12
50	1192.6	0.17	64.7	11
10	1195.4	0.40	21.1	11
0	1208.3	1.49	9.4	10

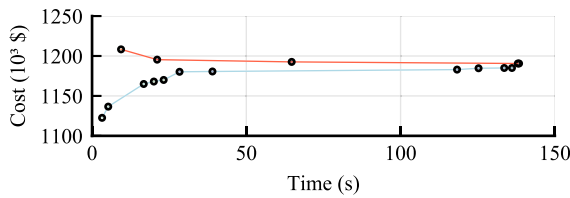


Fig. 1. Bounds for the 1354 PEGASE system.

achieved by $p = 100$. Columns 3 and 4 report the required computational time and number of iterations.

For the 6468 RTE system (Table 3) the result is quasi-optimal even for $p = 0$. For the 1354 PEGASE system (Table 4) the solution for $p = 0$ is already competitive, and required 9.4 seconds only. By increasing the complexity of the problem to $p = 10$, the cost gap is reduced by more than 1% for a reasonable solution time of 21.1 seconds. For $p = 50$, the CCGA required 64.7 seconds to converge, achieving a negligible cost gap of 0.17.

Despite the good results for small values of p , the cost gap is not observable before solving for $p = 100$. Thus, we adopted the strategy of Section 3.5 for obtaining bounds. We used multi-threading to solve problems in parallel. In the first thread we solved a sequence of problems for increasing values of p , starting with $p = 0$. We have stored the costs and times for the solutions of each p . In the second thread we solved for $p = 100$ and recorded solving time and the best bound of each iteration provided by Gurobi. A convergence plot from applying this method to the 1354 PEGASE system is illustrated in Fig. 1. The proposed strategy yields the true optimality gap and is a useful decision-making tool for system operators.

5. Conclusion

We presented an exact and scalable column-and-constraint-generation algorithm for the SCOPF problem with primary response of generators. Under the proposed framework, we add the disjunctions as necessary in an iterative process that does not involve subproblems. This is possible by a scheme that involves a master problem with valid post-contingency constraints, preprocessed structures that serve both as feasibility checkers and delayed cuts, and a numerical procedure that reduces the complexity of the master problem by exogenously calculating the nonconvex primary response. We also proposed a procedure for finding high-quality primal solutions that helps monitor the bounds for the method. As shown by the computational experiments, this approach scales to large instances of the SCOPF problem with primary response.

In future works we will generalize these techniques. As a first step, convexifications of ac power flow, such as those studied in the recent works of Coffrin and Van Hentenryck [12,13], will be used to model the system's response after contingencies.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] O. Alsac, B. Stott, Optimal load flow with steady-state security, *IEEE Trans. Power App. Syst.* PAS-93 (3) (1974) 745–751.
- [2] F. Bouffard, F.D. Galiana, J.M. Arroyo, Umbrella contingencies in security-constrained optimal power flow, 15th Power syst. Comput. Conf. PSCC 5 (2005).
- [3] Y. Li, J.D. McCalley, Decomposed SCOPF for improving efficiency, *IEEE Trans. Power Syst.* 24 (1) (2009) 494–495.
- [4] F. Capitanescu, J.M. Ramos, P. Panciatici, D. Kirschen, A.M. Marcolini, L. Platbrood, L. Wehenkel, State-of-the-art, challenges, and future trends in security constrained optimal power flow, *Elect. Power Syst. Res.* 81 (8) (2011) 1731–1741.
- [5] Q. Wang, J.D. McCalley, T. Zheng, E. Litvinov, Solving corrective risk-based security-constrained optimal power flow with lagrangian relaxation and benders decomposition, *Int. J. Elec. Power* 75 (2016) 255–264.
- [6] Y. Dvorkin, P. Henneaux, D.S. Kirschen, H. Pandžić, Optimizing primary response in preventive security-constrained optimal power flow, *IEEE Syst. J.* 12 (1) (2016) 414–423.
- [7] M. Velay, M. Vinyals, Y. Besanger, N. Retiere, Fully distributed security constrained optimal power flow with primary frequency control, *Int. J. Elec. Power* 110 (2019) 536–547.
- [8] Z. Zhou, T. Levin, G. Conzelmann, Survey of U.S. ancillary services markets, argonne national lab.(ANL), argonne, IL (united states), Tech. Rep. (2016).
- [9] L. Platbrood, F. Capitanescu, C. Merckx, H. Crisui, L. Wehenkel, A generic approach for solving nonlinear-discrete security-constrained optimal power flow problems in large-scale systems, *IEEE Trans. Power Syst.* 29 (3) (2013) 1194–1203.
- [10] F. Capitanescu, Critical review of recent advances and further developments needed in ac optimal power flow, *Electr. Power Syst. Res.* 136 (2016) 57–68.
- [11] F. Li, R. Bo, Dcopf-based lmp simulation: algorithm, comparison with acopf, and sensitivity, *IEEE Trans. Power Syst.* 22 (4) (2007) 1475–1485.
- [12] C. Coffrin, P.V. Hentenryck, A linear-programming approximation of ac power flows, *INFORMS J. Comput.* 26 (4) (2014) 718–734.
- [13] C. Coffrin, H.L. Hijazi, P.V. Hentenryck, The qc relaxation: a theoretical and computational study on optimal power flow, *IEEE Trans. Power Syst.* 31 (4) (2015) 3008–3018.
- [14] B. Eldridge, R.P. O'Neill, A. Castillo, Marginal loss calculations for the dcopf, Federal Energy Regulat. Commission, Tech. Rep (2017).
- [15] A. Marano-Marcolini, F. Capitanescu, J.L. Martinez-Ramos, L. Wehenkel, Exploiting the use of dc scopf approximation to improve iterative ac scopf algorithms, *IEEE Trans. Power Syst.* 27 (3) (2012) 1459–1466.
- [16] J.F. Restrepo, F.D. Galiana, Unit commitment with primary frequency regulation constraints, *IEEE Trans. Power Syst.* 20 (4) (2005) 1836–1842.
- [17] K. Karoui, H. Crisui, L. Platbrood, Modeling the primary reserve allocation in preventive and curative security constrained OPF, *Proc. IEEE PES Trans. Distrib. Conf. Expo. IEEE* (2010) 1–6.
- [18] J. Wang, M. Shahidehpour, Z. Li, Contingency-constrained reserve requirements in joint energy and ancillary services auction, *Trans. Power Syst.* 24 (3) (2009) 1457–1468.
- [19] D. Bertsimas, E. Litvinov, X.A. Sun, J. Zhao, T. Zheng, Adaptive robust optimization for the security constrained unit commitment problem, *Trans. Power Syst.* 28 (1) (2013) 52–63.
- [20] R. Rahmani, T.G. Crainic, M. Gendreau, W. Rei, The Benders decomposition algorithm: a literature review, *Eur. J. Oper. Res.* 259 (3) (2017) 801–817.
- [21] B. Zeng, L. Zhao, Solving two-stage robust optimization problems using a column-and-constraint generation method, *Oper. Res. Lett.* 41 (5) (2013) 457–461.
- [22] A.J. Ardakani, F. Bouffard, Identification of umbrella constraints in dc-based security-constrained optimal power flow, *IEEE Trans. Power Syst.* 28 (4) (2013) 3924–3934.
- [23] S. Babaeinejad-sarookolae, The power grid library for benchmarking ac optimal power flow algorithms, (2019). arXiv:1908.02788.