Coordination and Control of Distributed Energy Resources for Provision of Ancillary Services

Alejandro D. Dominguez-Garcia[†] and Christoforos N. Hadjicostis[‡†]
[†]Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA
[‡]Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus
E-mail: {aledan, chadjic}@ILLINOIS.EDU

***Abstract*—This paper discusses the utilization of distributed energy resources on the distribution side of the power grid to provide a number of ancillary services. While the individual capability of these resources to provide grid support might be very small, their presence in large numbers in many distribution networks implies that, under proper control, they can collectively become an asset for providing ancillary services. An example is the power electronics interface of a photovoltaic array mounted in a residential building roof. While its primary function is to control active power flow, when properly controlled, it can also be used to provide reactive power. This paper develops and analyzes distributed control strategies to enable the utilization of these distributed resources for provision of grid support services. We provide a careful analysis of the applicability capabilities and limitations of each of these strategies. Several simulation examples are provided to illustrate the proposed approaches.**

I. Introduction

On the distribution side of a power system, it has been acknowledged that there exist many distributed energy resources that can be potentially used to provide ancillary services to the grid they are connected to [1], [2], [3]. An example is the utilization of power electronics grid interfaces commonly used in distributed generation resources to provide reactive power support. While the primary function of these power electronics-based systems is to control active power flow, when properly controlled, they can also be used to provide reactive power control to the grid they are connected to. Another example is the utilization of plug-in-hybrid vehicles (PHEV) for providing active power for up and down regulation. For instance, such resources could be utilized for energy peak-shaving during peak hours and load-leveling at night [4].

Proper coordination and control of these distributed resources is key for enabling their utilization for ancillary services. One solution to this problem can be achieved through a centralized control strategy where each distributed resource is commanded from a central controller located, for example, at the substation that interconnects the distribution network and the transmission/subtransmission network. This central controller issues a command to each distributed resource so that collectively they account for the necessary amount of, for example, active or reactive power demanded by the central controller. To achieve this goal in this centralized fashion however, it is necessary to overlay a communication network connecting the central controller with each distributed resource, and requires knowledge of the distributed resources that are available on the distribution side at any given time. This centralized approach has been proposed in [2] to provide reactive power support in distribution feeders by coordinating distributed reactive power resources, assuming the existence of two-way communication between every pair of nodes that posses reactive power resources. Once a node detects that its voltage exceeds some limits, it requests all other nodes to increase (or decrease) the amount of reactive power they are providing until the voltage returns to normal values. While any node with communication capabilities can initiate a request for reactive power, the coordination of all resources is centralized in the sense that every other node communicates directly with the one that initiated the request. The problem of distributed resources coordination for reactive power support is also addressed in [3], which proposes a centralized strategy for the utilization of the power electronics interfaces in photovoltaic systems. The commercial product described in [5] also adopts a centralized control strategy to utilize solar systems mounted on utility poles for providing reactive power support.

In this paper we propose an alternative approach that utilizes distributed strategies for control and coordination of distributed energy resources in power grids. These strategies offer several advantages, including the following: i) they are more economical because they do not require communication between a centralized controller and the various devices, ii) they do not require complete knowledge of the distributed resources available, and iii) they can be more resilient to faults and/or unpredictable behavioral patterns by the distributed resources. The proposed approaches rely on a distributed control strategy where each distributed resource can exchange information with a number of other "close-by" resources, and subsequently make a local control decision based on this available information. Collectively, the local control decisions made by the resources should have the same effect as the centralized control strategy. Such a solution could rely on inexpensive and simple communication protocols, e.g., ZigBee technology [6], that would provide the required local exchange of information for the distributed control approach to work. We pursue this distributed approach, providing algorithms that enable utilization of distributed resources for grid support.

Our approach has been inspired by consensus and coordination problems, which have a rich history in both computer science (see, e.g., [7]), and control theory (see, e.g., [8], [9], [10], [11]). In networked systems that consist of several entities (also referred to as agents, or nodes), consensus is defined as having the nodes reach an agreement regarding some quantity of interest that depends on some initial state (value) of each node in the network. A typical example of an application of consensus is a network of sensors measuring the same variable, e.g., the temperature in a room; initially, the sensors readings might be different (due to measurement noise or other peculiarities at each node), and it is desirable to exchange information with neighboring sensors to reach an agreement on the room temperature, for example, by calculating the average temperature of all measurements. An example of coordination is the movement of a flock of birds, where each bird coordinates its movement based on the movements of close-by birds such that the flock moves in a direction determined by one of the birds called the leader [12].

In our setup, the distributed resources can be thought of as nodes in a network, where each node can exchange information with neighboring nodes such that, through an iterative process, each distributed resource in the network will compute the amount of active or reactive power that it needs to provide, such that the resources collectively provide the predetermined (requested) amount of active or reactive power. We provide algorithms that solve this coordination/cooperation problem when i) there is no limit on the amount of active or reactive power that each resource can provide (though some notion of fair distribution of the contribution of active or reactive power among resources might be imposed); and ii) the maximum amount of active or reactive power each resource can provide is limited, which is a more realistic case. For the unconstrained case, the algorithms provided follow from well-known results in consensus problems [8], [10]. *On the other hand, the distributed algorithms provided for the unconstrained case are new and the main theoretical contribution of this paper.*

We believe that the proposed distributed control strategies for provision of ancillary services address two features identified as key to achieving the *Smart Grid* vision. First, they enable the active participation of consumers via demand response. In this regard, consumers have the choice to enable resources, such as solar installations in buildings and PHEVs, to provide reactive (and active) power support, for which they can be paid for by the corresponding utility. Second, it allows asset optimization and efficient operation. One example is the utilization of distributed resources for reactive power control. In this regard, even if banks of switched capacitors or other existing means that provide reactive power control cannot be completely replaced by distributed reactive power resources, it is possible to reduce their size. Furthermore, by generating reactive power closer to the points where it is consumed, losses in transmission and distribution systems can be reduced.

The remainder of this paper is organized as follows. Section II provides some background on graph theory and provides the general form of the iterative distributed algorithms to be discussed. Section III describes strategies for cases where there are no constraints on node capacity. Section IV discusses strategies for cases where constraints are imposed on node maximum capacity. Section V presents concluding remarks.

II. Preliminaries

The exchange of information between nodes where resources are located can be described by a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{1, 2, \dots, n\}$ is the vertex set (each vertex corresponds to a node), and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of directed edges, where $(j, i) \in \mathcal{E}$ if node j can receive information from node i. The graph is undirected if and only if whenever $(j, i) \in \mathcal{E}$, then also $(i, j) \in \mathcal{E}$, i.e., if node j can receive information from node i, then node i can also receive information from node j. All nodes that can transmit information to node j are said to be neighbors of node j and are represented by the set $\mathcal{N}_j = \{i \in \mathcal{V} : (j, i) \in \mathcal{E}\}$. The number of neighbors of j is called the in-degree of j and denoted by $\mathcal{D}_j^-$. The number of nodes that have j as neighbor, i.e., j can transmit information to these nodes, is called the out-degree of j and is denoted by $\mathcal{D}_j^+$.

Let $\pi_j[k]$ be the amount of active or reactive power demanded from the distributed resource located in node j at the k round of information exchange between nodes. Then, the distributed algorithms we propose to determine the amount of resource that will be contributed by node j perform linear iterations of the form

$$\pi_j[k+1] = p_{jj}[k]\pi_j[k] + \sum_{i \in \mathcal{N}_j} p_{ji}[k]\pi_i[k], \tag{1}$$

where the $p_{ji}[k]$'s are a set of (potentially time-varying) weights[1]. Each node updates its demanded amount to be a linear combination of its own demanded amount and the demanded amount of its neighbors. As we will see, the choice of $p_{ji}[k]$'s will depend on the problem constraints. We discuss first the case where there are no constraints on the node capacity. While this is not realistic, the approach to solve the problem provides the foundations for subsequently addressing the constrained case.

III. Distributed Control Strategies without Constraints on Node Capacity

We assume there is a leading node that knows the total amount of active or reactive power ρ_d that needs to be collectively provided by the remaining n nodes. This leader can communicate with $l \geq 1$ nodes, and initially sends a command demanding ρ_d/l units of active or reactive power from each of them. Unless ρ_d changes, the leader will not subsequently communicate with the nodes.

Since the nodes do not have constraints on the amount of active or reactive power they can provide, a simple solution is that each of the l nodes that the leader initially communicated with provides exactly ρ_d/l of active or reactive power and the remaining $n - l$ nodes do not provide any active or reactive power. However, it is obvious that in the constrained case this strategy will not work if the l nodes the leader initially communicates to cannot provide the amount of active or reactive power demanded by the leader. Before this problem

[1] In this paper we discuss strategies with time-invariant weights p_{ij}, but we have also studied strategies with time-varying weights.

is addressed (in the next section), we provide an iterative algorithm that allows all the nodes to participate in providing an amount of active or reactive power, so that collectively they account for the total amount needed.

Let $\pi_j[k]$ be the active or reactive power demanded from node j at step k, and define the corresponding active or reactive power demand vector as $\pi[k] = [\pi_1[k],\ \pi_2[k], \dots, \pi_j[k], \dots, \pi_n[k]]'$. Define the collective active or reactive power demand as $\rho[k] = \sum_{j=1}^{n} \pi_j[k]$, and let ρ_d be the collective active or reactive power demanded from the leader. The objective is to design a distributed iterative algorithm that, at step k, updates the active or reactive power demand from node j based on i) its current active or reactive power demand $\pi_j[k]$, and ii) the current active or reactive power demanded from neighbors of j (nodes that can transmit information to j), such that after m steps the collective active or reactive power demand equals the total active or reactive demanded by the leader: $\rho[m] = \sum_{j=1}^{n} \pi_j[m] = \rho_d$.

A. Splitting Strategy

The simplest solution, which results in constant weights p_{ji}, is for each node j to equally split its current value among itself and the nodes that have j as neighbor, i.e., the nodes that j can transmit information to [10]. Thus, for each node j

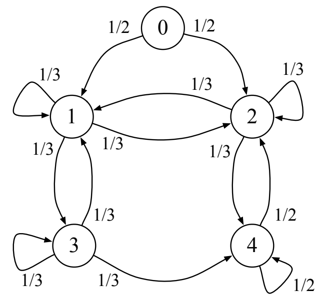
$$\pi_j[k+1] = \frac{1}{1+\mathcal{D}_j^+}\pi_j[k] + \sum_{i \in \mathcal{N}_j} \frac{1}{1+\mathcal{D}_i^+}\pi_i[k] \qquad (2)$$

where $\mathcal{D}_i^+$ the number of nodes that i can transmit information to (the out-degree of node i). Algorithm (2) does not necessarily split the total active or reactive power demand ρ_d evenly among all the nodes, but it ensures that $\sum_{j=1}^{N} \pi_j[k] = \rho_d,\ \forall k \geq 0$. Furthermore, provided the directed graph describing the exchanges between nodes has a single recurrent class, which necessarily makes it aperiodic by construction due to the fact that $\frac{1}{1+\mathcal{D}_j^+} \neq 0$, the steady state solution provided by (2) is unique. To see this, we can rewrite (2) in matrix form as

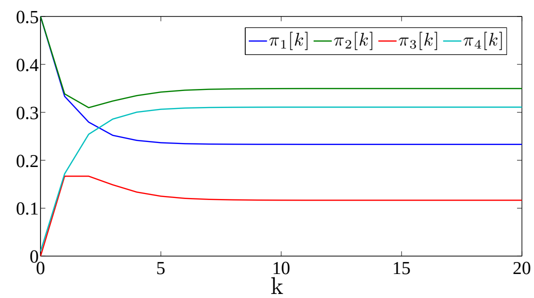
$$\begin{aligned} \pi[k+1] &= P_c \pi[k], \\ \pi[0] &= \pi_0, \end{aligned} \qquad (3)$$

where $\pi_0 = [\pi_1[0],\ \pi_2[0], \dots, \pi_j[0], \dots, \pi_n[0]]'$ with $\pi_i[0] = \rho_d / l$ if i is a neighbor of the leader node and $\pi_i[0] = 0$ otherwise. By construction, matrix P_c is *column stochastic*[2], i.e., the sum of the entries of each column adds up to one, and also primitive [13]. Since P_c is a column stochastic and primitive matrix, the Perron-Frobenius theorem for nonnegative matrices states that P_c has a unique eigenvalue with largest modulus at $\lambda_1 = 1$ (see e.g., [14], [13]). Let x be a right eigenvector of P_c associated with λ_1 and let y be a left eigenvector of P_c associated with λ_1 such that $x'y = 1$. Again, from the fact that P_c is column stochastic, the entries

[2]The structure of P_c is such that column j has entries $p_{ij} = \frac{1}{1+\mathcal{D}_j^+}$ for $i \in N_j \cup \{i\}$, and 0 otherwise. Thus, P_c is easily verified to be column stochastic.



(a) Network topology.



(b) Node demanded capacity evolution.

Fig. 1. Four-node network implementing even splitting strategy.

of vector y must be all equal. Without loss of generality, let $y = [1, 1, \dots, 1]'$, and since $x'y = 1$, the entries of x must add up to one. Then the steady-state solution of (3) (and therefore the steady-state solution of (2)) is given by

$$\pi^{ss} = xy'\pi_0 = \Big(\sum_{j=1}^{N} \pi_j[0]\Big)x. \qquad (4)$$

Since $\sum_{j=1}^{N} \pi_j[0] = \rho_d$ and the entries of x are nonnegative and add up to one, it follows that entries of π^{ss} are nonnegative and add up to ρ_d; therefore, the nodes collectively provide the total active or reactive power demand.

Example 1: Consider the network of nodes of Fig. 1(a). Let $\rho_d = 1$ and assume that the node initial values are zero. The leader is indexed by 0 and, as explained before, initially splits ρ_d in half and passes it to nodes 1 and 2. Then, following (2), each node updates its value as follows

$$\begin{aligned} \pi_1[k+1] &= \tfrac{1}{3}\big(\pi_1[k] + \pi_2[k] + \pi_3[k]\big), \\ \pi_2[k+1] &= \tfrac{1}{3}\big(\pi_1[k] + \pi_2[k]\big) + \tfrac{1}{2}\pi_4[k], \\ \pi_3[k+1] &= \tfrac{1}{3}\big(\pi_1[k] + \pi_3[k]\big), \\ \pi_4[k+1] &= \tfrac{1}{3}\big(\pi_2[k] + \pi_3[k]\big) + \tfrac{1}{2}\pi_4[k], \end{aligned} \qquad (5)$$

with $\pi_1[0] = \pi_2[0] = 1/2$, and $\pi_3[0] = \pi_4[0] = 0$. Letting $\pi[0] = [\pi_1[0],\ \pi_2[0],\ \pi_3[0],\ \pi_4[0]]'$, we have

$$\begin{aligned} \pi[k+1] &= P_c \pi[k], \\ \pi[0] &= [1/2,\ 1/2,\ 0,\ 0]', \end{aligned} \qquad (6)$$

where

$$P_c = \begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 \\ 1/3 & 1/3 & 0 & 1/2 \\ 1/3 & 0 & 1/3 & 0 \\ 0 & 1/3 & 1/3 & 1/2 \end{bmatrix}.$$

Fig. 1(b) shows the evolution of the node active or reactive power demands until they reach the steady-state solution to (6) given by $\pi^{ss} = [0.23,\ 0.35,\ 0.11,\ 0.31]'$. Note that nodes do not contribute equally because the matrix P_c is only column-stochastic but not row-stochastic. □

B. Even Splitting Strategy

A solution to reach even splitting can be easily obtained when $\sum_{i=1}^{n} p_{ji} = \sum_{i=1}^{n} p_{ij} = 1$ for all $j = 1, 2, \dots, n$, i.e., the sum of the weights that each node j uses to update its current value is equal to the sum of the weights used by node j to split its own value among itself and the nodes that have j as neighbor [10]. The simplest realization of such algorithm is obtained when the graph describing the exchanges of information is undirected, i.e., if node j can receive information from node i, then node i can also receive information from node j, which results in equal in- and out-degrees for each node, i.e., $\mathcal{D}_j^- = \mathcal{D}_j^+ := \mathcal{D}_j,\ \forall j = 1, \dots, n$. If we define the maximum degree of the network as $\mathcal{D} = \max_j\{\mathcal{D}_j\}$, one way even splitting can be achieved is by having each node j update its value as follows:

$$\pi_j[k+1] = \big(1 - \frac{|\mathcal{N}_j|}{1+\mathcal{D}}\big)\pi_j[k] + \sum_{i \in \mathcal{N}_j} \frac{1}{1+\mathcal{D}}\pi_i[k], \quad (7)$$

where $|\mathcal{N}_j| = \mathcal{D}_j^+$ denotes the number of elements in the set $\mathcal{N}_j$, i.e, the number of nodes that node j communicates with; and $\pi_j[0] = \rho_d/l$ if the leader is a neighbor of node j, and zero otherwise (in fact, even splitting can also be achieved if instead of $\mathcal{D}$, we use any upper bound $\mathcal{D}' \geq \mathcal{D}$ in (7)).

Note that the approach in (7), apart from requiring bi-directional information exchange between nodes, it also requires each node to know an upper bound $\mathcal{D}'$ on the maximum degree $\mathcal{D}$ of the network, which in practice can be easily implemented in the design by imposing that each node can only communicate with a limited number of nodes. Provided the directed graph describing the exchanges between nodes has a single recurrent class (which necessarily makes it aperiodic by construction due to the fact that the diagonal element $1 - \frac{|\mathcal{N}_j|}{1+\mathcal{D}} \neq 0,\ \forall j = 1, 2, \dots, n$), the steady state solution provided by (7) is unique and equal for all nodes j. To see this, rewrite (7) in matrix form as

$$\begin{aligned} \pi[k+1] &= P\pi[k], \\ \pi[0] &= \pi_0, \end{aligned} \quad (8)$$

where $\pi_0 = [\pi_1[0],\ \pi_2[0], \dots, \pi_j[0], \dots, \pi_n[0]]'$ with $\pi_i[0] = \rho_d/l$ if i is a neighbor of the leader node and $\pi_i[0] = 0$. By construction, the matrix P is *doubly stochastic*[3], i.e., the sum of entries of each row and each column adds up to one [13]. From the fact that P is doubly stochastic, and as a result of the Perron-Frobenius theorem, the steady-state solution of (8) is given by $\pi^{ss} = xy'\pi_0$, with $x'y = 1$, and where x is a right eigenvector associated with the eigenvalue $\lambda_1 = 1$ (the unique eigenvalue of P with largest modulus), and y is a left eigenvector associated with λ_1. Since P is doubly stochastic, all entries of y must be equal, all entries of x must be equal, and we can choose without loss of generality $x = [1, 1, \dots, 1]'$ and $y = \frac{1}{N}[1, 1, \dots, 1]'$. Then the steady-state solution of (8) (and therefore the steady-state solution of (7)) is given by

$$\pi^{ss} = \big(\frac{1}{N}\sum_{j=1}^{N}\pi_j[0]\big)[1, 1, \dots, 1]'. \quad (9)$$

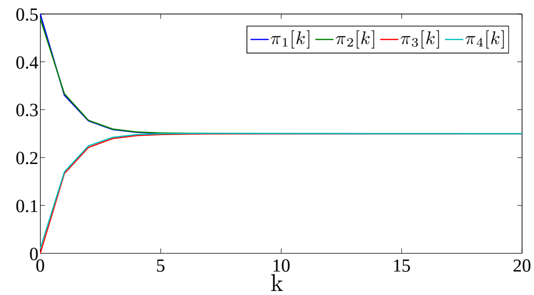


Fig. 2. Node demanded capacity evolution for four-node network with an even splitting strategy.

Example 2: Consider again the network of Fig. 1(a). By following (7), it is easy to see that, the entries of P in (8) are the same as the entries of P_c in Example 1 except for a few ones that are modified as follows: $p_{24} = 1/3$, $p_{44} = 2/3$, $p_{33} = 2/3$, and $p_{43} = 0$ (note that there is no longer communication between 3 and 4), and the rows also add up to one, so that matrix P becomes doubly stochastic. Fig. 2 shows the evolution of node demands, where it can be seen that the steady-state solution is $\pi^{ss} = [0.25,\ 0.25,\ 0.25,\ 0.25]'$. □

IV. Distributed Control Strategies with Constraints on Node Capacity

Armed with the analysis in the previous section, we now address the case where nodes have limits on the amounts of active or reactive power they can provide. Let π_j^{max}, for $j = 1, 2, \dots, n$, be the maximum active or reactive power that node j can provide (its maximum capacity), and define the corresponding maximum active or reactive power capacity vector as $\pi^{max} = [\pi_1^{max},\ \pi_2^{max}, \dots, \pi_n^{max}]'$. As before, we let $\rho[k] = \sum_{j=1}^{n}\pi_j[k]$ be the collective active or reactive power capacity demanded from the nodes at instant k, and ρ_d be the collective active or reactive power demand. We assume that $\rho_d \leq \sum_{j=1}^{n}\pi_j^{max} := \chi^{max}$.

[3]The structure of P is such that it is symmetric with $p_{ij} = p_{ji} = \frac{1}{1+\mathcal{D}}$ if $j \in N_i$ (i.e., $i \in N_j$), and 0 otherwise; each diagonal entry p_{ii} is chosen so that the sum of the entries of each row/column is one.

The objective is to design a distributed iterative algorithm that, at step k, updates the active or reactive power demanded from node j based on i) its current active or reactive power demand $\pi_j[k]$, and ii) the current active or reactive power demanded by neighboring nodes that communicate to j, such that after m steps:

1) $\pi_j[m]$ reaches a steady state value and $\pi_j[m] \leq \pi_j^{max}, \forall j$; and
2) the collective active or reactive power provided by all the nodes equals the total active or reactive demanded by the leader: $\rho[m] = \sum_{j=1}^{n} \pi_j[m] = \rho_d \leq \chi^{max}$.

A. Fair Splitting with Constraints on Network Topology

A simple solution to the constrained problem can be obtained if each node could compute (or knows) the maximum active or reactive power capacity χ^{max} that the nodes in the network can collectively provide, and the total active or reactive power demand ρ_d. Once each node has computed π^{max} and ρ_d, the total active or reactive power demand can be collectively provided by having each node j provide $\pi_j[m] = \frac{\rho_d}{\chi^{max}}\pi_j^{max} \leq \pi_j^{max}$. The problem with this approach is that by using distributed algorithms of the form given in (1), each individual node can neither compute χ^{max} nor ρ_d, because each node does not necessarily know the exact number of nodes in the network. At best, by assuming that i) each pair of nodes in the network can exchange information bidirectionally and, ii) there is a limit $\mathcal{D}$ on the number of nodes that a given node can exchange information with (or an upper bound $\mathcal{D}'$ on this maximum number of neighbors), each node could compute the quantities $\frac{\chi^{max}}{N}$ and $\frac{\rho_d}{N}$, where N is the total (but unknown) number of nodes in the network. This can be achieved if the nodes use the algorithm in (7) twice; first to exchange the initially demanded capacity from them and, second to exchange the maximum capacities they can provide. To summarize, if we set the initial conditions of (7) to be ρ_d/l if j is one of the l nodes the leader can communicate to and zero otherwise, the steady-state solution of (7) will be $\frac{\rho_d}{N}, \forall j = 1, \dots, n$; whereas, if we set the initial conditions of (7) to be π_j^{max}, the steady solution of (7) is $\frac{\sum_j \pi_j^{max}}{N} = \frac{\chi^{max}}{N}, \forall j = 1, \dots, n$. Then, by dividing these two quantities, we obtain that $\frac{\rho_d/N}{\chi^{max}/N} = \frac{\rho_d}{\chi^{max}}$, and the total demanded capacity ρ_d can be satisfied by having each node j set its contribution to

$$\pi_j = \frac{\rho_d/N}{\chi^{max}/N}\pi_j^{max} = \frac{\rho_d}{\chi^{max}}\pi_j^{max}. \tag{10}$$

Note that the nodes do not need to know N (the total number of nodes in the network) and can calculate $\frac{\chi^{max}}{N}$ and $\frac{\rho_d}{N}$ simultaneously by separately performing two local averaging operations at each iteration. It is important to keep in mind that the algorithm in (7) requires symmetric communication between nodes and an upper bound $\mathcal{D}'$ on the number of nodes a given node can communicate to (i.e., it requires an upper bound on the maximum degree $\mathcal{D}$ of the network).

Example 3: Consider again the network and even splitting algorithm of Example 2. Take $\rho_d = 1$ and $\pi^{max} = [0.4,\ 0.2,\ 0.4,\ 0.1]$ so that $\chi^{max} = 1.1 > \rho_d$. Each node runs twice (in parallel) the algorithm in (7). For the first one, the initial conditions for nodes 1 and 2 are 0.5 and 0 for nodes 3 and 4. For the second one, the initial conditions of each node are set to the corresponding entries of $\pi^{max} = [0.4,\ 0.2,\ 0.4,\ 0.1]$. The first step of the algorithm is equivalent to the unconstrained case of Example 1, displayed in Fig. 2, where it can be seen that the node converge to 0.25 (the total amount requested divided by the number of nodes), whereas the evolution of the second run is displayed in Fig. 3, where it can be see that each nodes converges to the average of the entries of π^{max}, i.e., $\sum_{j=1}^{4} \pi_j^{max}/4 = 0.275$. □

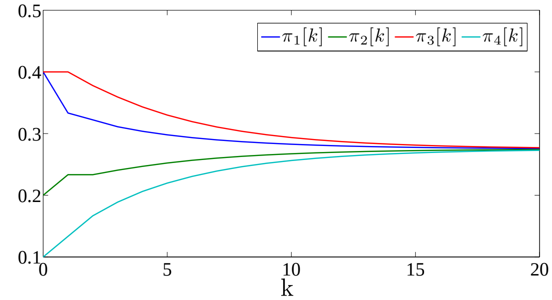


Fig. 3. Evolution of the algorithm run by each node to compute $\sum_{j=1}^{4} \pi_j^{max}/4$ (average of node maximum capacities).

B. Fair Splitting without Constraints on Network Topology

A solution can also be achieved without assuming bidirectional communication between nodes and without a limit on the number of nodes a given node can communicate with. In this case, the algorithm consists of three steps:

1) Each node j uses the simple splitting solution of (2), where $\pi^{ss} = [\pi_1^{ss}, \pi_2^{ss}, \dots, \pi_n^{ss}]'$ is the unique nonnegative vector with entries that sum to 1 that satisfy $\pi^{ss} = P_c \pi^{ss}$, where P_c was defined in (3).
2) Each node j computes $\mu = \min_j \{\frac{\pi_j^{max}}{\pi_j^{ss}}\}$, which can be done in finite time (bounded by the diameter of the network[4]) via the following algorithm:

$$\mu_j[k+1] = \min\{\mu_j[k], \min_{i \in \mathcal{N}_j}\{\mu_i[k]\}\},$$
$$\mu_j[0] = \frac{\pi_j^{max}}{\pi_j^{ss}}. \tag{11}$$

3) Each node j computes $\delta_j = \frac{\min_j\{\pi_j^{max}/\pi_j^{ss}\}}{\pi_j^{max}/\pi_j^{ss}} = \frac{\mu}{\mu_j[0]}$ (note that $0 < \delta_j \leq 1$) and adjusts the weights in (2) as

$$\pi_j[k+1] = \big(1 - \delta_j(1 - \frac{1}{1+\mathcal{D}_j^+})\big)\pi_j[k] + \sum_{i \in \mathcal{N}_j} \delta_i \frac{1}{1+\mathcal{D}_i^+}\pi_i[k]. \tag{12}$$

[4]The diameter of the network is defined to be the maximum shortest path between any pair of nodes in the network [15].

To see the steady-state solution reached by (12), we can rewrite (12) in matrix form as

$$\pi[k+1] = \big[P_c\Delta + (I - \Delta)\big]\pi[k] := \hat{P}\pi[k], \qquad (13)$$

with $\pi[0] = \pi_0 = [\pi_1[0],\ \pi_2[0], \ldots, \pi_j[0], \ldots, \pi_n[0]]'$, I being the identity matrix, and $\Delta = \mathrm{diag}\big(\delta_1, \delta_2, \ldots, \delta_n\big)$ being a diagonal matrix with $0 < \delta_j \leq 1,\ \forall j = 1, 2, \ldots, n$. Since i) P_c is a column stochastic matrix with a unique eigenvector corresponding to the eigenvalue of largest modulus $\lambda_1 = 1$; and ii) the non-zero diagonal entries of the matrix $\hat{P}$ are the same as the non-zero diagonal entries of the matrix P_c; it follows that the matrix $\hat{P} = P_c\Delta + (I - \Delta)$ is also a column stochastic matrix with a unique eigenvector corresponding to the eigenvalue of largest modulus $\lambda_1 = 1$ (see, e.g., [14]). Let π^{ss} be the solution to $\pi^{ss} = P_c\pi^{ss}$ as discussed in step 1) above, and let $\hat{\pi}^{ss}$ be the steady-state solution to (13). Then, we have $\hat{P}\hat{\pi}^{ss} = (P_c\Delta + (I - \Delta))\hat{\pi}^{ss} = \hat{\pi}^{ss}$, from where it follows that $P_c\Delta\hat{\pi}^{ss} = \Delta\hat{\pi}^{ss}$.

The Perron-Frobenius theorem ensures the uniques of the solution of (3) up to a positive constant. Thus $\Delta\hat{\pi}^{ss} = \alpha\pi^{ss}$ for some $\alpha > 0$ such that $\sum_{j=1}^{n} \hat{\pi}_j^{ss} = \rho_d$ (since $\hat{P}$ is column-stochastic, we know that $\sum_{j=1}^{n} \pi_j^{ss} = \rho_d$); therefore, $\delta_j\hat{\pi}_j^{ss} = \alpha\pi_j^{ss}$, where $0 < \delta_j \leq 1,\ \forall j = 1, \ldots, n$, from where it follows then that $\alpha\pi_j^{ss} \leq \hat{\pi}_j^{ss},\ \forall j = 1, \ldots, n$. If $\pi_j^{ss} < \pi_j^{max},\ \forall j = 1, \ldots, n$ (as desired), we have $\alpha \leq \frac{\pi_j^{max}}{\pi_j^{ss}}\ \forall j = 1, \ldots, n$ or $\alpha \leq \min_j \frac{\pi_j^{max}}{\pi_j^{ss}}$. By choosing $\alpha = \frac{\rho_d}{\sum_{j=1}^{n}\pi_j^{max}} \min_j\{\frac{\pi_j^{max}}{\pi_j^{ss}}\}$ and $\delta_j = \frac{\min_j\{\pi_j^{max}/\pi_j^{ss}\}}{\pi_j^{max}/\pi_j^{ss}}$, it follows that $\hat{\pi_j}^{ss} = \frac{\rho_d}{\sum_{j=1}^{n}\pi_j^{max}}\pi_j^{max} \leq \pi_j^{max}$ (and also $\sum_{j=1}^{n} \hat{\pi}_j^{ss} = \rho_d$), achieving fair splitting as in (10) but without imposing any constraints on the network topology.

Example 4: Consider again the network and splitting algorithm of Example 1. Take $\rho_d = 1$ and $\pi^{max} = [0.4,\ 0.2,\ 0.4,\ 0.1]$. The first step of the algorithm is equivalent to the unconstrained case of Example 1, displayed in Fig. 1(b). The second step (11) of the algorithm converges in three iterations as the diameter of the network is determined by the length of the path between nodes 3 and 4, which happens to be 3; thus $\mu_1[3] = \mu_2[3] = \mu_3[3] = \mu_4[3] = 0.3226$. Each node adjusts its weights according to the third step of the algorithm, which results in $\delta_1 = 0.185$, $\delta_2 = 0.565$, $\delta_3 = 0.088$, and $\delta_4 = 1$. The evolution of (12) is displayed in Fig. 4, where it can be seen that the steady-state solution is $\pi^{ss} = [0.359,\ 0.177,\ 0.376,\ 0.088]'$, with all the entries smaller than the corresponding entries of $\pi^{max} = [0.4,\ 0.2,\ 0.4,\ 0.1]$. $\square$

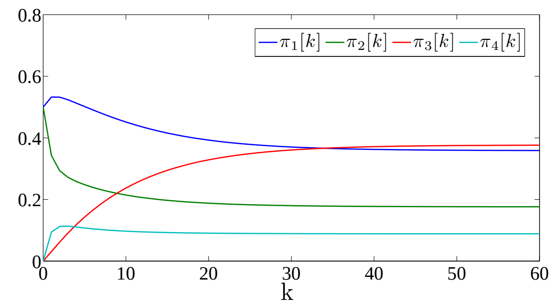


Fig. 4. Fair splitting algorithm evolution after nodes modified their weights.

V. Concluding Remarks and Future Work

We have studied distributed control strategies that can be used to determine (in a distributed fashion) the amount of active or reactive power that needs to be provided by distributed active and reactive power resources. These strategies have the potential to enable assets already present in distribution systems as active and reactive power support resources. Further work will investigate the existence of faster algorithms for the constrained case. It is also important to investigate the algorithms performance in the presence of faults, e.g., broken communication links, and nodes not updating their value.

References

[1] G. Joos, B. Ooi, D. McGillis, F. Galiana, and R. Marceau, “The potential of distributed generation to provide ancillary services,” in *Proc. IEEE PES Summer Meeting*, vol. 3, Seattle, WA, 2000.

[2] M. Baran and I. El-Markabi, “A multiagent-based dispatching scheme for distributed generators for voltage support on distribution feeders,” *IEEE Trans. on Power Systems*, vol. 22, no. 1, pp. 52 –59, Feb. 2007.

[3] S. B. K. Turitsyn, P. Sulc and M. Chertkov, “Distributed control of reactive power flow in a radial distribution circuit with high photovoltaic penetration,” in *IEEE Power and Energy Society General Meeting*, Minneapolis, MN, Jul. 2010.

[4] C. Guille and G. Gross, “A conceptual framework for the vehicle-to-grid (v2g) implementation,” *Energy Policy*, vol. 37, no. 11, pp. 4379–4390, Nov. 2009.

[5] Petra Solar. (2009) SunWave Communications System. South Plainfield, NJ. [Online]. Available: http://www.petrasolar.com/

[6] ZigBee Alliance. (2008, Jan.) ZigBee Specification. San Ramon, CA. [Online]. Available: http://www.zigbee.org

[7] N. Lynch, *Distributed Algorithms*. San Mateo, CA: Morgan Kaufmann Publishers, 1996.

[8] J. Tsitsiklis, “Problems in decentralized decision making and computation,” Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1984.

[9] A. Jadbabaie, J. Lin, and A. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” *IEEE Trans. on Automatic Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.

[10] R. Olfati-Saber, J. Fax, and R. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[11] S. Sundaram and C. Hadjicostis, “Distributed function calculation and consensus using linear iterative strategies,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 650–660, May 2008.

[12] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, “Flocking in fixed and switching networks,” *IEEE Trans. on Automatic Control*, vol. 52, no. 5, pp. 863–868, May 2007.

[13] R. Horn and C. Johnson, *Matrix Analysis*. New York, NY: Cambridge University Press, 1985.

[14] E. Seneta, *Non-negative Matrices and Markov Chains*, revised printing ed. New York, NY: Springer, 2006.

[15] C. Godsil and G. Royle, *Algebraic Graph Theory*. New York, NY: Springer, 2001.