2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)

Online Demand Response of Voltage-Dependent Loads for Corrective Grid De-Congestion

Mohammadhafez Bazrafshan
Dept. Civil, Environmental & Architectural Eng.
University of Colorado Boulder
Boulder, CO
hafez.bazrafshan@icloud.com

Hao Zhu
Dept. Electrical & Computer Eng.
The University of Texas at Austin
Austin, TX
haozhu@utexas.edu

Amin Khodaei
Ritchie School of Eng. & Computer Science
University of Denver
Denver, CO
amin.khodaei@du.edu

Nikolaos Gatsis
Dept. Electrical & Computer Eng.
The University of Texas at San Antonio
San Antonio, TX
nikolaos.gatsis@utsa.edu

***Abstract*—During grid overload or upon occurrence of certain contingencies, a corrective action is required to eliminate congestion and reduce transmission line thermal limit violations. In this paper, we propose to use demand-responsive loads for such a purpose. Cost considerations include power retrieved from the slack reserves and the dis-utility of consumers for providing demand-response actions. Violations of voltage and generator reactive power limits are also accounted for. The idea is to topologically re-arrange the consumption of flexible loads to achieve grid de-congestion while maintaining the aggregate network power consumption constant to avoid interference with frequency control procedures. Our formulation is based on nonlinear power flows and easily allows the inclusion of voltage-dependent loads. An online gradient projection algorithm with closed-form updates is developed to solve the non-convex grid de-congestion problem. Approximate gradient calculations based on fast-decoupled load flow are further provided to simplify the algorithm and make it amenable to distributed implementation.**

I. Introduction

During the occurrence of contingencies or in periods of grid overload, a corrective action is required for grid de-congestion. Three options are typically available to system operators for this purpose. The first is corrective generator rescheduling [1], an action which is bound by the corresponding ramp constraints. A second traditional option is to additionally resort to load-shedding and admit loss of loads [2], [3]. A more recent option, enabled by communication capabilities of the smart grid, is demand-response (DR) [4]. The latter topologically rearranges the consumption of flexible loads so as to ameliorate line flow and voltage violations.

In preventive frameworks, DR has proved to be a viable option. Examples include security-constrained unit commitment problems [5] and security enhancement of operation reserves [6]. In these works, essentially, a demand pattern for a set of critical contingencies are decided. The result of preventive calculations may be used as look-up tables once similar contingencies occur in the network.

DR has also been applied in corrective frameworks. For instance, the work in [7] alleviates transformer overloads during contingencies by optimizing DR decisions. Transformer loss of life is calculated based on IEEE thermal and aging standards. Although multi-period, this work does not consider power flow equations. Corrective voltage-stability enhancement is achieved in [8] by leveraging DR of flexible loads. It is demonstrated that flexible loads can quickly adopt a different spatial pattern which in turn increases the distance to voltage collapse—as measured by the smallest singular value (SSV) of the power flow Jacobian. In such a setup, total network consumption remains constant to maintain system frequency. Based on power flow linearizations and computation of SSV sensitivities, an iterative algorithm where each step is a linear program is designed. An extension to a two-period setting is given in [9], where the second period acts as payback to ensure the energy consumed by each load returns to its nominal value. Impact of voltage-dependent load models, namely, ZIP loads and induction machines on a similar problem is further analyzed in [10]. It is indeed highlighted that optimal loading patterns may vary significantly for various load models.

In this paper, we utilize DR as a corrective tool to achieve grid de-congestion. The term de-congestion refers to ameliorating line flow congestion as measured by line thermal limit violations. Costs of dis-utility for providing DR services, cost of purchasing power from slack reserves, and associated penalties with voltage and generator reactive power limit violations are further considered. The formulation requires that the aggregate real power demand stay unchanged during

This work was supported by the National Science Foundation under Grants ECCS-1802319, ECCS-1807097, and ECCS-1847125.

978-1-5386-8099-5/19/$31.00 ©2019 IEEE

the corrective period so as to not interfere with frequency stability procedures. The developed algorithm features real-time updates that can be directly applied to the network in real-time. In comparison with our previous work [11] where we developed an online DR algorithm for voltage stability enhancement, this work provides four distinct novelties: (a) The new formulation easily handles any type of voltage-dependent load, (b) the cost of DR actions is considered, (c) DR actions are used for grid de-congestion, and (d) approximate gradient calculations are provided that highly simplify the algorithm and make it amenable to distributed implementation.

Our online algorithm is indeed inspired by recent developments in feedback-based optimization methods in optimal power flow (OPF) applications. In these approaches, real-time measurements are used to evaluate network conditions and calculate the next control action. Three main categories of these approaches can be identified. The first category, utilizes the the physical grid as an implicit power flow solver [12]–[14]. The controller setpoints rely on gradient calculation based on a mathematical model using the implicit function theorem. The real-time controller updates in the second category [15]–[18] is heavily dependent on linearized power flow equations—typically a valid option for distribution networks. Per iteration of the algorithm, real-time measurements are used in place of uncontrollable quantities. A third category directly develops a projected gradient descent method for an OPF problem where the variations in controls and system states are constrained by the power flow manifold [19], [20]. At every iteration, the controller updates are directly applied to the physical network.

Our present work is directly in line with the first group [12]–[14] in that we use the implicit function theorem to compute the gradients used for controller updates. A recent work [21], computes generator actions to alleviate line and voltage violations in real time. The algorithm in [21] uses the fast decoupled load flow (FDLF) to linearize the relationship between the power system states and control actions and additionally utilizes piece-wise linear approximations of the penalty functions to develop an efficient linear program at every iteration. In our work, DR is used for corrective action. We do not employ linearizations and instead leverage the structure of the constraints to develop closed-form controller updates. FDLF assumptions are only leveraged to approximate gradient calculations and simplify algorithm implementation.

This paper is structured as follows. Section II entails the proposed formulation. The solution methodology is detailed in Section III and is organized into problem reformulation, closed-form updates, gradient calculation, and algorithm implementation. Numerical tests are provided in Section IV.[1]

II. Network model and description of the problem

Denote by $(\mathcal{N}, \mathcal{E})$ the graph of a power network where $\mathcal{N}$ is the set of nodes and $\mathcal{E} := \{(i,j) : i < j, \quad i,j \in \mathcal{N}\}$ is the set of branches. In power system lexicon, nodes represent buses and branches represent transformers and transmission lines. For nodes $n \in \mathcal{N}$, denote by θ_n and v_n the voltage phase and magnitude. Denote further by $\rho_n(\theta_n, v_n)$ and $\gamma_n(\theta_n, v_n)$ the voltage-dependent parametric load model. For instance, the real power consumption of a ZIP load can explicitly be modeled as $\rho_n(\theta_n, v_n) = g_n^\ell v_n^2 + v_n i_n^\ell \cos\theta_n + p_n^\ell$ with g_n^ℓ, i_n^ℓ, and p_n^ℓ denoting the specified nominal conductance, nominal current, and nominal power consumption of load at node n.

Define the partition $\mathcal{N} = \{s\} \cup \mathcal{V} \cup \mathcal{P}$, where s is the slack bus, $\mathcal{V}$ is the set of voltage-controlled buses (henceforth $\mathcal{V}$-buses), and $\mathcal{P}$ is the remaining set of buses (henceforth, $\mathcal{P}$-buses). Let $\mathcal{G} \subseteq \mathcal{N}$ be the set of generators and denote by p_n and q_n the real and reactive power injection of the generator connected at node $n \in \mathcal{G}$. Define a matrix $C_g \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{G}|}$ where the (n, n_g)-th element of C_g is equal to 1 if node $n \in \mathcal{G}$ is the n_g-th node in $\mathcal{G}$ for an arbitrarily chosen order. We define following power balance functions:

$$\begin{aligned}\mathfrak{p}(\theta, v, p, d) := {} & [v][\cos\theta]G[v]\cos\theta + [v][\sin\theta]B[v]\cos\theta \\ & + [v][\sin\theta]G[v]\sin\theta - [v][\cos\theta]B[v]\sin\theta \\ & - C_g p + [\rho(\theta, v)]d \qquad \text{(1a)}\\ \mathfrak{q}(\theta, v, q, e) := {} & [v][\sin\theta]G[v]\cos\theta - [v][\cos\theta]B[v]\cos\theta \\ & - [v][\cos\theta]G[v]\sin\theta - [v][\sin\theta]B[v]\sin\theta \\ & - C_g q + [\gamma(\theta, v)]e. \qquad \text{(1b)}\end{aligned}$$

In (1), G and B are the real and imaginary parts of the bus admittance matrix. These power balance functions are based on the standard power flow equations in [22], [23] with additional modifications geared for this paper. The variables d and e represent a DR factor for $n \in \mathcal{N}$. For instance, varying d_n from 0 to 1 allows node n to go from zero to regular power consumption. The power flow equations are

$$\mathfrak{p}(\theta, v, p, d) = 0 \qquad \text{(2a)}$$

$$\mathfrak{q}(\theta, v, q, e) = 0 \qquad \text{(2b)}$$

The proposed DR formulation for grid de-congestion is

$$\underset{\theta,v,p,q,d,e}{\text{minimize}} \quad c_s(p_s) + \sum_{n\in\mathcal{D}} \kappa_n(d_n, \rho_n(\theta_n, v_n)) \qquad \text{(3a)}$$

subject to (2) and

$$\underline{d}_n \leq d_n,\ n \in \mathcal{D} \qquad \text{(3b)}$$

$$\sum_{n\in\mathcal{D}} d_n = \sum_{n\in\mathcal{D}} d_n^0 \qquad \text{(3c)}$$

$$\underline{e}_n \leq e_n \leq \bar{e}_n,\ n \in \mathcal{D} \qquad \text{(3d)}$$

$$\underline{p}_s \leq p_s \leq \bar{p}_s \qquad \text{(3e)}$$

$$\underline{v}_n \leq v_n \leq \bar{v}_n,\ n \in \mathcal{P} \qquad \text{(3f)}$$

$$\underline{q}_n \leq q_n \leq \bar{q}_n,\ n \in \mathcal{V} \cup \{s\} \qquad \text{(3g)}$$

$$\mathfrak{s}_b^f(\theta, v) \leq \bar{s}_b,\ b \in \mathcal{E} \qquad \text{(3h)}$$

$$\mathfrak{s}_b^t(\theta, v) \leq \bar{s}_b,\ b \in \mathcal{E} \qquad \text{(3i)}$$

[1] *Notation:* Vectors are denoted by lower-case letters. For vector x, index set $\mathcal{A}$, and an index $i \in \mathcal{A}$, denote respectively by x_i and $x_{\mathcal{A}}$, the element of x corresponding to index i and a vector of size $|\mathcal{A}|$ comprised of x_i's for $i \in \mathcal{A}$. For a vector x, $[x]$ denotes a diagonal matrix with x on the diagonal. Transpose of $(.)$ is denoted by $(.)'$.

$$\theta_s = 0, v_n = v_n^{\mathrm{sp}}, n \in \mathcal{V} \cup \{s\}, p_n = p_n^{\mathrm{sp}}, n \in \mathcal{G} \setminus \{s\} \quad \text{(3j)}$$
$$q_n = q_n^{\mathrm{sp}}, n \in \mathcal{G} \cap \mathcal{P}, d_n = d_n^0, e_n = e_n^0, n \in \mathcal{N} \setminus \mathcal{D} \quad \text{(3k)}$$

In (3), $c_s(p_s)$ is the cost of providing power from slack reserves during the correction period. The set $\mathcal{D}$ collects the set of nodes $n \in \mathcal{N}$ that participate in DR. The dis-utility functions $\kappa_n(d_n, \rho_n(\theta_n, v_n))$ measure the cost imposed on consumer $n \in \mathcal{D}$ for providing DR services. These dis-utility functions are designed so that if d_n equals 1 the dis-utility is zero and their values increase as d_n varies away from 1.

Constraint (3b) requires that a minimum amount of load, given by the factor $\underline{d}_n$ must be supported at node $n \in \mathcal{D}$. Constraint (3c) requires that the aggregate network demand remains constant during the corrective period. A similar constraint is utilized in [8]–[10]. The quantities d_n^0 and e_n^0 represent the initial real and reactive demand adjustment factors at the beginning of the corrective action. Constraint (3d) provides lower and upper limits $\underline{e}_n$ and $\bar{e}_n$ for reactive power compensation at node $n \in \mathcal{D}$. The assumption here is that reactive power can be independently controlled for nodes participating in DR. If a fixed power factor is required by the load, this constraint can be modified accordingly.

Constraints (3e), (3f), and (3g) enforce respectively the limits on power drawn from the slack reserves, the voltage bounds on $\mathcal{P}$-buses, as well as reactive power limits of $\mathcal{V}$-buses and the slack bus. Quantities $\mathfrak{s}_b^f(\theta, v)$ and $\mathfrak{s}_b^t(\theta, v)$ measure the squared magnitude of the from and to power flows on branch $b \in \mathcal{E}$ respectively. These quantities are upper bounded by the squared magnitude of line rating $\bar{\mathfrak{s}}_b$ in (3h) and (3i). Constraints listed in (3j) and (3k) ensure the appropriate setpoints for the slack bus, $\mathcal{V}$-buses, and $\mathcal{P}$-buses as well as DR factors of buses not participating in DR.

The DR problem (3) is non-convex. Indeed, not only the power flow equations (2) are nonlinear, the multiplication of DR factors d and e respectively with the nonlinear voltage-dependent load models $\rho(\theta, v)$ and $\gamma(\theta, v)$ are complicated. Although convex relaxations of power flow equation could be used [24], obtaining a feasible solution that satisfies the power flows might not be guaranteed for this problem. We seek a method that can interact with the grid to produce an online feasible solution with small costs in the next section.

III. Solution methodology

A projected gradient-descent algorithm is proposed here to approximately solve the non-convex problem (3). Our algorithm relies on a reformulation of (3) presented next. Details on closed-form projections, approximate gradient calculations, and algorithm implementation are postponed to later sections.

A. Reformulation

Let us first collect the variables in (3) in the vector $z = (\theta, v, p, q, d, e)$ where $\theta, v \in \mathbb{R}^{|\mathcal{N}|}$, $p, q \in \mathbb{R}^{|\mathcal{G}|}$ and $d, e \in \mathbb{R}^{|\mathcal{N}|}$. Assigning as parameters the indices of the corresponding variables given by the setpoints listed in constraints (3j) and (3k), verifies that the unknowns in (3) are the quantities $\theta_{\mathcal{V}\cup\mathcal{P}}$, $v_{\mathcal{P}}$, p_s, $q_{\mathcal{V}\cup\{s\}}$, $d_{\mathcal{D}}$, and $e_{\mathcal{D}}$ amounting to $2|\mathcal{N}| + 2|\mathcal{D}|$ variables. One may recall at this point that in a transmission power flow, the values $x = (\theta_{\mathcal{V}\cup\mathcal{P}}, v_{\mathcal{P}}, p_s, q_{\mathcal{V}\cup\{s\}})$ are not directly controllable. Indeed, by controlling the $2|\mathcal{D}|$ variables d and e the quantities in x may all be determined by the *physics* of the problem, i.e., the power flows (2). We shall thus formally interpret (2) as

$$f(x, u) = 0, \tag{4}$$

where the function $f(.) : \mathbb{R}^{2|\mathcal{N}|+2|\mathcal{D}|} \to \mathbb{R}^{2|\mathcal{N}|}$ is continuously differentiable and we have $u = (d_{\mathcal{D}}, e_{\mathcal{D}})$. Suppose that $\partial f / \partial x$ evaluated at an operating point (x^0, u^0) is invertible. By the implicit function theorem [25, Theorem 9.28], x is a *locally* one-to-one function of u^0 in a neighborhood $\mathcal{U}$ of u^0, that is,

$$x = x(u), u \in \mathcal{U} \tag{5}$$

Derivative of x at u^0 is computed as follows:

$$\frac{\partial x}{\partial u}\Big|_{u^0} = -\left(\frac{\partial f}{\partial x}\Big|_{(x^0,u^0)}\right)^{-1} \frac{\partial f}{\partial u}\Big|_{(x^0,u^0)}. \tag{6}$$

Equation (6) proves useful later in this paper for the computation of the gradient in our algorithm.

Informed by (5), in the set $\mathcal{U}$, (3) is equivalent to

$$\begin{aligned}
\underset{(d_{\mathcal{D}}, e_{\mathcal{D}}) \in \mathcal{U}}{\text{minimize}} \quad & c_s(p_s(d_{\mathcal{D}}, e_{\mathcal{D}})) \\
& + \sum_{n \in \mathcal{D}} \kappa_n(d_n, \rho_n(\theta_n(d_n, e_n), v_n(d_n, v_n))) && \text{(7a)} \\
\text{subject to} \quad & \text{(3b)}, \text{(3c)}, \text{(3d)}, \text{(5) and} \\
& \underline{p}_s \leq p_s(d_n, e_n) \leq \bar{p}_s && \text{(7b)} \\
& \underline{v}_n \leq v_n(d_n, e_n) \leq \bar{v}_n, \ n \in \mathcal{P} && \text{(7c)} \\
& \underline{q}_n \leq q_n(d_n, e_n) \leq \bar{q}_n, \ n \in \mathcal{V} \cup \{s\} && \text{(7d)} \\
& \mathfrak{s}_b^f(\theta(d, e), v(d, e)) \leq \bar{s}_b, \ b \in \mathcal{E} && \text{(7e)} \\
& \mathfrak{s}_b^t(\theta(d, e), v(d, e)) \leq \bar{s}_b, \ b \in \mathcal{E} && \text{(7f)}
\end{aligned}$$

Constraints (7b)–(7f) rely entirely on u and cannot be directly enforced. An approximate problem that penalizes these constraints must be administered. Define the penalty function

$$\phi(z) = \max\{z, 0\}^2. \tag{8}$$

The approximate problem is

$$\begin{aligned}
\underset{u}{\text{minimize}} \quad & \Phi(u; \mu) && \text{(9a)} \\
\text{subject to} \quad & \text{(3b)}, \text{(3c)}, \text{(3d)}, \text{ and (5)}, && \text{(9b)}
\end{aligned}$$

where $\mu = (\mu_s^p, \mu_n^v, \mu_n^q, \mu_b^{\mathfrak{s}})$ is a vector penalty parameter and

$$\begin{aligned}
\Phi(u; \mu) := {} & c_s(p_s) + \kappa_n(d_n, \rho_n(\theta_n(d_n, e_n), v_n(d_n, e_n))) \\
& + \mu_s^p \Big(\phi(p_s(u) - \bar{p}_s) + \phi(\underline{p}_s - p_s(u))\Big) \\
& + \sum_{n \in \mathcal{P}} \mu_n^v \left(\phi(v_n(u) - \bar{v}_n) + \phi(\underline{v}_n - v_n(u))\right) \\
& + \sum_{n \in \mathcal{V} \cup \{s\}} \mu_n^q \Big(\phi(q_n(u) - \bar{q}_n) + \phi(\underline{q}_n - q_n(u))\Big) \\
& + \sum_{b \in \mathcal{E}} \mu_b^{\mathfrak{s}} \Big(\phi(\mathfrak{s}_b^f(u) - \bar{s}_b) + \phi(\mathfrak{s}_b^t(u) - \bar{s}_b)\Big) && \text{(10)}
\end{aligned}$$

B. Projected gradient descent

Problem (9) is solved with projected gradient-descent:

$$d_{\mathcal{D}}^{(k+1)} = \Pi_{\mathcal{Z}} \left(d_{\mathcal{D}}^{(k)} - \alpha^{(k)} \left(\frac{\partial \Phi(u;\mu)}{\partial d_{\mathcal{D}}} \Big|_{u^{(k)}} \right)' \right) \quad (11)$$

$$e_{\mathcal{D}}^{(k+1)} = \Pi_{\mathcal{B}} \left(e_{\mathcal{D}}^{(k)} - \alpha^{(k)} \left(\frac{\partial \Phi(u;\mu)}{\partial e_{\mathcal{D}}} \Big|_{u^{(k)}} \right)' \right) \quad (12)$$

where $\mathcal{Z}$ concisely represents constraints (3b) and (3c) while $\mathcal{B}$ is the box constraint (3d). In (11), Π is the projection operator,superscript (k) denotes the iteration index, and $\alpha^{(k)}$ is a step-size parameter. Given the gradient $\frac{\partial \Phi(u;\mu)}{\partial e_{\mathcal{D}}}\big|_{u^{(k)}}$, update (12) is an Euclidean projection on a box constraint which has a closed-form solution. Update (11) seems slightly more complicated but it also permits a closed-form solution as we explain next. It can be shown that $d_{\mathcal{D}}^{(k+1)}$ can be obtained by the following update:

$$d_{\mathcal{D}}^{(k+1)} = \left(\sum_{n \in \mathcal{D}} (d_n^0 - \underline{d}_n) \right) y^* + \underline{d}_{\mathcal{D}} \quad (13)$$

where y^* is the optimal value of the quadratic program:

$$\underset{y \geq 0}{\text{minimize}} \quad \| y - \frac{d_{\mathcal{D}}^{(k)} - \alpha \frac{\partial \Phi(u;\mu)}{\partial d_{\mathcal{D}}} \big|'_{u^{(k)}} - \underline{d}_{\mathcal{D}}}{\sum\limits_{n \in \mathcal{D}} (d_n^0 - \underline{d}_n)} \|_2 \quad (14a)$$

$$\text{subject to} \quad \|y\|_1 = 1. \quad (14b)$$

The quadratic program (14) is an Euclidean projection on the probability simplex [26] whose solution may be obtained via bisection on the one dimensional dual variable [27, Exercise 1], or via a sorting-based non-iterative method [28]. Gradient computations are detailed in the next section.

C. Gradient calculation and approximation

The derivative of $\Phi(u;\mu)$ with respect to u is given by

$$\begin{aligned}\frac{\partial \Phi}{\partial u} = & \frac{\partial c_s}{\partial p_s}\frac{\partial p_s}{\partial u} + \sum_{n \in \mathcal{D}} \frac{\partial \kappa_n}{\partial u} + \frac{\partial \kappa_n}{\partial \theta_n}\frac{\partial \theta_n}{\partial u} + \frac{\partial \kappa_n}{\partial v_n}\frac{\partial v_n}{\partial u} \\ & + \mu_s^p \frac{\partial \phi(p_s)}{\partial p_s}\frac{\partial p_s}{\partial u} \\ & + \sum_{n \in \mathcal{P}} \mu_n^v \left(\frac{\partial \phi(v_n - \bar{v}_n)}{\partial v_n} + \frac{\partial \phi(\underline{v}_n - v_n)}{\partial v_n} \right) \frac{\partial v_n}{\partial u} \\ & + \sum_{n \in \mathcal{V} \cup \{s\}} \mu_n^q \left(\frac{\partial \phi(q_n - \bar{q}_n)}{\partial q_n} + \frac{\partial \phi(\underline{q}_n - q_n)}{\partial q_n} \right) \frac{\partial q_n}{\partial u} \\ & + \sum_{b \in \mathcal{E}} \mu_b^{\mathfrak{s}} \left(\frac{\partial \phi(\mathfrak{s}_b^f - \bar{s}_b)}{\partial \mathfrak{s}_b^f}\frac{\partial \mathfrak{s}_b^f}{\partial u} + \frac{\partial \phi(\mathfrak{s}_b^t - \bar{s}_b)}{\partial \mathfrak{s}_b^t}\frac{\partial \mathfrak{s}_b^t}{\partial u} \right) \quad (15)\end{aligned}$$

In (15), the expressions $\frac{\partial c_s}{\partial p_s}$, $\frac{\partial \kappa_n}{\partial u}$, $\frac{\partial \kappa_n}{\partial \theta_n}$, $\frac{\partial \kappa_n}{\partial v_n}$ are obtained by differentiating the cost functions $c_s(.)$ and $\kappa_n(.)$. The calculation of cost functions $c_s(.)$ and $\kappa_n(.)$ and their derivatives are decentralized per bus since these values solely rely on local measurements—p_s for the slack bus and θ_n, v_n for $n \in \mathcal{D}$. The expressions $\frac{\partial \phi(.)}{\partial v_n}$, $\frac{\partial \phi(.)}{\partial q_n}$, $\frac{\partial \phi(.)}{\partial \mathfrak{s}_b^f}$, and $\frac{\partial \phi(.)}{\partial \mathfrak{s}_b^t}$ are easily computed by acknowledging that the derivative of $\phi(.)$ in (8) is $2\max\{z, 0\}$. These calculations are also decentralized per bus for $\frac{\partial \phi(.)}{\partial v_n}$, $\frac{\partial \phi(.)}{\partial q_n}$, and per branch for $\frac{\partial \phi(.)}{\partial \mathfrak{s}_b^f}$ and $\frac{\partial \phi(.)}{\partial \mathfrak{s}_b^t}$.

The major challenge in computing the gradient is in obtaining the values for $\frac{\partial p_s}{\partial u}$, $\frac{\partial \theta_n}{\partial u}$, $\frac{\partial v_n}{\partial u}$, and $\frac{\partial q_n}{\partial u}$ for n in the appropriate sets. These values are entries of $\frac{\partial x}{\partial u}$ from (6) where we recall that $x = (\theta_{\mathcal{V} \cup \mathcal{P}},\ v_{\mathcal{P}},\ p_s,\ q_{\mathcal{V} \cup \{s\}})$. The quantity $\frac{\partial f}{\partial u}$ is obtained by differentiating (1) and selecting appropriate indices in the following:

$$\frac{\partial \mathfrak{p}}{\partial d} = [\rho(\theta, v)] \qquad \frac{\partial \mathfrak{p}}{\partial e} = \mathbf{O}_{|\mathcal{N}| \times |\mathcal{N}|} \quad (16a)$$

$$\frac{\partial \mathfrak{q}}{\partial d} = \mathbf{O}_{|\mathcal{N}| \times |\mathcal{N}|} \qquad \frac{\partial \mathfrak{q}}{\partial e} = [\gamma(\theta, v)]. \quad (16b)$$

These computations are decentralized per node as the load models $\rho(.)$ and $\gamma(.)$ only depend on local variables per bus. Computation of $(\frac{\partial f}{\partial x}\big|_{(x^0,u^0)})^{-1}$ in (6) is more challenging as it first involves differentiating (1) with respect to θ, v, p, and q and then performing matrix inversion. More taxing is that this computation needs to be performed per iteration k as the value of (x, u) evolves. Fortunately, the derivatives with respect to p and q are fixed per iteration and given as follows:

$$\frac{\partial \mathfrak{p}}{\partial p} = -C_g \qquad \frac{\partial \mathfrak{p}}{\partial q} = \mathbf{O}_{|\mathcal{N}| \times |\mathcal{G}|} \quad (17a)$$

$$\frac{\partial \mathfrak{q}}{\partial p} = \mathbf{O}_{|\mathcal{N}| \times |\mathcal{G}|} \qquad \frac{\partial \mathfrak{q}}{\partial q} = \ -C_g \quad (17b)$$

Computing the derivatives of (1) with respect to θ and v vary per iteration. Based on the decoupled power flow principles [29], one can use the following approximations per iteration:

$$\frac{\partial \mathfrak{p}}{\partial \theta} = -B + [d][\frac{\partial \rho}{\partial \theta}] \qquad \frac{\partial \mathfrak{p}}{\partial v} = \ [d][\frac{\partial \rho}{\partial v}] \quad (18a)$$

$$\frac{\partial \mathfrak{q}}{\partial d} = [d][\frac{\partial \gamma}{\partial \theta}] \qquad \frac{\partial \mathfrak{q}}{\partial \theta} = -\ B + [\frac{\partial \gamma}{\partial v}] \quad (18b)$$

To further simplify matters, we suggest to compute the values in (18) only at the initial point of the corrective action. In this way, the inversion process of $(\frac{\partial f}{\partial x}\big|_{(x^0,u^0)})^{-1}$ in (6) can be done only once at the start of the algorithm. Our numerical simulations show that this approximate gradient calculation does not significantly compromise the algorithm performance. Detailed overview of the implementation is discussed next.

D. Algorithm Implementation

Algorithm 1 details the implementation of the proposed algorithm. The stepsize parameter α, the stopping criteria ϵ, and the penalty parameters μ are selected before the start of the algorithm. Per step k, measurements $x(u^{(k)})$ are collected across buses and branches. The gradient value of $\frac{\partial \Phi}{\partial u}$ is calculated following the details in Section III-C. Then, the closed-form updates (11) and (12) are carried out by a central operator. The new values of $u^{(k)}$ are broadcast to buses $n \in \mathcal{D}$. Further insight into the implementation is provided in Fig. 1. In the figure, oval shapes represent buses and branches while boxes represent computational elements. Buses are divided into the slack-bus, $\mathcal{V}$-buses, and $\mathcal{P}$-buses. Any bus $n \in \mathcal{D}$ can access its own particular load model and contribute to

Algorithm 1 Grid De-congestion via Online DR

```
Select step-size parameter $\alpha$, penalty parameters $\mu$
Select objective improvement threshold $\epsilon$
Set $\Phi(u^{(-1)};\mu) = +\infty$ and $k = 0$
Gather measurements $x(u^{(0)})$
Compute $\Phi(u^{(0)};\mu)$ and $\frac{\partial \Phi(u;\mu)}{\partial u}\Big|_{u^{(0)}}$
while $\Phi(u^{(k-1)};\mu) - \Phi(u^{(k)};\mu) > \epsilon$ do
    Apply updates (11) and (12)
    $k \leftarrow k+1$
    Gather measurements $x(u^{(k)})$
    Compute $\Phi(u^{(k)};\mu)$ and $\frac{\partial \Phi(u;\mu)}{\partial u}\Big|_{u^{(k)}}$
end while
```

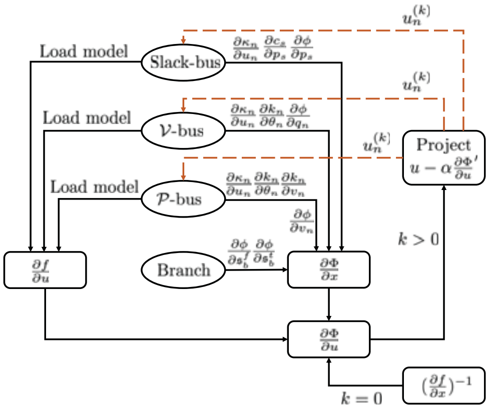
$u_n^{(k)}$
Load model
Slack-bus
$\frac{\partial \kappa_n}{\partial u_n}\frac{\partial c_s}{\partial p_s}\frac{\partial \phi}{\partial p_s}$
$u_n^{(k)}$
Load model
$\mathcal{V}$-bus
$\frac{\partial \kappa_n}{\partial u_n}\frac{\partial k_n}{\partial \theta_n}\frac{\partial \phi}{\partial q_n}$
$u_n^{(k)}$
Project
$u - \alpha\frac{\partial \Phi}{\partial u}'$
Load model
$\mathcal{P}$-bus
$\frac{\partial \kappa_n}{\partial u_n}\frac{\partial k_n}{\partial \theta_n}\frac{\partial k_n}{\partial v_n}$
$\frac{\partial \phi}{\partial v_n}$
$k > 0$
$\frac{\partial f}{\partial u}$
Branch
$\frac{\partial \phi}{\partial \mathfrak{s}_b^f}\frac{\partial \phi}{\partial \mathfrak{s}_b^t}$
$\frac{\partial \Phi}{\partial x}$
$\frac{\partial \Phi}{\partial u}$
$(\frac{\partial f}{\partial x})^{-1}$
$k = 0$

Fig. 1. A high-level schematic of the proposed algorithm.

the calculation of $\frac{\partial f}{\partial u}$ according to (16). To evaluate $\frac{\partial \Phi}{\partial x}$, all buses calculate the value of $\frac{\partial k_n}{\partial u_n}$, $\frac{\partial k_n}{\partial \theta_n}$, and $\frac{\partial k_n}{\partial v_n}$ according to the per-bus dis-utility function and particular load model. The slack-bus computes $\frac{\partial c_s}{\partial p_s}$ and $\frac{\partial \phi}{\partial p_s}$. The $\mathcal{V}$-buses and $\mathcal{P}$-buses also respectively compute the values of $\frac{\partial \phi}{\partial q_n}$ and $\frac{\partial \phi}{\partial v_n}$. These computations are all decentralized per bus. The branches further contribute to $\frac{\partial \Phi}{\partial x}$ by calculating $\frac{\partial \phi}{\partial \mathfrak{s}_b^f}$ and $\frac{\partial \phi}{\partial \mathfrak{s}_b^t}$. These computations are distributed between neighboring buses.

Having computed the values of $\frac{\partial f}{\partial u}$ in a decentralized manner and the values of $\frac{\partial \Phi}{\partial x}$ in a distributed fashion, the computation of $\frac{\partial x}{\partial u}$ through (6) is straightforward. An approximation of $(\frac{\partial f}{\partial x})^{-1}$ can be computed from (18) at the start of the corrective DR procedure. This approximation highly simplifies the implementation and allows for a distributed update of $\frac{\partial x}{\partial u}$ per iteration. Finally, the central operator can collect the calculated gradient $\frac{\partial \Phi}{\partial u}$ and perform the closed-form updates (11) and (12). Ultimately, the new DR factors are distributed across the corresponding buses to determine new setpoints for the next iteration. This procedure is continued until the measurements $x(u^{(k)})$ indicate that the improvement rate of $\Phi(u^{(k)};\mu)$ is satisfactorily small.

IV. Numerical Experiments

We test the proposed algorithm on the 39-bus transmission system here. The network is overloaded by a ZIP model whose constant-admittance, constant-current, and constant-power portions are respectively set to 10%, 50%, and 70% of the constant-power consumption of the original network. The cost of slack-bus reserves is considered equal to the per-unit power supplied during the correction period, that is, we set $c_s(p_s) = p_s$. The dis-utility function is chosen as

$$\kappa_n(d_n, \rho_n(\theta_n, v_n)) = \rho_n(\theta_n, v_n)(d_n^0 - d_n)^2, n \in \mathcal{D}. \quad (19)$$

The function in (19) is zero if the DR factor d_n remains at d_n^0, and increases and scales by the per-unit load consumption otherwise. The set $\mathcal{D}$ is selected as buses with a non-zero nominal real power demand. We set $\underline{d}_n = 0.7$ for $n \in \mathcal{D}$. We only investigate real power DR by setting $\underline{e}_n = \bar{e}_n = 1$.

TABLE I
Cost comparison

Objective	Before DR	True Gradient	Approx. Gradient
$\Phi(u;\mu)$	16817.78	2838.09	3605.31
$c_s(p_s)$	19.49	18.00	18.14
$\sum_{n\in\mathcal{D}} \kappa_n$	0.0	1.25	1.36
$\phi(p_s)$	169.79	133.22	136.46
$\phi(v)$	$<$ 1e-4	$<$ 1e-4	$<$ 1e-4
$\phi(q)$	35.05	20.78	21.85
$\phi(\mathfrak{s})$	16593.45	2664.84	3427.50

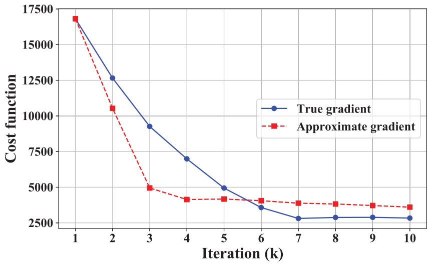

Fig. 2. Algorithm performance on the overloaded 39-bus system

Table I summarizes the algorithm performance after 10 iterations with $\alpha = 0.00001$ and $\mu = 1$. The first column of the table are the breakdown of $\Phi(u;\mu)$. The penalties for the slack bus power injection p_s, voltage violations v_n for $n \in \mathcal{P}$, generator reactive powers q_n for $n \in \mathcal{V}\cup\{s\}$, and line flows $\mathfrak{s}_b^f$ and $\mathfrak{s}_b^t$ for $b \in \mathcal{E}$ are abbreviated respectively by $\phi(p_s)$, $\phi(v)$, $\phi(q)$ and $\phi(\mathfrak{s})$. We observe in the table that before applying DR the total cost is dominated by line congestion costs $\phi(\mathfrak{s})$.

By applying algorithm 1 using the true gradient all the penalty parameters decrease. In particular, the congestion cost $\phi(\mathfrak{s})$ decreases significantly yielding a very low total cost after the corrective DR action. Applying the proposed algorithm using the approximate gradient also reduces the congestion cost significantly. Although the final cost using the approximate gradient is slightly worse than that of the true gradient, the implementation is highly simplified.

The iterations of the algorithm with both the true gradient and the approximate gradient are demonstrated in Fig. 2. The x-axis denotes the iteration values k and the y-axis denotes the cost function value. It is observed that both algorithms significantly decrease the objective function with the true gradient ultimately settling at a lower cost value. The iterations k of our proposed method can be interpreted as time steps k as each update is applied to the network in real time.

Figure 3 demonstrates the real-time fluctuation of the real power DR factors d_n for $n \in \mathcal{D}$ using the approximate

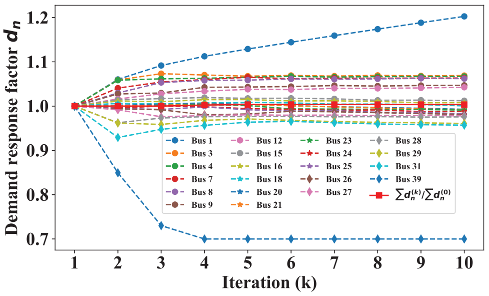

Fig. 3. Online coordination of DR factors.

gradient. A similar figure may be obtained for the true gradient implementation. The set $\mathcal{D}$ is explicitly provided in the legend. For each $n \in \mathcal{D}$, the profile of the real power DR factors $d_n^{(k)}$ varies smoothly throughout the evolution of the algorithm. It is observed that none of the DR factors violate the lower limit of 0.7. The solid red line indicates the value of $\sum_{n\in\mathcal{D}} d_n^{(k)} / \sum_{n\in\mathcal{D}} d_n^{(0)}$. Indeed, this value is always guaranteed to be equal to 1 by the projection step (11).

V. Conclusion

This paper proposes demand response be used as a corrective tool to reduce line thermal violations. While the aggregate power demand stays unchanged to avoid triggering frequency control actions, consumption of demand responsive loads may be redistributed between buses to achieve de-congestion. The presented network model accounts for nonlinear power flows and voltage-dependent loads. The objective is to minimize costs of providing DR services, purchasing power from slack reserves, and penalties associated with voltage and generator reactive power limit violations. The optimization model is a non-convex program. Instead of convex approximations or relaxations, the structure of the constraints are leveraged to derive an online gradient projection algorithm that tackles the non-convex program. While examining the theoretical properties of the proposed algorithm is postponed to future work, numerical experiments on an overloaded network with ZIP loads evidence the applicability of the proposed algorithm for online corrective grid de-congestion via demand response.

References

[1] A. Monticelli, M. V. F. Pereira, and S. Granville, "Security-Constrained Optimal Power Flow with Post-Contingency Corrective Rescheduling," *IEEE Trans. Power Syst.*, vol. 2, no. 1, pp. 175–180, 1987.

[2] A. Shandilya, H. Gupta, and J. Sharma, "Method for generation rescheduling and load shedding to alleviate line overloads using local optimisation," *IEE Proc. C Gener. Transm. Distrib.*, vol. 140, no. 5, p. 337, 1993.

[3] D. Hazarika and A. Sinha, "Method for optimal load shedding in case of generation deficiency in a power system," *Int. J. Electr. Power Energy Syst.*, vol. 20, no. 6, pp. 411–420, Aug. 1998.

[4] US Department of Energy, "Benefits of Demand Response in Electricity Markets and Recommendations for Achieving Them," Tech. Rep., 2005. [Online]. Available: https://eetd.lbl.gov/sites/all/files/publications/report-lbnl-1252d.pdf

[5] A. Khodaei, M. Shahidehpour, and S. Bahramirad, "SCUC With Hourly Demand Response Considering Intertemporal Load Characteristics," *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 564–571, Sept. 2011.

[6] Y. Wang, I. R. Pordanjani, and W. Xu, "An Event-Driven Demand Response Scheme for Power System Security Enhancement," *IEEE Trans. Smart Grid*, vol. 2, no. 1, pp. 23–29, Mar. 2011.

[7] M. Humayun, A. Safdarian, M. Z. Degefa, and M. Lehtonen, "Demand Response for Operational Life Extension and Efficient Capacity Utilization of Power Transformers During Contingencies," *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 2160–2169, July 2015.

[8] M. Yao, J. L. Mathieu, and D. K. Molzahn, "Using demand response to improve power system voltage stability margins," in *Proc. IEEE Manchester PowerTech*, June 2017, pp. 1–6.

[9] M. Yao, D. K. Molzahn, and J. L. Mathieu, "An Optimal Power Flow Approach to Improve Power System Voltage Stability Using Demand Response," *IEEE Trans. Control. Netw. Syst.*, April 2019, to be published. [Online]. Available: https://ieeexplore.ieee.org/document/8686211

[10] ——, "The impact of load models in an algorithm for improving voltage stability via demand response," in *Proc. 55th Annu. Allert. Conf. Commun. Control. Comput.*, Oct. 2017, pp. 149–156.

[11] M. Bazrafshan, N. Gatsis, and H. Zhu, "Real-Time Voltage Stability Enhancement via Demand Response," submitted.

[12] L. Gan and S. H. Low, "An Online Gradient Algorithm for Optimal Power Flow on Radial Networks," *IEEE J. Sel. Areas Commun.*, no. 3, pp. 625–638, Mar. 2016.

[13] Y. Tang, K. Dvijotham, and S. Low, "Real-Time Optimal Power Flow," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2963–2973, Nov. 2017.

[14] Y. Tang and S. Low, "Distributed algorithm for time-varying optimal power flow," in *Proc. 56th Annu. Conf. Decis. Control*, Dec. 2017, pp. 3264–3270.

[15] A. Bernstein and E. Dall'Anese, "Real-Time Feedback-Based Optimization of Distribution Grids: A Unified Approach," *IEEE Trans. Control. Netw. Syst.*, 2019, to be published. [Online]. Available: https://ieeexplore.ieee.org/document/8767939

[16] X. Zhou, E. Dall'Anese, L. Chen, and A. Simonetto, "An Incentive-Based Online Optimization Framework for Distribution Grids," *IEEE Trans. Autom. Control*, vol. 63, no. 7, pp. 2019–2031, 2018.

[17] K. Baker, A. Bernstein, E. Dall'Anese, and C. Zhao, "Network-cognizant voltage droop control for distribution grids," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 2098–2108, Mar. 2018.

[18] Y. Zhang, E. Dall'Anese, and M. Hong, "Dynamic ADMM for real-time optimal power flow," in *Proc. Glob. Conf. Signal Inf. Process.*, Nov. 2017, pp. 1085–1089.

[19] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, "Projected gradient descent on Riemannian manifolds with applications to online power system optimization," in *Proc. 54th Annu. Allert. Conf. Commun. Control. Comput.*, Sept. 2016, pp. 225–232.

[20] A. Hauswirth, A. Zanardi, S. Bolognani, F. Dörfler, and G. Hug, "Online optimization in closed loop on the power flow manifold," in *Proc. PowerTech Conf.*, Jun 2017, pp. 1–6.

[21] N. Mazzi, B. Zhang, and D. S. Kirschen, "An Online Optimization Algorithm for Alleviating Contingencies in Transmission Networks," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 5572–5582, Sept. 2018.

[22] R. D. Zimmerman and C. E. Murillo-Sánchez, "Matpower 6.0 User's Manual," 2016.

[23] G. B. Giannakis, V. Kekatos, N. Gatsis, S.-J. Kim, H. Zhu, and B. F. Wollenberg, "Monitoring and Optimization for Power Grids: A Signal Processing Perspective," *IEEE Signal Process. Mag.*, vol. 30, no. 5, pp. 107–128, Sept. 2013.

[24] J. A. Taylor, *Convex Optimization of Power Systems*. Cambridge University Press, 2015.

[25] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed. New York: McGraw-Hill, 1976.

[26] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l1-ball for learning in high dimensions," in *Proc. 25th Int. Conf. on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 272–279. [Online]. Available: http://doi.acm.org/10.1145/1390156.1390191

[27] S. Boyd, "EE364b Homework 4," Tech. Rep. [Online]. Available: https://see.stanford.edu/materials/lsocoee364b/hw4sol.pdf

[28] Y. Chen and X. Ye, "Projection Onto A Simplex," Tech. Rep., 2011. [Online]. Available: http://www.math.ufl.edu/

[29] B. Stott and O. Alsac, "Fast Decoupled Load Flow," *IEEE Trans. Power Appar. Syst.*, vol. PAS-93, no. 3, pp. 859–869, May 1974.