

RESEARCH

Open Access

A distributed approach to the OPF problem

Tomaso Erseghe

Abstract

This paper presents a distributed approach to optimal power flow (OPF) in an electrical network, suitable for application in a future smart grid scenario where access to resource and control is decentralized. The non-convex OPF problem is solved by an augmented Lagrangian method, similar to the widely known ADMM algorithm, with the key distinction that penalty parameters are constantly increased. A (weak) assumption on local solver reliability is required to always ensure convergence. A certificate of convergence to a local optimum is available in the case of bounded penalty parameters. For moderate sized networks (up to 300 nodes, and even in the presence of a severe partition of the network), the approach guarantees a performance very close to the optimum, with an appreciably fast convergence speed. The generality of the approach makes it applicable to any (convex or non-convex) distributed optimization problem in networked form. In the comparison with the literature, mostly focused on convex SDP approximations, the chosen approach guarantees adherence to the reference problem, and it also requires a smaller local computational complexity effort.

Keywords: Alternating direction method of multipliers; Augmented Lagrangian methods; Convergence guarantee; Distributed processing; Optimal power flow; Smart grid

Introduction

One of the key aspects of the current research trends for the future smart grid is the possibility of devising distributed algorithms for solving a global problem. This corresponds to the idea of a decentralized access to generation/storage resources, as well as to the much more challenging task of decentralized control.

The typical smart grid problem taken into consideration for distributed optimization is that of optimal power flow (OPF), that is, the optimal management of electrical power throughout the grid under a number of (electrical) constraints (e.g., the satisfaction of a power request from a load, the presence of a dispatchable/non dispatchable renewable generator or of a storage system). The OPF problem, being non-convex in nature in both the target function and the constraints, is very difficult to solve. For this reason, a widely used approach is to *map* it into a (somehow) close convex problem, and then solve the convex counterpart by means of distributed methods, e.g., the alternating direction method of multipliers. In this context, semi definite programming (SDP) relax-

ations have emerged as a common option, e.g., see Lavaei and Sojoudi et al. [1-4], Lam, Tse, and Zhang et al. [5,6], Dall'Anese and Giannakis et al. [7,8], Gayme and Topcu [9], and Erseghe and Tomasin [10]. One of the limits of this approach lies in the lack of adherence to the original problem, and in fact, optimality of the solution can only be ensured for very specific networks. But complexity is also an issue, since the number of variables involved in the local processing is squared with respect to its natural size. A few other worth mentioning approaches are available from the literature. Šulc et al. [11] exploit the (convex) LinDistFlow approximation as a lower complexity alternative to SDP relaxation. Magnusson et al. [12] avoid SDP relaxation and propose a sequential convex approximation approach, which, however, is known to imply slow convergence speeds. Instead, the *consensus and innovation* approach has been applied to the (convex) DC-OPF problem by Hug and Kar et al. [13,14], but the chosen distributed algorithm only provides approximate solutions even in the considered convex scenario.

The kind of approach we follow is alternative to the main trend in the literature, in the sense that we do not consider any convex relaxation and work directly on the

Correspondence: erseghe@dei.unipd.it
Università di Padova, Dipartimento di Ingegneria dell'Informazione, via G.
Gradenigo 6/b, Padova, Italy

non-convex OPF problem. In this way, we can guarantee adherence to the original problem and develop an algorithm which is capable of identifying local minima. This idea was originally exploited in [15] where a distributed algorithm based upon ADMM was proposed. This algorithm provided undeniable evidence of the goodness of the intuition but had two major drawbacks. First, optimization for speed was cumbersome and required *centralized* coordination. Second, no guarantee on convergence was available, and in fact the algorithm often failed to converge. Although the convergence failure did not practically prevent the algorithm output for being usable, convergence is an issue that practically limits the algorithm speed.

In this paper, we wish to solve the above cited issues. To simplify system parameters and improve convergence speed, we remap the distributed problem in such a way to reveal the network *power flow*. In the ADMM formulation, the power flow variables are adequately weighted in order to force the algorithm to solve an approximate linear problem in the power flow variables in the first iterations (similarly to what happens with DC-OPF). The approximation is progressively abandoned in later iterations. This corresponds to the practical intuition that a linear power flow exchange problem provides a solution which is close to the optimum (some preliminary results on this aspect were recently presented at an international conference [16]). We also modify the plain ADMM algorithm and reinterpret it as a non-convex augmented Lagrangian method (see the work of Martinez and Birgin et al. [17,18]) where penalty parameters are constantly updated (increased) to always guarantee convergence. More specifically, a global convergence guarantee is available under the assumption that local solvers are efficient, in the sense that they can guarantee the identification of a (feasible) local minimum. This might not be an easy task in general, but it is a reasonable assumption when the number of local variables is controlled. Furthermore, a certificate of convergence to a local optimum is available when penalty parameters are bounded. The kind of coordination involved in this process is only local and therefore defines a fully distributed algorithm.

The rest of this paper is organized as follows. First, the reference OPF problem is presented and put in a networked form readily usable for obtaining a distributed algorithm. Then the distributed approach is discussed in abstract form and its convergence properties proved. Application to the specific OPF problem is then detailed, and the proposed distributed algorithm is finally tested in meaningful scenarios.

The OPF problem

We first introduce the OPF problem in its natural (centralized) formulation.

Standard formulation

Consider an electrical network of N nodes at steady state, where V_i , P_i , and Q_i represent, respectively, the local complex voltage, and the node's active and reactive powers. Assume that, at node i , a *local* cost is associated to active power production through a cost function $f_i(P_i)$. Assume that the electrical neighbors of node i are identified through the *neighbors* set \mathcal{N}_i , and that the line admittance Y_{ij} , $j \in \mathcal{N}_i$, is known for each physical connection. Then the standard OPF problem has the form

$$\begin{aligned} & \min \sum_{i \in \mathcal{N}} f_i(P_i) \\ & \text{w.r.t. } V_i \in \mathbb{C}, P_i, Q_i \in \mathbb{R}, i \in \mathcal{N} \\ & \text{s.t. } P_i + jQ_i = V_i \sum_{j \in \mathcal{N}_i} Y_{ij}^* V_j^* \\ & \underline{V}_i \leq |V_i| \leq \bar{V}_i \\ & \underline{P}_i \leq P_i \leq \bar{P}_i, \\ & \underline{Q}_i \leq Q_i \leq \bar{Q}_i \end{aligned} \quad (1)$$

where $\mathcal{N} = \{1, \dots, N\}$ is the nodes set. The first constraint in (1) refers to power flow equations (i.e., Kirchoff's laws). The remaining constraints are voltage and power constraint limitations, with \underline{V}_i , \bar{V}_i , \underline{P}_i , \bar{P}_i , \underline{Q}_i , \bar{Q}_i local upper and lower bounds.

For the ease of simplicity here we refer to a basic OPF problem, but additional constraints can be easily added to (1), e.g., power flow constraints on specific lines. Constraints referred to resources such as storage systems and renewable generators (dispatchable or not dispatchable) can be included by suitably selecting the cost factor f_i , by introducing proper corrections to the cost function, or by inserting a time variable. Discrete variables can be also included in the problem formulation (e.g., the tap changing of the transformers, or the cost to turn on/off a generator), in which case a mixed-integer programming solver will be needed. The results that follow are valid for all the above generalizations.

Region-based formulation

We now wish to fully capture the network relations in (1), in such a way to be used in a distributed implementation. The idea is to *partition* the network in R regions, where the sets \mathcal{R}_k , $k = 1, \dots, R$, identify nodes belonging to region k . We have

$$\mathcal{N} = \bigcup_{k=1}^R \mathcal{R}_k, \quad \mathcal{R}_k \cap \mathcal{R}_h = \emptyset, \forall k \neq h. \quad (2)$$

Because of power flow equations in (1), the voltages of interest in region k are those belonging to set

$$\mathcal{V}_k = \bigcup_{i \in \mathcal{R}_k} \mathcal{N}_i \quad (3)$$

where \mathcal{N}_i identify the neighbors of node i . Note that set \mathcal{V}_k includes set \mathcal{R}_k as a subset, as well as all those nodes which belong to neighbor regions and which have a direct connection (edge) with one of the nodes of \mathcal{R}_k . Accordingly, we identify the *local* voltage vectors \mathbf{x}_k with entries $x_{k,\ell}$ by

$$\mathbf{x}_k = [x_{k,\ell}]_{\ell \in \mathcal{V}_k}, \quad x_{k,\ell} = V_\ell \quad (4)$$

and the corresponding constraint region

$$\mathcal{X}_k = \left\{ \begin{array}{l} V_\ell \leq |x_{k,\ell}| \leq \bar{V}_\ell, \forall \ell \in \mathcal{V}_k, \\ \underline{P}_i \leq P_i \leq \bar{P}_i, \\ \underline{Q}_i \leq Q_i \leq \bar{Q}_i, \\ P_i + jQ_i = x_{k,i} \sum_{j \in \mathcal{N}_i} Y_{ij}^* x_{k,j}^*, \forall i \in \mathcal{R}_k \end{array} \right\} \quad (5)$$

collecting voltage constraints, active and reactive power constraints, and power flow constraints, and to which we may add any additional constraint of interest. Regions \mathcal{X}_k are deliberately chosen to be compact (closed and bounded) in order to strengthen later derivations and results.

Hence, a region-based equivalent formalization for (1) corresponds to the non-convex problem

$$\begin{aligned} & \min \sum_{k \in \mathcal{R}} F_k(\mathbf{x}_k) \\ & \text{w.r.t. } \mathbf{x}_k \in \mathcal{X}_k, k \in \mathcal{R} \\ & \text{s.t. } x_{k,\ell} = x_{h,\ell}, \forall \ell \in \mathcal{V}_k \cap \mathcal{V}_h, k, h \in \mathcal{R} \end{aligned} \quad (6)$$

where $\mathcal{R} = \{1, \dots, R\}$, function

$$F_k(\mathbf{x}_k) = \sum_{\ell \in \mathcal{R}_k} f_\ell(P_\ell) \quad (7)$$

collects local cost functions, and where the constraint is forcing equivalence between duplicated (voltage) variables in vectors \mathbf{x}_k .

Capturing the power flow

The formalization given in (6), although correct, is somehow unsatisfactory in terms of the slow convergence speed involved with its distributed implementation, and in terms of the difficulty in optimizing its system parameters

(see [15]). The key point is that we are not using any electrical intuition that could help the distributed processing. The intuition we use is illustrated in Figure 1.

The idea with Figure 1 is the following. Consider two neighboring regions k , and h , and edge (i, j) connecting the two regions, i.e., with $i \in \mathcal{R}_k$ and $j \in \mathcal{R}_h$. It also is $\{i, j\} \subset \mathcal{V}_k$ and $\subset \mathcal{V}_h$. Then, equivalence between the local variables can be written in the form

$$\begin{aligned} x_{k,i} &= x_{h,i} \\ x_{k,j} &= x_{h,j} \end{aligned} \quad (8)$$

which is equivalent to the constraint in (6). However, equivalence can be also written in the form

$$\begin{aligned} x_{k,i} - x_{k,j} &= x_{h,i} - x_{h,j} \\ x_{k,i} + x_{k,j} &= x_{h,i} + x_{h,j} \end{aligned} \quad (9)$$

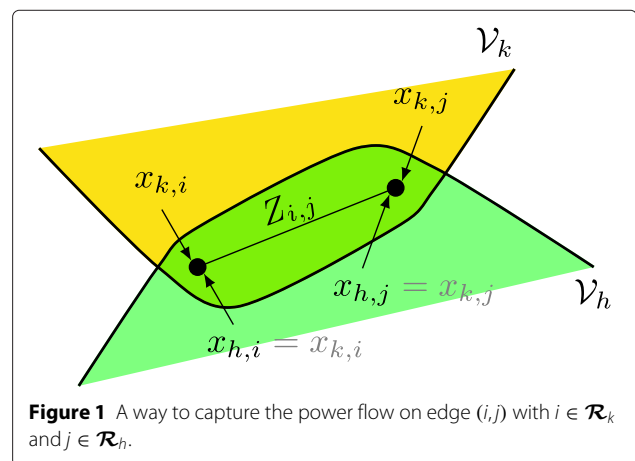
where the first equivalence captures the *power flow*, since the power flowing through line (i, j) is of the form $Z_{ij}|V_i - V_j|^2$, i.e., it only depends on voltage differences as from the first of (9).

The corresponding formulation for the OPF problem can then be compactly written by using sets

$$\mathcal{O}_k = \left\{ (i, j) \mid i \in \mathcal{R}_k, j \in \mathcal{N}_i \cap (\mathcal{V}_k \setminus \mathcal{R}_k) \right\} \quad (10)$$

collecting in region k those edges connecting a node of \mathcal{R}_k to a node in a neighbor region. By further introducing two auxiliary variables \mathbf{z}^- and \mathbf{z}^+ belonging to the linear spaces

$$\begin{aligned} \mathcal{Z}^- &= \{ \mathbf{z}^- \mid z_{ij}^- = -z_{ji}^-, \forall (i, j) \in \mathcal{O}_k, k \in \mathcal{R} \} \\ \mathcal{Z}^+ &= \{ \mathbf{z}^+ \mid z_{ij}^+ = z_{ji}^+, \forall (i, j) \in \mathcal{O}_k, k \in \mathcal{R} \} \end{aligned} \quad (11)$$



the OPF problem becomes

$$\begin{aligned}
 & \min \sum_{k \in \mathcal{R}} F_k(\mathbf{x}_k) \\
 & \text{w.r.t. } \mathbf{x}_k \in \mathcal{X}_k, k \in \mathcal{R} \\
 & \quad \mathbf{z}^- \in \mathcal{Z}^-, \mathbf{z}^+ \in \mathcal{Z}^+ \\
 & \text{s.t. } \rho(x_{k,i} - x_{k,j}) = z_{i,j}^-, \\
 & \quad \zeta(x_{k,i} + x_{k,j}) = z_{i,j}^+, \forall (i,j) \in \mathcal{O}_k, k \in \mathcal{R}
 \end{aligned} \tag{12}$$

where two positive constants ρ, ζ are used to differently weigh the power flow constraint on \mathbf{z}^- (providing convergence on an approximate linear problem on power flow variables) from the full equivalence constraint on \mathbf{z}^+ . The linear constraints in (12) can be also expressed in the compact matrix notation

$$\mathbf{z}_k = \begin{bmatrix} z_{i,j}^- \\ z_{i,j}^+ \end{bmatrix}_{(i,j) \in \mathcal{O}_k} = \mathbf{A}_k \mathbf{x}_k \tag{13}$$

where \mathbf{A}_k is a sparse matrix of size $2|\mathcal{O}_k| \times |\mathcal{V}_k|$. In the typical case of large regions having a few connections with neighbors it is $|\mathcal{O}_k| \ll |\mathcal{V}_k|$.

The distributed approach

We now introduce the distributed algorithm in a general and abstract form, in order to assess its properties and capture its structure with a compact notation.

Reference optimization problem

The kind of problem we wish to solve in (12) is a non-convex problem of the form

$$\begin{aligned}
 & \min F(\mathbf{x}) \\
 & \text{w.r.t. } \mathbf{x} \in \mathcal{X}, \mathbf{z} \in \mathcal{Z} \\
 & \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{z}
 \end{aligned} \tag{14}$$

where $\mathbf{x} = [\mathbf{x}_k]_{k \in \mathcal{R}}$ collects all variables, $\mathbf{z} = [\mathbf{z}_k]_{k \in \mathcal{R}}$ collects all auxiliary variables, $F(\mathbf{x}) = \sum_{k \in \mathcal{R}} F_k(\mathbf{x}_k)$ is separable, $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_R$ is a Cartesian product, set $\mathcal{Z} = \mathcal{Z}^- \times \mathcal{Z}^+$ is a linear space with associated projector \mathbf{L}_Z , and $\mathbf{A} = \text{diag}(\mathbf{A}_1, \dots, \mathbf{A}_R)$ has a block diagonal form. The results given in the following further consider \mathcal{X} bounded (as we already assumed), and $F(\mathbf{x})$ continuous. We finally assume that (14) has a solution.

The smoothness of functions involved with the OPF problem ensure that a one-to-one relation exists between local minima of problem (14) and the corresponding Karush Kuhn Tucker (KKT) conditions. We have (e.g., see [19])

Theorem 1. (KKT stationary points)

The KKT stationary point conditions associated with the primal problem (14) are given by

$$\begin{aligned}
 & \mathbf{0} \in \partial F(\mathbf{x}) + \partial \eta_{\mathcal{X}}(\mathbf{x}) + \mathbf{A}^T \boldsymbol{\lambda} \\
 & \mathbf{A}\mathbf{x} = \mathbf{z} \\
 & \mathbf{x} \in \mathcal{X}, \mathbf{z} \in \mathcal{Z}, \boldsymbol{\lambda} \perp \mathcal{Z}
 \end{aligned} \tag{15}$$

where ∂ is the proximal sub-gradient operator, and where $\eta_{\mathcal{A}}$ is the indicator function of set \mathcal{A} , with $\eta_{\mathcal{A}}(\mathbf{a}) = 0$ if $\mathbf{a} \in \mathcal{A}$ and $+\infty$ if $\mathbf{a} \notin \mathcal{A}$. Conditions (15) identify the local minima of (14). \square

Augmented Lagrangian formalization

No global minimum assurance is given in the present context, since the Lagrangian associated with problem (14) may suffer of a primal-dual gap. A remedy in this respect is to use a Powell Hestenes Rockafellar (PHR) augmented Lagrangian formulation. The augmented Lagrangian associated with problem (14) can be written in the form

$$\begin{aligned}
 L(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}) &= F(\mathbf{x}) + \eta_{\mathcal{X}}(\mathbf{x}) + \eta_{\mathcal{Z}}(\mathbf{z}) \\
 &+ \boldsymbol{\lambda}^T (\mathbf{A}\mathbf{x} - \mathbf{z}) + \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{z}\|_{\boldsymbol{\epsilon}}^2
 \end{aligned} \tag{16}$$

where $\|\mathbf{x}\|_{\boldsymbol{\epsilon}}^2 = \mathbf{x}^T \text{diag}(\boldsymbol{\epsilon})\mathbf{x}$ is a scaled norm, and where the entries of $\boldsymbol{\epsilon}$ are strictly positive. In (16), the couple (\mathbf{x}, \mathbf{z}) plays the role of primal variables, while $(\boldsymbol{\lambda}, \boldsymbol{\epsilon})$ play the role of dual variables (Lagrange multipliers). The dual function associated with (16) is

$$D(\boldsymbol{\lambda}, \boldsymbol{\epsilon}) = \min_{\mathbf{x}, \mathbf{z}} L(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}) . \tag{17}$$

The PHR augmented Lagrangian of (16) is well defined, in the sense that it ensures the typical properties of ordinary Lagrangians of convex functions, i.e., the zero duality gap property and the applicability of a saddle point theorem. The result is given in ([20], Theorem 11.59). Incidentally, we are using a vector of weighting factors $\boldsymbol{\epsilon}$ instead of a unique multiplication by scalar factor ϵ . This, however, does not modify derivation nor the final result.

Theorem 2. (Rockafellar-Wets)

1. Zero duality gap Let $(\mathbf{x}^*, \mathbf{z}^*)$ be a solution to the primal problem (14), and let $(\boldsymbol{\lambda}^*, \boldsymbol{\epsilon}^*)$ be any maximizer of the dual function (17). The corresponding duality gap is zero, that is, we have

$$F(\mathbf{x}^*) = D(\boldsymbol{\lambda}^*, \boldsymbol{\epsilon}^*) . \tag{18}$$

2. Saddle point The solutions in 1 identify a saddle point of PHR augmented Lagrangian (16), that is

$$\begin{aligned} (\mathbf{x}^*, \mathbf{z}^*) &\in \operatorname{argmin}_{\mathbf{x}, \mathbf{z}} L(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}^*, \boldsymbol{\epsilon}^*) \\ (\boldsymbol{\lambda}^*, \boldsymbol{\epsilon}^*) &\in \operatorname{argmax}_{\boldsymbol{\lambda}, \boldsymbol{\epsilon} \geq \mathbf{0}} L(\mathbf{x}^*, \mathbf{z}^*, \boldsymbol{\lambda}, \boldsymbol{\epsilon}). \end{aligned} \quad (19)$$

Conversely, any saddle point (19) identifies a primal and dual solution, as from 1. \square

In this context, the search for an optimum point can be turned into the search for a saddle point of the PHR augmented Lagrangian, which is in general more effective in terms of efficiency and speed. However, since only a local optimization point may be available for the first of (19) (because of non-convexity), then only *local* saddle points can be practically identified. It is then interesting to observe the following result, which is a straightforward consequence of the fact that local minima/maxima conditions of (19) correspond to KKT stationary point conditions (15), as the reader can easily verify.

Theorem 3. *There exists a one-to-one correspondence between local minima of the original problem (14), KKT stationary points (15), and local saddle points of the PHR augmented Lagrangian in (19).* \square

As a consequence, the search for local minima can be mapped into a search for local saddle points of the augmented Lagrangian.

Alternating direction search for a local saddle point

The search for a local saddle point can be dealt with by using the method of [17] (see also [18]). In our context, the method can be mapped into an alternating direction algorithm of the form

$$\begin{aligned} \mathbf{x}_{t+1} &\in \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} L(\mathbf{x}, \mathbf{z}_t, \boldsymbol{\lambda}_t, \boldsymbol{\epsilon}_t) \\ \mathbf{z}_{t+1} &\in \operatorname{argmin}_{\mathbf{z} \in \mathcal{Z}} L(\mathbf{x}_{t+1}, \mathbf{z}, \boldsymbol{\lambda}_t, \boldsymbol{\epsilon}_t) \\ \boldsymbol{\lambda}_{t+1} &= \boldsymbol{\lambda}_t + \mathbf{E}_t(\mathbf{A}\mathbf{x}_{t+1} - \mathbf{z}_{t+1}) \end{aligned} \quad (20)$$

where $\mathbf{E}_t = \operatorname{diag}(\boldsymbol{\epsilon}_t)$, and where $\boldsymbol{\epsilon}_t$ is suitably updated at each cycle by guaranteeing $\boldsymbol{\epsilon}_{t+1} \geq \boldsymbol{\epsilon}_t$. Note that, differently from [17], and similarly to what we have in ADMM, an independent update is used for \mathbf{x}_t and \mathbf{z}_t . In turn, differently from ADMM, the weighting parameters $\boldsymbol{\epsilon}_t$ are updated in order to ensure convergence of the process in a non-convex scenario.

Throughout the process, we assume that the commutation property

$$\mathbf{L}_{\mathcal{Z}}\mathbf{E}_t = \mathbf{E}_t\mathbf{L}_{\mathcal{Z}} \quad (21)$$

holds, which corresponds to the request

$$\epsilon_{k,i,j} = \epsilon_{h,j,i}, \quad (i, j) \in \mathcal{O}_k, j \in \mathcal{R}_h, k, h \in \mathcal{R}. \quad (22)$$

We also assume that

$$\boldsymbol{\lambda}_0 \perp \mathcal{Z}. \quad (23)$$

These are light hypotheses guaranteeing that (20) simplifies to updates

$$\begin{aligned} \mathbf{x}_{t+1} &\in \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} F(\mathbf{x}) + \frac{1}{2} \|\mathbf{A}\mathbf{x} - (\mathbf{z}_t - \mathbf{E}_t^{-1}\boldsymbol{\lambda}_t)\|_{\boldsymbol{\epsilon}_t}^2 \\ \mathbf{z}_{t+1} &= \mathbf{L}_{\mathcal{Z}}\mathbf{A}\mathbf{x}_{t+1} \\ \boldsymbol{\lambda}_{t+1} &= \boldsymbol{\lambda}_t + \mathbf{E}_t(\mathbf{A}\mathbf{x}_{t+1} - \mathbf{z}_{t+1}) \end{aligned} \quad (24)$$

and we also have

$$\mathbf{z}_{t+1} \in \mathcal{Z}, \quad \boldsymbol{\lambda}_{t+1} \perp \mathcal{Z} \quad (25)$$

so that the third line in KKT conditions (15) is satisfied throughout the iterative process. Note that the update of \mathbf{x}_t in the first of (24) corresponds to the parallel of a number of local updates because F is separable, and \mathcal{X} is a Cartesian product. In addition, since the full minimum for the first of (24) may be not available, we relax the result by assuming that a local minimum is achieved and that the target function in this local minimum \mathbf{x}_{t+1} is smaller than or equal to the function value in \mathbf{x}_t . Therefore, a reliability assumption on the local solver is required. Although this might be in general a strong request (e.g., see [21]), especially when the local constraints identify a very small feasibility region, we expect it to be reasonably met when the number of local variables is not too large (i.e., for small regions).

Interestingly, given the fact that \mathcal{X} is bounded, then both sequences $\{\mathbf{x}_t\}$ and $\{\mathbf{z}_t\}$ are bounded. This may not be the case for $\{\boldsymbol{\lambda}_t\}$, but it is convenient to force this property by assuming

$$\boldsymbol{\lambda}_{t+1} = \mathcal{P}[\boldsymbol{\lambda}_t + \mathbf{E}_t(\mathbf{A}\mathbf{x}_{t+1} - \mathbf{z}_{t+1})] \quad (26)$$

with $\mathcal{P}[\boldsymbol{\lambda}] = \max(\boldsymbol{\lambda}_{\min}, \min(\boldsymbol{\lambda}, \boldsymbol{\lambda}_{\max}))$ a projection onto a compact box. The reason for this action will become clearer later on in the proof of Theorem 5.

Concerning penalty parameters $\boldsymbol{\epsilon}_t$, in the centralized fashion of [17] the update criterion on $\boldsymbol{\epsilon}_t$ is of the form

$$\boldsymbol{\epsilon}_{t+1} = \begin{cases} \boldsymbol{\epsilon}_t & \text{if } \Gamma_{t+1} \leq \theta \Gamma_t \\ \tau \boldsymbol{\epsilon}_t & \text{otherwise} \end{cases} \quad (27)$$

with constants $0 < \theta < 1$ and $\tau > 1$, and with

$$\Gamma_t = \|\mathbf{A}\mathbf{x}_t - \mathbf{z}_t\|_{\infty} \quad (28)$$

a measure of the primal gap (in infinity norm), in such a way to increase the penalty only if the primal gap is not decreasing sufficiently. The criterion can be also made local. The approach we propose is the following. We first check the primal gap decrease in region k via

$$\check{\epsilon}_{k,t+1} = \begin{cases} \|\epsilon_{k,t}\|_{\infty} \mathbf{1} & \text{if } \Gamma_{k,t+1} \leq \theta \Gamma_{k,t} \\ \tau \|\epsilon_{k,t}\|_{\infty} \mathbf{1} & \text{otherwise} \end{cases} \quad (29)$$

with $\mathbf{1}$ the all-ones vector, and with

$$\Gamma_{k,t} = \|\mathbf{A}_k \mathbf{x}_{k,t} - \mathbf{z}_{k,t}\|_{\infty} \quad (30)$$

the local gap. We then select the smallest $\epsilon_{t+1} \geq \check{\epsilon}_{t+1}$ satisfying (29), which in our context implies

$$\epsilon_{k,i,j,t+1} = \max(\check{\epsilon}_{k,i,j,t+1}, \check{\epsilon}_{h,j,i,t+1}) \quad (31)$$

where $(i, j) \in \mathcal{O}_k, j \in \mathcal{R}_h, k, h \in \mathcal{R}$. This approach only requires local message exchanges. With this definition, the update is such that if one value of $\epsilon_{k,t}$ grows to ∞ , then all the values in the network do so, as it is for the centralized counterpart (27).

The proposed solution is summarized in Algorithm 1.

Algorithm 1: Alternating direction search method

- 1 Update variable \mathbf{x}_{t+1} using the first of (24). When a global minimum guarantee is not available, a local minimum must be identified, with the guarantee that the target function is decreased with respect to its value at \mathbf{x}_t .
 - 2 Update auxiliary variables $\mathbf{z}_{t+1} = \mathbf{L}_{\mathcal{Z}} \mathbf{A} \mathbf{x}_{t+1}$.
 - 3 Update the Lagrange multipliers $\boldsymbol{\lambda}_{t+1}$ using (26).
 - 4 Update the penalty parameters ϵ_{t+1} using (27) or (29)-(31).
-

Convergence guarantees

The important characteristic of Algorithm 1 is that, in the given scenario, it provides a distributed solution. The main difference with the inspiring technique of [17] lays in the use of an alternating search with respect to \mathbf{x} and \mathbf{z} (versus the joint minimum search on (\mathbf{x}, \mathbf{z})), this being the key point for obtaining a *distributed* algorithm. Nevertheless, the algorithm always converges (despite the non-convex scenario), and convergence guarantees essentially equivalent to those of [17] can be derived.

We separately treat the case where the penalty constant parameters are bounded and the case where they are unbounded. For bounded parameters we have the following result.

Theorem 4. (Bounded penalties)

Consider Algorithm 1, and assume that the sequence of penalty parameters $\{\epsilon_t\}$ is bounded. We have:

1. Sequences $\{\mathbf{z}_t\}$ and $\{\boldsymbol{\lambda}_t\}$ converge to finite values, \mathbf{z}^* and $\boldsymbol{\lambda}^*$, respectively.
2. There exists a finite limit point (accumulation point) for the sequence $\{\mathbf{x}_t\}$, and if $\mathbf{A}^T \mathbf{A}$ is invertible then sequence $\{\mathbf{x}_t\}$ is further guaranteed to converge to a finite value \mathbf{x}^* .
3. The triplets $(\mathbf{x}^*, \mathbf{z}^*, \boldsymbol{\lambda}^*)$, with \mathbf{x}^* any limit point of $\{\mathbf{x}_t\}$, satisfy the KKT conditions of (15), hence all limit points \mathbf{x}^* identify a local minimum to the original problem. Even more, in the limit $t \rightarrow \infty$ any triplet $(\mathbf{x}_t, \mathbf{z}_t, \boldsymbol{\lambda}_t)$ satisfies the KKT stationarity conditions, i.e., identifies a local minimum and satisfies the constraint $\mathbf{A} \mathbf{x}_t = \mathbf{z}_t$. □

Proof of Theorem 4. Consider that the sequence of penalty parameters $\{\epsilon_t\}$ is bounded, to have $\epsilon_t = \epsilon_{\infty}$ for $t \geq t_0$. For both (27) and (29), we have that $\Gamma_{t+1} \leq \theta \Gamma_t$ for $t > t_0$, and therefore λ_t is bounded and converges to a finite value λ_{∞} (also in case the projection (26) is limiting the value to its maximum).

Now, by exploiting equivalence $\mathbf{z}_t = \mathbf{L}_{\mathcal{Z}} \mathbf{A} \mathbf{x}_t$, we rewrite the update of \mathbf{x}_t in (24) in the form

$$\mathbf{x}_{t+1} \in \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmin}} F(\mathbf{x}) + \frac{1}{2} \|\mathbf{I} - \mathbf{L}_{\mathcal{Z}}\mathbf{A}\mathbf{x}\|_{\epsilon_t}^2 + \frac{1}{2} \|\mathbf{L}_{\mathcal{Z}}\mathbf{A}(\mathbf{x} - \mathbf{x}_t)\|_{\epsilon_t}^2 + \boldsymbol{\lambda}_t^T \mathbf{A} \mathbf{x}. \quad (32)$$

By then using the shorthand notation

$$g_t = F(\mathbf{x}_t) + \eta_{\mathcal{X}}(\mathbf{x}_t) + \frac{1}{2} \|\mathbf{A} \mathbf{x}_t - \mathbf{z}_t\|_{\epsilon_{\infty}}^2 + \boldsymbol{\lambda}_{\infty}^T \mathbf{A} \mathbf{x}_t$$

$$\zeta_t = (\boldsymbol{\lambda}_t - \boldsymbol{\lambda}_{\infty})^T \mathbf{A}(\mathbf{x}_t - \mathbf{x}_{t+1}),$$

and $\Delta g_t = g_{t+1} - g_t$, from (32) we have

$$\Delta g_t + \frac{1}{2} \|\mathbf{z}_{t+1} - \mathbf{z}_t\|_{\epsilon_{\infty}}^2 \leq \zeta_t \leq |\zeta_t|, \quad t > t_0 \quad (33)$$

which implies $\Delta g_t \leq |\zeta_t|$ for $t > t_0$. By noting that $\|\mathbf{A}(\mathbf{x}_t - \mathbf{x}_{t+1})\|$ is bounded because \mathcal{X} is assumed bounded, and by recalling that $\lim_{t \rightarrow \infty} \boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{\infty}$, then it also is $\lim_{t \rightarrow \infty} |\zeta_t| = 0$. This is sufficient to guarantee that Δg_t converges to 0 for $t \rightarrow \infty$, which can be proved by contradiction. Specifically, if Δg_t does not converge to 0 then there exists an infinite sequence for which $|\Delta g_t| \geq \epsilon > 0$. Moreover, since $\Delta g_t \leq |\zeta_t|$, where the right value can be made arbitrarily small for large t , there

also exists an infinite sequence for which $\Delta g_t \leq -\epsilon$. By denoting the sequence as $\mathcal{S}_\epsilon \subset (t_0, \infty)$, this would imply

$$g_\infty - g_{t_0} = \sum_{t \notin \mathcal{S}_\epsilon} \Delta g_t + \sum_{t \in \mathcal{S}_\epsilon} \Delta g_t \leq \sum_{t \notin \mathcal{S}_\epsilon} |\zeta_t| - \sum_{t \in \mathcal{S}_\epsilon} \epsilon .$$

Since $|\zeta_t|$ is guaranteed to be exponentially decreasing because of the assumption $\Gamma_{t+1} \leq \theta \Gamma_t$, the above implies $g_\infty = -\infty$, hence a contradiction. Therefore, g_t converges to a finite value, and, as a consequence of (33), the weighted norm $\|\mathbf{z}_{t+1} - \mathbf{z}_t\|_{\epsilon_\infty}^2$ converges to 0, i.e., \mathbf{z}_t converges to a finite value too. These results justify points 1 and 2.

To conclude with point 3, since \mathbf{x}_{t+1} is assumed a local minimum, from (32) we also have, for $t > t_0$,

$$\begin{aligned} \mathbf{0} \in & \partial F(\mathbf{x}_{t+1}) + \partial \eta_{\mathcal{X}}(\mathbf{x}_{t+1}) + \mathbf{A}^T \boldsymbol{\lambda}_\infty \\ & + \mathbf{A}^T \mathbf{E}_\infty (\mathbf{I} - \mathbf{L}_{\mathcal{Z}}) \mathbf{A} \mathbf{x}_{t+1} \\ & + \mathbf{A}^T \mathbf{E}_\infty (\mathbf{z}_{t+1} - \mathbf{z}_t) + \mathbf{A}^T (\boldsymbol{\lambda}_t - \boldsymbol{\lambda}_\infty) \end{aligned}$$

and since the values on the second and third lines tend to $\mathbf{0}$ in the limit, then in the limit, the KKT stationary point conditions (15) are satisfied.

As a consequence, bounded penalty parameters guarantee a convergence of the algorithm to a KKT stationary point, i.e., they imply the identification of a local minimum. Note that the result is sufficiently strong also in the case where $\mathbf{A}^T \mathbf{A}$ is not invertible (see second part of point 3). This is an important property since the invertibility of $\mathbf{A}^T \mathbf{A}$ is only ensured for a single-node regions choice $\mathcal{R}_k = \{k\}$.

The result for unbounded parameters assumes that the ill conditioning associated with very large/infinite values is adequately solved, e.g., by locally normalizing the minimization in (32) by the maximum penalty value $\|\boldsymbol{\epsilon}_{k,t}\|_\infty$. We have

Theorem 5 (Unbounded penalties). *Consider Algorithm 1, and assume that the sequence of penalty parameters $\{\boldsymbol{\epsilon}_t\}$ is unbounded. We have:*

1. Sequence $\{\mathbf{z}_t\}$ converges to a finite value, \mathbf{z}^* .
2. There exists a finite limit point for the sequence $\{\mathbf{x}_t\}$, and if $\mathbf{A}^T \mathbf{A}$ is invertible then sequence $\{\mathbf{x}_t\}$ is ensured to converge to a finite value \mathbf{x}^* . \square

Proof. *The results in the proof of Theorem 4 can be applied by suitably (locally) normalizing parameters. The kind of replacements we use are*

$$\begin{aligned} \boldsymbol{\epsilon}_t & \implies \tilde{\boldsymbol{\epsilon}}_t = \left[\frac{\boldsymbol{\epsilon}_{k,t}}{\|\boldsymbol{\epsilon}_{k,t}\|_\infty} \right]_{k=1,\dots,R} \\ F(\mathbf{x}) & \implies \tilde{F}(\mathbf{x}) = \sum_{k=1}^R \frac{F_k(\mathbf{x}_k)}{\|\boldsymbol{\epsilon}_{k,t}\|_\infty} \\ \boldsymbol{\lambda}_t & \implies \tilde{\boldsymbol{\lambda}}_t = \left[\frac{\boldsymbol{\lambda}_{k,t}}{\|\boldsymbol{\epsilon}_{k,t}\|_\infty} \right]_{k=1,\dots,R} \end{aligned}$$

which have the characteristic of providing bounded quantities. For both (27) and (29), all entries $\boldsymbol{\epsilon}_{k,t}$ are diverging by construction, hence $\tilde{\boldsymbol{\lambda}}_t$ is ensured to converge to $\mathbf{0}$ in the limit. Convergence is also guaranteed to be exponential, because of the presence of parameter $\tau > 1$ in the update of penalty parameters. These properties are fundamental and are ensured by use of projection (26). Furthermore, $\tilde{\boldsymbol{\epsilon}}_t$ is guaranteed to converge to the all ones vector $\mathbf{1}$. By then investigating the counterparts to g_t and ζ_t , namely,

$$\begin{aligned} \tilde{g}_t &= \tilde{F}(\mathbf{x}_t) + \eta_{\tilde{\mathcal{X}}}(\mathbf{x}_t) + \frac{1}{2} \|\mathbf{A} \mathbf{x}_t - \mathbf{z}_t\|_{\tilde{\boldsymbol{\epsilon}}_t}^2 \\ \tilde{\zeta}_t &= \tilde{\boldsymbol{\lambda}}_t^T \mathbf{A} (\mathbf{x}_t - \mathbf{x}_{t+1}) \end{aligned}$$

we still verify that properties $\lim_{t \rightarrow \infty} |\tilde{\zeta}_t| = 0$ and

$$\Delta \tilde{g}_t + \frac{1}{2} \|\mathbf{z}_{t+1} - \mathbf{z}_t\|_{\tilde{\boldsymbol{\epsilon}}_t}^2 \leq \tilde{\zeta}_t \leq |\tilde{\zeta}_t| \tag{34}$$

hold, and we also have that $\Delta \tilde{g}_t$ converges to 0. Hence \tilde{g}_t converges to a finite value, so that there exist limit points for the sequence $\{\mathbf{x}_t\}$. From (34) we also find that \mathbf{z}_t converges to a finite value. This proves the theorem.

Note that Theorem 5, although being able to prove convergence of both sequences $\{\mathbf{x}_t\}$ and $\{\mathbf{z}_t\}$, cannot guarantee that the limit solution is feasible, i.e., it satisfies $\mathbf{A} \mathbf{x}_t = \mathbf{z}_t$. As a matter of fact, in the limit, the minimization in (32) assumes the (approximate) form

$$\mathbf{x}_{t+1} \in \operatorname{argmax}_{\mathbf{x} \in \mathcal{X}} \|\mathbf{I} - \mathbf{L}_{\mathcal{Z}}\mathbf{A}\mathbf{x}\|^2 + \|\mathbf{L}_{\mathcal{Z}}\mathbf{A}(\mathbf{x} - \mathbf{x}_t)\|^2 \tag{35}$$

which corresponds to an iterative algorithm for performing a projection of \mathbf{x} onto the feasible space $\mathcal{X} \cap \{\mathbf{x} | \mathbf{A} \mathbf{x} = \mathbf{L}_{\mathcal{Z}} \mathbf{A} \mathbf{x}\}$, and in this context, the contribution $\|\mathbf{L}_{\mathcal{Z}} \mathbf{A} (\mathbf{x} - \mathbf{x}_t)\|^2$ plays the role of a proximity operator, forcing vicinity to the solution available from the previous step. Therefore, if the algorithm used to solve the local problem (32) is sufficiently powerful, then convergence to a feasible point is also ensured in the limit. This is the case, in practice, only for moderately non-convex scenarios.

The distributed OPF algorithm

The distributed OPF algorithm that we obtain by applying Algorithm 1 to problem (12) is summarized in Algorithm 2. The local penalty parameters update (29)-(31) is used.

Algorithm 2: Distributed OPF processing in region k
(t denotes the iteration number)

```

1 for  $t = 0$  to  $\infty$  do
2   if  $t = 0$  then
3     Initialize local voltages  $\mathbf{x}_{k,0}$ 
4   else
5     Update local voltages  $\mathbf{x}_{k,t}$  via
6       
$$\mathbf{x}_{k,t} \in \underset{\mathbf{x}_k \in \mathcal{X}_k}{\operatorname{argmin}} F_k(\mathbf{x}_k) + \frac{1}{2} \mathbf{x}_k^T \mathbf{D}_{k,t} \mathbf{x}_k + \mathbf{y}_{k,t}^T \mathbf{x}_k$$

7       where
8         
$$\mathbf{D}_{k,t} = \mathbf{A}_k^T \operatorname{diag}(\boldsymbol{\epsilon}_{k,t-1}) \mathbf{A}_k$$

9         
$$\mathbf{y}_{k,t} = \mathbf{A}_k^T (\operatorname{diag}(\boldsymbol{\epsilon}_{k,t-1}) \mathbf{z}_{k,t-1} - \boldsymbol{\lambda}_{k,t-1})$$

10        A local minimum must be identified, with the
11        guarantee that the target function is
12        decreased with respect to its value at  $\mathbf{x}_{k,t-1}$ .
13   end if
14   Prepare messages  $\mathbf{m}_{k,t} = \mathbf{A}_k \mathbf{x}_{k,t}$ 
15    $\Rightarrow$  Broadcast messages  $\mathbf{m}_{k,t}$  to neighbor regions
16    $\Leftarrow$  Receive messages  $\mathbf{m}_{h,t}$  from neighbor regions  $h$ 
17   Update auxiliary variables via
18     
$$\mathbf{z}_{k,i,j,t}^- = \frac{1}{2} (\mathbf{m}_{k,i,j,t}^+ - \mathbf{m}_{h,j,i,t}^-)$$

19     
$$\mathbf{z}_{k,i,j,t}^+ = \frac{1}{2} (\mathbf{m}_{k,i,j,t}^+ + \mathbf{m}_{h,j,i,t}^-)$$

20   if  $t = 0$  then
21     Initialize Lagrange multipliers  $\boldsymbol{\lambda}_{k,0}$ . If no a
22     priori information is available, then set them
23     to  $\mathbf{0}$ .
24     Initialize the local gap  $\Gamma_{k,0} = \infty$ 
25     Initialize penalty parameters  $\boldsymbol{\epsilon}_{k,0} = \mathbf{1} \xi$ 
26   else
27     Update Lagrange multipliers via
28       
$$\boldsymbol{\lambda}_{k,t} = \mathcal{P}[\boldsymbol{\lambda}_{k,t-1} + \operatorname{diag}(\boldsymbol{\epsilon}_{k,t-1}) (\mathbf{A}_k \mathbf{x}_{k,t} - \mathbf{z}_{k,t})]$$

29     where  $\mathcal{P}$  is a projection onto box  $[\boldsymbol{\lambda}_{\min}, \boldsymbol{\lambda}_{\max}]$ .
30     Update the local gap  $\Gamma_{k,t} = \|\mathbf{A}_k \mathbf{x}_{k,t} - \mathbf{z}_{k,t}\|_\infty$ 
31     Locally update penalty parameters
32       
$$\boldsymbol{\epsilon}_{k,t} = \begin{cases} \|\boldsymbol{\epsilon}_{k,t-1}\|_\infty \mathbf{1} & \text{if } \Gamma_{k,t} \leq \theta \Gamma_{k,t-1} \\ \tau \|\boldsymbol{\epsilon}_{k,t-1}\|_\infty \mathbf{1} & \text{otherwise} \end{cases}$$

33      $\Rightarrow$  Broadcast  $\boldsymbol{\epsilon}_{k,i,j,t}$  to neighbor regions
34      $\Leftarrow$  Receive  $\boldsymbol{\epsilon}_{h,j,i,t}$  from neighbor region  $h$ 
35     Correct local penalty parameters via
36       
$$\boldsymbol{\epsilon}_{k,i,j,t+1} = \max(\boldsymbol{\epsilon}_{k,i,j,t+1}, \boldsymbol{\epsilon}_{h,j,i,t+1})$$

37   end if
38 end for

```

Note that two local message exchanges (denoted with arrows) are required in lines 10 to 11 and lines 22 to 23 to exchange, respectively, the updated values $\mathbf{x}_{k,t}$ (in order to update auxiliary variables) and the tentative penalty parameters updates $\boldsymbol{\epsilon}_{k,t}$ (in order to make sure that the final update satisfies (21)). In principle, a single message exchange could be obtained by postponing the penalty parameters correction of line 24 after the auxiliary variable update in line 13, at the cost of some sub optimality in performance.

Overall, the local processing effort of Algorithm 2 is light. The algorithm complexity is determined by the update of \mathbf{x}_t in line 5, which corresponds to a region-based optimization problem, and which can be efficiently solved by state-of-the-art methods, e.g., interior point methods (IPMs). The remaining actions require a limited effort, especially in the standard case where a few connections are active with neighboring regions and auxiliary vectors are short (i.e., $|\mathcal{O}_k| \ll |\mathcal{V}_k|$).

We finally underline that five key parameters are used in Algorithm 2, and these need to be accurately set for good performance. We have:

1. Weighting constants ρ and ζ (they define matrices \mathbf{A}_k , see (12)-(13)). They should be chosen in such a way that $\rho \gg \zeta > 0$, in order to force the algorithm towards an approximate linear solution on power flow variables.
2. Initialization value for penalty parameters ξ . It should be set to a small value to guarantee a good algorithm outcome even when starting from a point very far from the optimum.
3. Penalty parameters update constants $0 < \theta < 1$ and $\tau > 1$. In order to avoid a rapid increasing behavior on penalty parameters, the constants should be set to values close to 1.

Performance assessment

The algorithm performance is tested using three different scenarios, namely: 1) the wide area network IEEE Power System Test Case Archive [22]; 2) the IEEE PES Distribution Test Feeder [23,24]; 3) a microgrid topology generated according to the model proposed in [25]. The networks in Scenarios 2) and 3) have a tree topology, while Scenario 1) involves networks with many loops where algorithm convergence may be an issue. All chosen scenarios are moderate sized networks, with moderate non-convexities, which constitute the applicability field of the proposed algorithm. Applicability to more complex networks with more severe non-convexities and a high number of loops (e.g., the Polish system models) requires use of some additional (quasi centralized) coordination between entities, and will be the subject of future investigation.

Description of the scenarios

A power losses minimization problem under voltage and power constraints is considered (i.e., $f_i(P_i) = P_i$), and the following settings are used in the various scenarios:

1. Networks sizes $N = 30, 57, 118$, and 300 are used. Constraints and load requests are set as from the MATPOWER distribution [26].
2. The $N = 123$ nodes network is used in single-phase fashion. The chosen settings are inspired by [6]. Load requests are set as given in the dataset, and generating capabilities ranges are added in the form $|Q_{G,i}| \leq 1.2|Q_{L,i}|$, and $0 \leq P_{G,i} \leq 30$ kW, where the subscript L stands for *load* and G for *generation*. Voltage regulation is applied with $0.94 \leq |V_i| \leq 1.06$.
3. A unique network is selected with $N = 120$. The network is generated as four joint small-world graphs with 30 nodes (to limit the depth of the graph) and rewiring probability $p = 0.4$ (see also details in [16]). Lines lengths have an exponential distribution with parameter $\mu = 65.86$ m and a minimum distance set to 10 m. The impedance value is chosen $2.9400 + j0.0861 \Omega/\text{km}$ (class 1, 10 mm^2 cables). Load requests are randomly generated with an uniform distribution in $[0, 3]$ kW, and with a uniform $\cos \phi$ with $\phi \in [-\frac{\pi}{8}, \frac{\pi}{8}]$. 20% of the nodes are given generation capabilities, randomly distributed in $[0, 10]$ kW for active power and $[-20, 20]$ kVAr for reactive power. Voltage regulation is applied in the range $0.9 \leq |V_i| \leq 1.1$.

Region partitioning

Region partitioning is a fundamental aspect for ensuring a good performance. Ideally, compact regions with very few outer connections guarantee limited complexity, accuracy of the solution, and controlled computational time. In the considered scenarios, region partitioning is chosen in such a way that a unique generator is available in each region, and the region further includes those loads which are electrically closer (in terms of line impedance) to the generator. Since this corresponds to an excessively fine partitioning in Scenario 2), for the IEEE feeder, the region choice is made in such a way that a local controller is placed at each network bifurcation point, and the associated region corresponds to all those nodes which are electrically closer to it (in terms of line impedance).

Simulation tools

The local optimization problem (see line 5 of Algorithm 2, or see the first of (24)) is solved by using IPOPT [27], an efficient IPM solver which allows a MatLab interface. Although a true optimality guarantee is not available, IPM methods are known to perform very well for OPF kind of problems. MUMPS linear solver is used within IPOPT,

and the *warm start* option is used in such a way to start the local minimization process using the solution available from the previous iteration (this reduces computational times). The code is run on a MacBook Air and is written in MatLab [28].

Convergence test in the considered scenarios

A test on the behavior of Algorithm 2 in the three different scenarios using the parameters of Table 1 is illustrated in Figure 2. The starting point is chosen to be the all-ones vector $x_{k,0} = \mathbf{1}$, and Lagrange multipliers are initially set to zero, $\lambda_{k,0} = \mathbf{0}$. This corresponds to the unavailability of any *a priori* information on both position and Lagrange multipliers and is therefore a worst case scenario. Iterations are stopped (and convergence is declared) when the primal gap $\|Ax_t - z_t\|_\infty$ (infinity norm) reaches 10^{-4} . The maximum values for Lagrange multipliers are set to $\lambda_{\max} = 10^3 \cdot \mathbf{1}$, $\lambda_{\min} = -10^3 \cdot \mathbf{1}$.

For the three scenarios considered, Figure 2 shows in the first column the voltages V_i (amplitude and phase diagram) at convergence, together with the active voltage constraints. Observe that all voltage limitations are met.

The second column of Figure 2 shows the behavior of the primal gap in norm 2 and norm ∞ as a function of the iteration number t . Although the curves are not strictly decreasing, they are clearly diminishing to zero-gap value. The penalty parameters update, illustrated in the third column of Figure 2, shows the ability of (29)-(31) of keeping a small gap between maximum and minimum values of ϵ_t . The fact that the parameters are always increasing is due to the sub optimality of the distributed criterion with respect to the centralized criterion (27) which would be more effective in limiting the increase of penalty parameters. Nevertheless, the algorithm converges to points very close to the optimum (see Table 1) despite the very badly chosen initial point. In this respect, the local IPM solvers are fully capable of resolving the limit problem (35) and hence guarantee convergence to a feasible point. Note that the slower convergence is experienced with Scenario 2), i.e., the IEEE feeder with $N = 123$. This is due to the fact that this is the network with highest depth due to its radial structure. This makes the distributed process particularly challenging since agreement must be obtained between regions that are very far one from the other.

Finally, in the fourth column of Figure 2, we provide the locally determined reactive power regulation ($Q_{G,i}$ stands for reactive power at generators), which show a converging behavior in accordance with the fact that the primal gap is vanishing. A perfectly equivalent behavior is found for active powers (but this is not shown in figure).

Performance evaluation

A more in-depth performance measure for the tests of Figure 2 is given in Table 1, where the distributed

Table 1 Performance starting from a remote point

	Network					
	IEEE 30	IEEE 57	IEEE 118	IEEE 300	IEEE feeder 123	radial 120
IPOPT						
Generated power P_G	190.80 MW	1.26 GW	4.25 GW	23.74 GW	3.53 MW	169.81 kW
Number of iterations	12	10	20	33	17	22
Processing time	0.17 s	0.29 s	1.56 s	20.97 s	1.55 s	1.45 s
Algorithm 2						
Number of regions R	5	7	54	69	24	25
ρ	3	3	5	5	10	10
ζ	1/3	1/3	1/5	1/5	1/10	1/10
ξ	3	3	10	10	30	30
θ	0.99	0.99	0.99	0.99	0.99	0.99
τ	1.02	1.02	1.02	1.02	1.02	1.02
Generated power P_G	191.07 MW	1.26 GW	4.26 GW	23.79 GW	3.55 MW	170.78 kW
Gap	0.14%	0.002%	0.25%	0.23%	0.42%	0.57%
Number of iterations	110	144	186	216	246	85
Processing time (aggregate)	8.93 s	20.03 s	119.18 s	214.88 s	110.22 s	32.28 s
Max processing time per region	2.08 s	6.42 s	3.25 s	12.16 s	10.36 s	5.18 s
Average processing time per region	1.79 s	2.86 s	2.21 s	3.11 s	4.59 s	1.29 s

approach of Algorithm 2 is compared with the performance of a centralized IPOPT solver.

Note that the performance gap with respect to a central solver is always below a 1% error, which is an impressive performance considering that we are dealing with a worst case situation, and that we are approaching the problem in distributed form with a severe network partitioning. As a matter of fact, the outstanding performance of IPMs is mainly due to their central coordination capabilities (e.g., see [15]). Incidentally, we observed that the performance of Algorithm 2 is almost independent of the chosen settings. As a consequence, the performance gap in Table 1 coincides with the ultimate accuracy that could be achieved after thousands of iterations for every studied case.

By inspecting the references, the reader can further appreciate the substantial improvement with respect to the performance of the ADMM-based algorithm of [15], and the sensibly improved network size and partitioning performance with respect to the preliminary algorithm version of [16].

Processing times

Some information on the processing times involved with Algorithm 2 is given in both Table 1 and Figure 3.

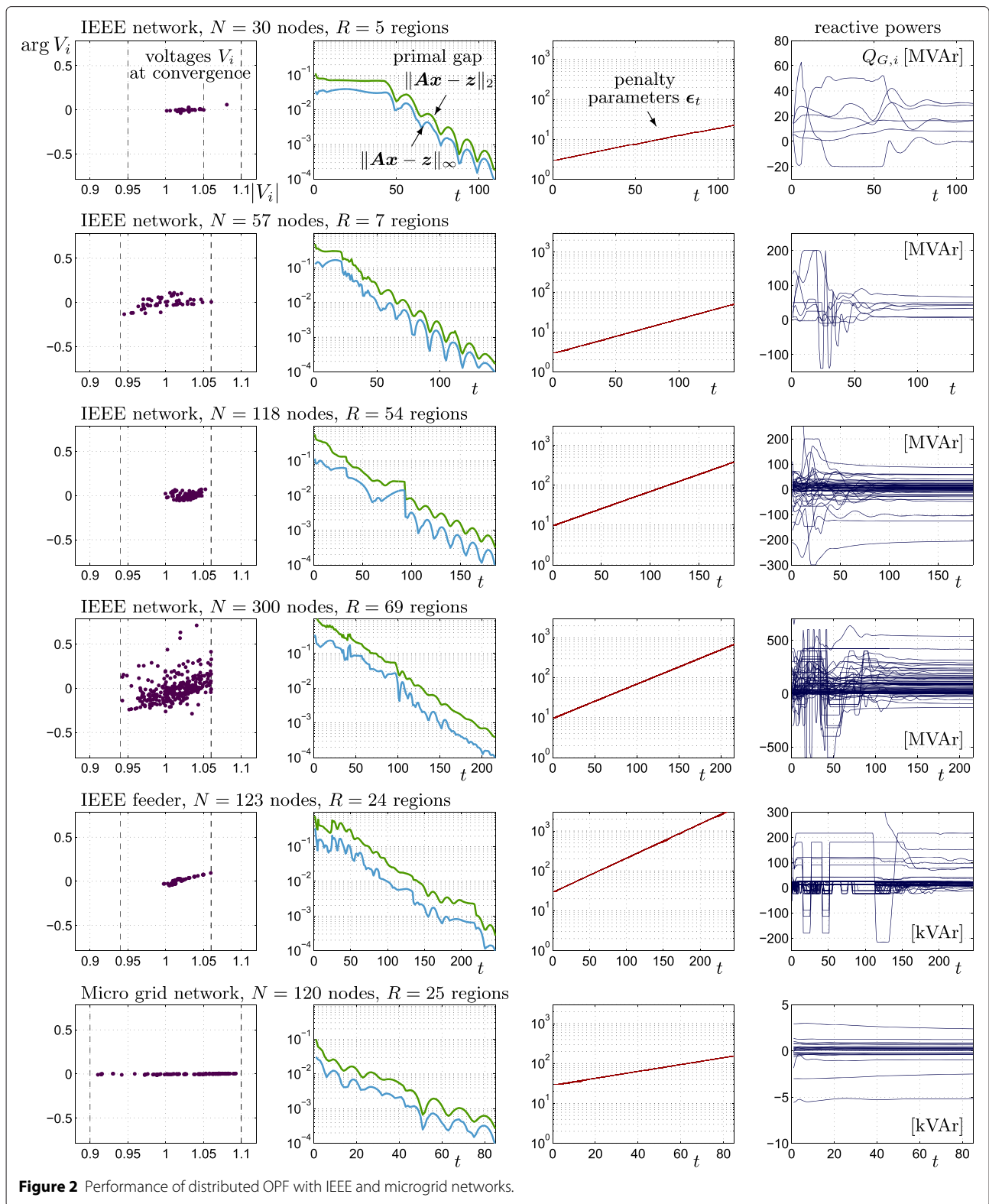
Figure 3 shows, for the six networks under consideration, the maximum local processing time and the aggregate processing time per iteration. These are almost constant throughout the iterative process, evidencing the

fact that the processing time is approximately linear in the number of iterations. From Table 1, we can further extract some information on the time needed per region (the *max processing time per region*), which is in a range between 2 and 13 s, the value being in agreement with the literature on distributed OPF (e.g., see [6]).

Observe that communication delays were not taken into account in Figure 3 and Table 1, and in fact these can be made negligible by choosing a suitable communication technique. High data rate communication standards with associated short packet lengths are to be preferred. This is the case, for example, of broadband power line communication techniques which can guarantee packet lengths of less than a millisecond [29] and which can be deployed in small area applications (e.g., in micro grids). WiMax is a wireless alternative in these scenarios. For wide area applications, instead, optical fiber communications (e.g., gigabit Ethernet) are an appropriate solution.

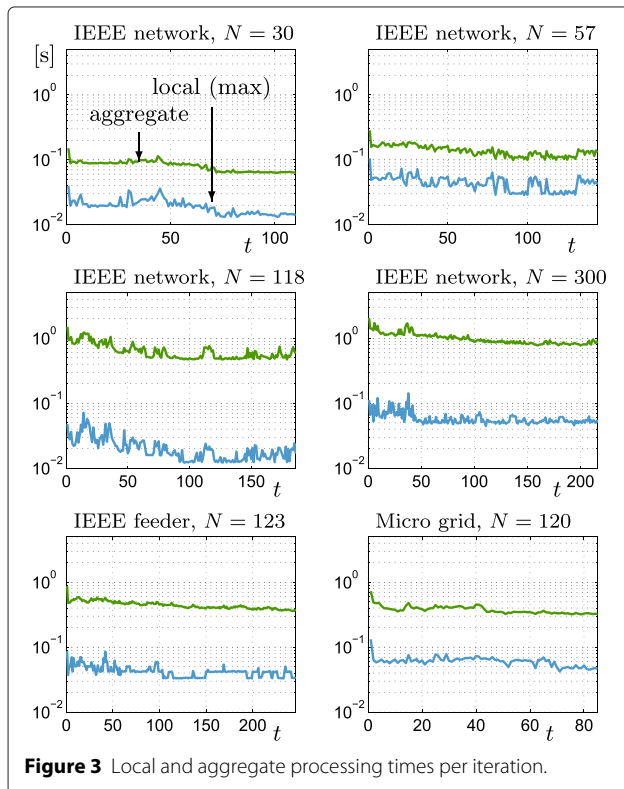
Conclusions

In this paper, we proposed a distributed algorithm for OPF regulation based upon a non-convex formulation. By suitably controlling penalty parameters, the algorithm was proven to always converge under a proper assumption on local solver reliability. A certificate of convergence to a local minimum is also available under the request that penalty factors are bounded. The algorithm was shown to provide a reliable performance also in a worst case situation where the search for the optimum is initialized



on a point very far from its final destination. The algorithm was proven to be efficient and fast and to be also robust with respect to a severe network partitioning. Its

required computational effort was found to be of the order of state-of-the-art methods (using convex problem approximations to ease the convergence issue), with the



added value of allowing for a full adherence to the original problem since no (convex) approximation is used.

On the applicability side, the distributed algorithm is readily applicable on moderate time scales (tens of seconds) and on moderate sized networks (up to 300 nodes) for system optimization purposes, not concerning fast regulation (e.g., fault or protection issues require much faster time scales). In this scenario, the algorithm is also expected to be robust to packet losses, because of its alternating direction structure.

Applicability to larger network sizes, with many loops, and more severe non-convexities is instead out of the scope of the present work. As a matter of fact, the proposed alternating direction search allows distributing the processing burden, but might not find an agreement (or it might take too long) in harsh situations. To overcome these difficulties, two strategies can be jointly employed. On the one side, some criteria to determine the optimal region partition strategy should be identified. On the other side, some additional coordination between agents should be used, e.g., a proper distributed generalization of the techniques used in the work of Martinez and Birgin et al. [18] which could also be capable of closing the performance gap with respect to a centralized solver. Use of recent advances on ADMM accelerated methods and scaling techniques (e.g., see [30]) is also an interesting option but need to be suitably adapted to a non-convex context. These aspects are left for future investigations.

Competing interests

The author declares that he has no competing interests.

Authors' information

Tomaso Erseghe was born in 1972. He received the Laurea (M.Sc degree) and the Ph.D. in Telecommunication Engineering from the University of Padova, Italy in 1996 and 2002, respectively. Since 2003, he is Assistant Professor (Ricamatore) at the Department of Information Engineering, University of Padova. His current research interest is in the fields of distributed algorithms for telecommunications, and smart grids optimization. His research activity also covered the design of ultra-wideband transmission systems, properties and applications of the fractional Fourier transform, and spectral analysis of complex modulation formats.

Received: 12 December 2014 Accepted: 14 April 2015

Published online: 23 May 2015

References

1. J Lavaei, SH Low, Zero duality gap in optimal power flow problem. *IEEE Trans. Power Syst.* **27**(1), 92–107 (2012)
2. S Sojoudi, J Lavaei, in *IEEE Conference on Decision and Control (CDC)*. On the exactness of semidefinite relaxation for nonlinear optimization over graphs: Part I (Florence, Italy, 2013), pp. 1043–1050
3. S Sojoudi, J Lavaei, in *IEEE Conference on Decision and Control (CDC)*. On the exactness of semidefinite relaxation for nonlinear optimization over graphs: part II (Florence, Italy, 2013), pp. 1051–1057
4. R Madani, S Sojoudi, J Lavaei, Convex relaxation for optimal power flow problem: mesh networks. *IEEE Trans. Power Syst.* **30**(1), 199–211 (2015)
5. AYS Lam, B Zhang, DN Tse, in *IEEE 51st Annual Conference on Decision and Control (CDC 2012)*. Distributed algorithms for optimal power flow problem (Maui, HI, 2012), pp. 430–437
6. B Zhang, AYS Lam, A Dominguez-Garcia, DN Tse, An optimal and distributed method for voltage regulation in power distribution systems. To appear in *IEEE Trans. Power Syst.*
7. E Dall'Anese, H Zhu, GB Giannakis, Distributed optimal power flow for smart microgrids. *IEEE Trans. Smart Grid.* **4**(3), 1464–1475 (2013)
8. E Dall'Anese, SV Dhople, BB Johnson, GB Giannakis, Decentralized optimal dispatch of photovoltaic inverters in residential distribution systems. *IEEE Trans. Energy Conv.* **29**(4), 957–967 (2014)
9. D Gayme, U Topcu, Optimal power flow with large-scale storage integration. *IEEE Trans. Power Syst.* **28**(2), 709–717 (2013)
10. T Erseghe, S Tomasin, Power flow optimization for smart microgrids by SDP relaxation on linear networks. *IEEE Trans. Smart Grid.* **4**(2), 751–762 (2013)
11. P Šulc, S Backhaus, M Chertkov, Optimal distributed control of reactive power via the alternating direction method of multipliers. *IEEE Trans. Energy Conversion.* **29**(4), 968–977 (2014)
12. S Magnusson, PC Weeraddana, C Fischione, A distributed approach for the optimal power flow problem based on ADMM and sequential convex approximations. To appear in *IEEE Trans. on Control of Network Systems*
13. S Kar, G Hug, in *Power and Energy Society General Meeting, 2012 IEEE*. Distributed robust economic dispatch in power systems: a consensus + innovations approach (San Diego, CA, 2012), pp. 1–8
14. J Mohammadi, S Kar, G Hug, Distributed approach for DC optimal power flow calculations. *arXiv (2014)*. arxiv.org/abs/1410.4236
15. T Erseghe, Distributed optimal power flow using ADMM. *IEEE Trans. Power Syst.* **29**(5), 2370–2380 (2014)
16. T Erseghe, in *IEEE International Conference on Smart Grid Communications, 2014*. A distributed algorithm for fast optimal power flow regulation in smart grids (Venice, Italy, 2014)
17. R Andreani, EG Birgin, JM Martínez, ML Schuverdt, On augmented lagrangian methods with general lower-level constraints. *SIAM J. Optimization.* **18**(4), 1286–1309 (2007)
18. E Birgin, J Martínez, *Practical Augmented Lagrangian Methods for Constrained Optimization*. (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014)
19. OL Mangasarian, *Nonlinear Programming*, vol. 10. (Society for Industrial and Applied Mathematics, Philadelphia, 1994)

20. RT Rockafellar, R Wets, in *Fundamental Principles of Mathematical Sciences. Variational analysis*, vol. 317 (Springer Berlin, 1998)
21. A Castillo, RP O'Neill, Computational performance of solution techniques applied to the ACOPF. Federal Energy Regulatory Commission, Optimal Power Flow Paper. **5** (2013)
22. RD Christie, Power Systems Test Case Archive. www.ee.washington.edu/research/pstca
23. WH Kersting, in *IEEE Power Engineering Society Winter Meeting, 2001. Radial distribution test feeders*, vol. 2 (Columbus, OH, 2001), pp. 908–912
24. Group, D.T.F.W.: Distribution test feeders. ewh.ieee.org/soc/pes/dsacom/testfeeders 2010
25. GA Pagani, M Aiello, Power grid network evolutions for local energy trading. arXiv (2012). arxiv.org/abs/1201.0962
26. RD Zimmerman, CE Murillo-Sánchez, RJ Thomas, Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *Power Systems, IEEE Trans.* **26**(1), 12–19 (2011)
27. A Wächter, LT Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* **106**, 25–57 (2006)
28. MATLAB, *Version 7.13.0.564 (R2011b)*. (The MathWorks Inc., Natick, Massachusetts, 2011)
29. AR Di Fazio, T Erseghe, E Ghiani, M Murrone, P Siano, F Silvestro, Integration of renewable energy sources, energy storage systems, and electrical vehicles with smart power distribution networks. *J. Ambient Intell. Humanized Comput.* **4**(6), 663–671 (2013)
30. T Goldstein, B O'Donoghue, S Setzer, R Baraniuk, Fast alternating direction optimization methods. *SIAM J. Imaging Sci.* **7**(3), 1588–1623 (2014)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Immediate publication on acceptance
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ springeropen.com
