Distributed Power Allocation in Prosumer Networks ⋆

Thiagarajan Ramachandran, * Zak Costello, Peter Kingston, Santiago Grijalva, Magnus Egerstedt

* *Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: tramachandran3@gatech.edu, z.costello@gatech.edu, kingston@gatech.edu, sgrijalva@ece.gatech.edu, magnus@ece.gatech.edu)*

Abstract: Due to requirements of renewable and distributed energy integration,sustainability and energy security, the existing power grid is undergoing radical changes.The lines between producers and consumers are becoming blurred. In this paper, we provide a constrained formulation of the power allocation problem, which is emerging in such producer-consumer hybrid environments, and obtain decentralized protocols for its solution. We show that the information required by the individual prosumers to solve the problem varies depending on the choice of constraints in the optimization problem. As a consequence of this, we demonstrate that there exist fundamental trade-offs between the information required and the convergence rates associated with the power allocation problem.

Keywords: power grid, distributed optimization, information-exchange networks

1. INTRODUCTION

In recent years, changes in the power industry have been posing challenges to the power grid. As renewable resources drop in cost and approach price parity with fossil power, intermittent sources will become a larger part of total generation. Additionally, power generation will be more distributed, with residential customers more frequently having generation capacity. As this shift occurs, the lines between producer and consumer become less clear leading to a hybrid prosumer. In the near future, any agent on the power grid will be able to have generation capacity, storage capacity, and loads. For a detailed discussion on this trend, see Ipakchi and Albuyeh [2009] and Willis and Scott [2000].

The fundamental problem in power system operation is balancing power generation and power demand. Because electricity cannot be stored in large quantities, this balance needs to occur in real-time in a delicate maner. Deviation from this balance will result in system frequency deviaiton and a potential blackout. Each type of power system, such electric utilities, microgrids and buildings need to address the issue of powerbalancing. In this paper we approach the problem of balancing using a generic power balancing agent: the prosumer, which was introduced in Grijalva and Tariq [2011]. As the electricity infrastructure becomes more complex and equipped with sensing and information networks (e.g. the smart grid), the number of elements that need to be controlled will drastically increased, since potentially every load to invidual appliances can be controlled in an intelligent manner to achieve enhance reliability or economic optimization. This optimization and control problem cannot be solved in a centralized manner, hence a distributed approach needs to be developed, and the prosumer is the key concept of this approach. In fact, the power grid is undergoing a shift from a heterogeneous network where producers and consumers are easily delineated to a homogeneous prosumer network. We model the power grid as a collection of connected prosumer agents using a DC power flow model and investigate distributed algorithms which can be used to control power allocation in such a networks. This power flow model is chosen to focus on the structural implications of solving distributed minimization problems instead on the details of the model, even though it should be noted that all results in this paper carry over to AC flows as well.

The novelty and contributions in this paper fall along three different dimensions. The first is the utilization of the prosumer abstraction which allows us to reason about heterogeneous actors on the grid in a unified manner, in an autonomous control framework. The second is a formulation of a power allocation problem for the said prosumer network in such a way that computation of the solution can be carried out in a decentralized manner. The third is an investigation of how the different solutions to this problem have different structural implications for the information exchange network needed to accompany the physical power transmission network.

In this paper, we model the power allocation problem as a constrained weighted least squares problem assuming a DC power flow model and derive two scalable decentralized controllers which solves the problem and compare their topological implications on the required information network. The power allocation problem is essentially a distributed optimization problem. Other work which is technically similar to ours goes under the heading of multi-

⋆ This work done is supported by the US Department of Energy through its Advanced Research Projects Agency (ARPA-E).

agent optimization, including Nedic and Ozdaglar [2007], Ozdaglar [2007] and Terelius et al. [2011].Nevertheless, none of the aforementioned work examine the connections between algorithmic choices and the information topology induced by that choice which is the key contribution of this paper.

The outline of this paper is as follows: In Section 2, we present an unconstrained formulation of the power allocation problem. It is a minimization problem where the decision variable are the power flows along the transmission lines. In a real power network, flows would be controlled by changing power generation at prosumer nodes. But, it is unclear from a prosumer perspective what control action an individual node should take given this problem formulation. This motivates Section 3. In order to make clear what control actions should be taken and which agent should take them, we reformulate the unconstrained problem as a constrained optimization problem where the decision variable is now the power produced by the prosumers. A weighted least squares sum of residuals is minimized subject to a power conservation constraint, where the residual is the discrepancy between desired power $\mathcal{N}$ and actual power p. In Section 4, we recast the constrained optimization problem in Section 3 in a different fashion which allows us to extract decentralized control laws. In Section 5, we solve the constrained optimization defined in Section 4 under different equivalent constraints and obtain two different decentralized control laws: an edge-oriented law and a node-oriented control law. In order to understand the information requirements of the edge oriented control law, it is recast as a node based control law by exploiting the relationship between the edge states and the node states in Section 6. Then we compare implied information network topologies and convergence rates between the two control laws in Sections 6 and 7 respectively.

2. PROBLEM STATEMENT

We consider a set of k prosumer agents, denoted by $V = \{v_1, \ldots, v_k\}$. These agents are assumed to be connected over a transmission network that they use for communications and power exchange. Formally, we represent the network by a connected, undirected graph $\mathcal{G} = (V, E)$ on V with m edges, in which the presence of an edge $(v_i, v_j) \in E$ indicates that agents v_i and v_j can communicate and exchange power bidirectionally. For the purpose of notational ease, we associate an arbitrary orientation to $\mathcal{G}$ which will serve to give meaning to positive and negative flows along edges. Indeed, for each edge $e_i \in E$, we define a *signed power flow* $r_i \in \mathbb{R}$ across e_i, and collect these flows in a vector $r = (r_1, \ldots, r_m)^T \in \mathbb{R}^m$. In the subsequent sections, we will present distributed algorithms that manipulate flow vectors to satisfy the energy needs of the various agents.

We assume that each agent i in the network has computed its desired power need $\mathcal{N}_i \in \mathbb{R}$, by taking into account its personal load, storage and generation capabilities. The value $\mathcal{N}_i$ is negative if agent i desires to generate/produce power and positive if it wants to consume power. Agents with positive $\mathcal{N}_i$ require more power than they produce, and are requesting power from the network. Likewise, agents with negative $\mathcal{N}_i$ have excess power available, which they are offering for distribution. We collect the power needs of the individual agents in the vector $\mathcal{N} = (\mathcal{N}_1, \ldots, \mathcal{N}_k)^T$.

In a physical power network, the power produced by any node must subsequently be consumed by some other node or nodes. For example, if prosumer i is generating 100W of power, then there must be an prosumer or a group of prosumers which is consuming 100W of power. Therefore, prosumers cannot produce or consume power in isolation. The actual power which is being produced or consumed by an agent is a determined by the amount of power which is being injected and withdrawn from the transmission/distribution infrastructure represented as power flow along the edges. The power $p_i \in \mathbb{R}$ that is actually available to prosumer i is determined by the power flows along edges incident to v_i. Letting $\mathcal{D}(\mathcal{G}) \in \mathbb{R}^{k \times m}$ be the node-edge incidence matrix for $\mathcal{G}$ (given the arbitrary orientation) and given a flow $r \in \mathbb{R}^m$, we define the power vector $p \in \mathbb{R}^k$ by

$$p = \mathcal{D}r \ . \tag{1}$$

The vector r represents the power flows along the edges whose directionality is determined by the incidence matrix $\mathcal{D}$. We would like to compute a power-flow vector r such that the net weighted discrepancy between the power $p = \mathcal{D}r$ and the desired power $\mathcal{N}$ is minimized. This problem can be phrased as least-squares optimization problem,

$$\min_r \ \frac{1}{2}(\mathcal{D}r - \mathcal{N})^T W (\mathcal{D}r - \mathcal{N}), \tag{2}$$

where $W = W^T \succ 0$ is a diagonal, positive definite weight matrix.

The interpretation is that we are optimizing over power flows in order to ensure that agents' power needs are satisfied as closely as possible, in a least-squares sense. The weight matrix W captures the relative importance of each agent's need in the network. If agent j is a critical facility (e.g a hospital) , whose power needs are important, then the w_{jj} term is made larger. Also, smaller agents like electric vehicles have smaller tolerances and poor safety mechanisms and cannot handle large power fluctuations. This is also taken into account when assigning weights for the prosumers.

First of all, we immediately see that, as this is a standard linear least-squares problem, the optimal r can be found by computing an appropriate Moore-Penrose psuedo-inverse. Unfortunately, such a computation is inherently centralized and it will not endow the individual prosumers with distributed actions. Instead, one possible way to determine r is to differentiate the cost in (2) with respect to r and perform a (hopefully distributed) gradient descent to find the optimal value of r. But, since the dimension of the r is equal to the number of edges in the network and since the transmission infrastructure is a passive component and does not perform computations, this computation can unfortunately not be distributed among the prosumers in the network.

In this paper, we present an alternative formulation of the above problem as a constrained optimization problem where the decision variable is actually p, the available power at each individual node, instead of the flows r used in the definition of (2). We will show that we can

recover not only the flows r, but also obtain an alternative characterization of the flows as potential differences across adjacent nodes. We also point out the relationship between the structure of the solution and the constraints it imposes on the information topology of the grid.

Before proceeding, we give a helpful alternate characterization of feasible power flows, in Lemma 1, which is really just a direct consequence of the fundamental theorem of linear algebra:

Lemma 1. A vector $p \in \mathbb{R}^k$ can be expressed as $p = \mathcal{D}r$ for some $r \in \mathbb{R}^m$, if and only if $\mathbf{1}^T p = 0$, where $\mathbf{1} \triangleq (1, 1, \ldots, 1)^T \in \mathbb{R}^k$.

Proof. We must show that range$(\mathcal{D}) \perp$ span $\mathbf{1}$. Since range$(\mathcal{D})^\perp$ = null$(\mathcal{D}^T)$ and it is also known that null$(\mathcal{D}^T)$ = span$(\mathbf{1})$ for weakly connected, directed networks, we have range$(\mathcal{D})^\perp$ = span$(\mathbf{1})$. ∎

In short, the requirement $\mathbf{1}^T p = 0$ expresses the conservation law that the power generated in the network equal the power consumed.

3. ANTICIPATING AVAILABLE POWER

Using Lemma 1, it is possible to solve for p directly without computing the flow r, as the solution to the minimization problem

$$\begin{aligned} &\min_p \frac{1}{2}(p-\mathcal{N})^T W (p-\mathcal{N}) \\ &\text{s.t. } \mathbf{1}^T p = 0 \end{aligned} \quad . \tag{3}$$

By Lemma 1, the constraint $\mathbf{1}^T p = 0$ is equivalent to asserting that p belongs to range$(\mathcal{D})$. The Lagrangian for the above problem is given by

$$L(p, \upsilon) = \frac{1}{2}(p-\mathcal{N})^T W (p-\mathcal{N}) + \upsilon \mathbf{1}^T p \tag{4}$$

where $\upsilon \in \mathbb{R}$ being the Lagrange multiplier. Setting $\frac{\partial L}{\partial p}|_{p=p^*, \upsilon=\upsilon^*} = 0$ and solving for the minimizer p^*, we have $p^* = \mathcal{N} - \upsilon W^{-1}\mathbf{1}$ and $\upsilon^* = \frac{\mathbf{1}^T\mathcal{N}}{\mathbf{1}^T W^{-1}\mathbf{1}}$. When $W = I$ is the identity matrix, we have $\upsilon^* = \frac{1}{k}\mathbf{1}^T\mathcal{N}$, which is the average of the network's power needs. The term $\upsilon^* W^{-1}\mathbf{1}$ can be interpreted as the optimal disparity vector as it represents the difference between the power need $\mathcal{N}$ and the optimal power vector p^*.

Let $\mathcal{F} \triangleq \{p \mid p = \mathcal{N} - \alpha W^{-1}\mathbf{1} \text{ for } \alpha \in \mathbb{R}\}$. We can understand the role of the Lagrange multiplier υ by projecting the vectors in $\mathcal{F}$ onto the span$(\mathbf{1})$. The projection of a vector p onto span$(\mathbf{1})$, derived using normal equations, is given by $\frac{\mathbf{1}^T p}{\mathbf{1}^T\mathbf{1}}$.The projection $\Pi : \mathbb{R} \to \mathbb{R}$ of a vector $p \in \mathcal{F}$ parameterized by α onto span$(\mathbf{1})$ is given by the following equation:

$$\Pi(\alpha) = \frac{\mathbf{1}^T}{\mathbf{1}^T\mathbf{1}}\mathcal{N} - \frac{\mathbf{1}^T}{\mathbf{1}^T\mathbf{1}}(\alpha W^{-1}\mathbf{1}) \tag{5}$$

Note that $\Pi(\alpha) = 0$ when $\alpha = \upsilon^* = \frac{\mathbf{1}^T\mathcal{N}}{\mathbf{1}^T W^{-1}\mathbf{1}}$. The Lagrange multiplier υ^* determines the magnitude of the error $\alpha W^{-1}\mathbf{1}$ such that the projection of the error term onto span$(\mathbf{1})$ cancels out the projection of the power need $\mathcal{N}$ onto span$(\mathbf{1})$ thus rendering p^* orthogonal to span$(\mathbf{1})$.

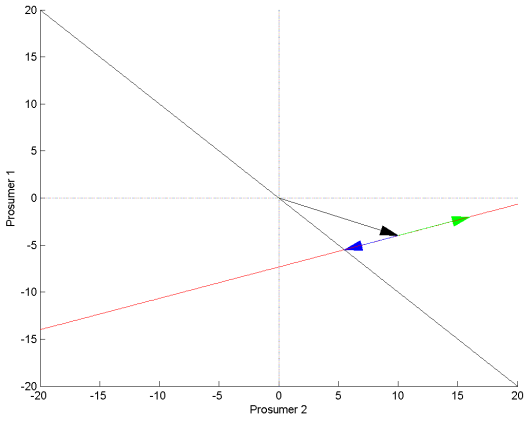


Fig. 1. The red line represents the set $\mathcal{F}$. The black line is the orthogonal complement of span$(\mathbf{1})$. The vector in black represents the power need $\mathcal{N}$. The vector in blue is the optimal disparity vector expressed as $\upsilon^* W^{-1}\mathbf{1}$ which terminates on the orthogonal complement of span$(\mathbf{1})$ as required. The vector in green is a example of a suboptimal disparity vector given by $\alpha W^{-1}\mathbf{1}$ where α was chosen at random.

This is illustarted in Figure 1 for a two prosumer system where $\mathcal{N} = [10 \ -4]^T$ and weight matrix $W \in \mathbb{R}^{2\times 2}$ where $w_{ij} = 0$ when $i \neq j$, $w_{11} = 1$ and $w_{22} = 4$.

We can use the fact that $p^* \in \mathcal{F}$ and $\Pi(\upsilon^*) = 0$ to reformulate (3) as a constrained optimization problem.

4. CONSTRAINED FORMULATION

In this section, we reformulate (2) as an equivalent, constrained minimization problem which allows us to generate decentralized control laws depending on the characterization of the constraint $p \in \mathcal{F}$.

Consider the following optimization problem:

$$\begin{aligned} &\min_p \frac{1}{2} p^T W p \\ &\text{s.t. } p \in \mathcal{F} \end{aligned} \tag{6}$$

where $\mathcal{D}$ is the incidence matrix. We can show that the optimization problem defined in (6) is equivalent to (3) as follows.

Lemma 2. Let p^* be the minimizer for (3) and $\tilde{p}^*$ be the minimizer for (6). Then $p^* = \tilde{p}^*$.

Proof.

Let $p^* = \mathcal{N} - \upsilon^* W^{-1}\mathbf{1}$ where $\upsilon^* = \frac{\mathbf{1}^T\mathcal{N}}{\mathbf{1}^T W^{-1}\mathbf{1}}$ denote the minimizer for (3) and let $h(\alpha) = \mathcal{N} - \alpha W^{-1}\mathbf{1}$. The range of h is the constraint space $\mathcal{F}$ and the tangent to $\mathcal{F}$ at a point $p \in \mathcal{F}$ is given by $\frac{\partial h}{\partial \alpha} = W^{-1}\mathbf{1}$

Since $\tilde{p}^*$ is a minimizer for (6), it lies in $\mathcal{F}$.

Let $J = \frac{1}{2} p^T W p$. From the Karush-Kuhn-Tucker conditions, we know that, at $\tilde{p}^*$, the gradient $\nabla J(\tilde{p}^*)$ can be expressed as a linear combination of the surface normals to the constraint surface $\mathcal{F}$. Therefore, at $\tilde{p}^*$, the gradient $\nabla J(\tilde{p}^*)$ will be orthogonal to the tangent vector to $\mathcal{F}$ at

$\tilde{p}^*$. The tangent vector to $\mathcal{F}$, $\frac{\partial h}{\partial \alpha}$ at any point is simply the vector $W^{-1}\mathbf{1}$.Then we have:

$$\nabla J(\tilde{p}^*)^T(W^{-1}\mathbf{1}) = \tilde{p}^{*T}\mathbf{1} = 0 \tag{7}$$

Since $\tilde{p}^*$ lies in $\mathcal{F}$, $\tilde{p}^* = \mathcal{N} - \alpha^* W^{-1}\mathbf{1}$ for some $\alpha^* \in \mathbb{R}$. Substituting $\tilde{p}^* = \mathcal{N} - \alpha^* W^{-1}\mathbf{1}$ in (7),we obtain

$$\tilde{p}^{*T}\mathbf{1} = \mathbf{1}^T\mathcal{N} - \mathbf{1}^T(\alpha^* W^{-1}\mathbf{1}) = 0 \tag{8}$$

and $\alpha^* = \upsilon^* = \frac{\mathbf{1}^T\mathcal{N}}{\mathbf{1}^T W^{-1}\mathbf{1}}$. Therefore, we have $p^* = \tilde{p}^*$. ■

Also, note that $\nabla J(\tilde{p}^*)^T(W^{-1}\mathbf{1})$ referred to in (7) is simply the directional derivative $D_e J(p)$ along the direction $e = W^{-1}\mathbf{1}$. The cost function J was chosen in a such way that the directional derivative $D_e J(p)$ is proportional to the projection of p onto span($\mathbf{1}$) along the direction $e = W^{-1}\mathbf{1}$. Therefore, $D_e J(\tilde{p}^*) = 0$ implies that the projection of $\tilde{p}^*$ onto span($\mathbf{1}$) is zero. This means that $\tilde{p}^*$ is orthogonal to $\mathbf{1}$ and consequently satisfies the conservation constraint.

For a general discussion on the application of projection theorems to obtain solutions to linear least-squares problem, see Luenberger [1997].

5. FLOWS, POTENTIALS AND LAGRANGE MULTIPLIERS

We now point out the connection between the unconstrained optimization problem where we optimize over flows r defined by (2) and the constrained optimization problem where we solve for the feasible power p defined by (6). We can recast (6) in a much more concrete fashion by noting that error space span($W^{-1}\mathbf{1}$) is the null space of the matrix $\mathcal{D}^T W$. Then we can rewrite (6) as

$$\begin{aligned} &\min_p \frac{1}{2} p^T W p \\ &\text{s.t. } \mathcal{D}^T W(\mathcal{N} - p) = 0 \end{aligned} \tag{9}$$

whose Lagrangian is given by

$$L(p, \eta) = \frac{1}{2} p^T W p + \eta^T \mathcal{D}^T W(\mathcal{N} - p) \tag{10}$$

where $\eta \in \mathbb{R}^m$ is the Lagrange multiplier. We can obtain p as a function of η by setting $\frac{\partial L}{\partial p}^T = Wp - W\mathcal{D}\eta = 0$. We do so, and obtain $p = \mathcal{D}\eta$. We can then determine the Lagrange dual function by substituting $p = \mathcal{D}\eta$ in (10) and obtain

$$g(\eta) = \inf_p L(p, \eta) = -\frac{1}{2}\eta^T \mathcal{D}^T W \mathcal{D}\eta + \eta^T \mathcal{D}^T W \mathcal{N}. \tag{11}$$

The minimizer for (10) is $p^* = D\eta^*$ where $\eta^* = \underset{\eta}{\text{argmax}}\, g(\eta)$. We can use this fact to define an update law for η by performing gradient ascent on (11) by letting

$$\dot{\eta} = \frac{\partial g}{\partial \eta} = -\mathcal{D}^T W \mathcal{D}\eta + \mathcal{D}^T W \mathcal{N}. \tag{12}$$

The Lagrange multiplier η^* is identical to the flow r^* with respect to the orientation defined by the incidence matrix $\mathcal{D}$, where r^* is the minimizer for the problem defined by (2). The $\mathcal{D}^T W \mathcal{D}$ matrix is called the weighted edge Laplacian. For a more detailed discussion of the weighted edge Laplacian, see Zelazo et al. [2007].

So, the protocol defined by (12) in conjunction with $p = D\eta$, solves the problem defined by (2).

The Lagrange multiplier η determines the feasible power vector p the same way the flow vector r defined in (2) does. Since η is a Lagrange multiplier, its dimension is determined by the dimension of the constraints which is equal to m, where m is the number of edges in the network. As a result, the control law given by (12) becomes a edge oriented control law. We can use this observation to obtain a nodal control law by recasting the above m-dimensional constraint into a n-dimensional one,where n is the number of nodes.

An alternative way to characterize the constraint $p \in \mathcal{F}$ is to use the graph Laplacian $\mathcal{L}$ instead of the transpose of the node-edge incidence matrix $\mathcal{D}$. The graph Laplacian shares the same nullspace with $\mathcal{D}^T$; i.e null($\mathcal{D}^T$) = null($\mathcal{L}$) = $\mathbf{1}$. So, we can use this fact to reformulate (6) as follows:

$$\begin{aligned} &\min_p \frac{1}{2} p^T W p \\ &\text{s.t. } \mathcal{L}W(\mathcal{N} - p) = 0 \end{aligned} \tag{13}$$

whose Lagrangian is then

$$L(p, q) = \frac{1}{2} p^T W p + q^T \mathcal{L} W(\mathcal{N} - p) \tag{14}$$

where $q \in \mathbb{R}^n$ is the Lagrange multiplier. Let g be the Lagrange dual function associated with the Lagrangian given by (14). We can then proceed to solve this problem in a similar fashion to (9) and obtain $p = \mathcal{L}q$ and an update law for q given by

$$\dot{q} = \frac{\partial g}{\partial q} = -\mathcal{L}W\mathcal{L}q + \mathcal{L}W\mathcal{N}. \tag{15}$$

The Lagrange multiplier q^* here provides an alternate way to define the flow r^* as the potential difference between the nodes, where the role of the potentials is played by the Lagrange multiplier q^*. The flows are then defined as $r^* = \mathcal{D}^T q^*$ and since the feasible power at each node is defined as $p^* = \mathcal{D}r^*$, we have $p^* = \mathcal{D}\mathcal{D}^T q^* = \mathcal{L}q^*$.

6. INFORMATION REQUIREMENTS

In this section, we will identify the information topology which must be supported by the prosumer network to execute the gradient ascent protocols defined by (12) and (15).

Since the Lagrange multiplier q lies in $\mathbb{R}^n$, where n is the number of prosumers in the network, we can imagine prosumer i to be responsbile for updating the value q_i. We can then infer the information required by prosumer i to perform the q-update by inspecting the protocol given by (15). Then, the $\mathcal{L}W\mathcal{L}q$ term implies that agent i needs to keep track of the potentials q_j such that $j \in cl(N_i)$ where $cl(N_i)$ denotes the closure of the neighbourhood set of i. Therefore protocol (15) requires each prosumer to keep track of the q-values of its neighbours and the q-values of neighbours-of-its-neighbours (i.e neighbours that are 2-hops away or less).

To identify the information required by an individual prosumer i to execute the edge-oriented update law given by (12), we need to recast it into a nodal update law.

Define node-potentials to be a vector $q \in \mathbb{R}^n$. Then the flow η induced by the potential q is $\mathcal{D}^T q$. To obtain a update law for q, we note that $\dot{\eta} = \mathcal{D}^T \dot{q}$. Since (12) already defines an update law for the flows η, we have

$$\dot{\eta} = \mathcal{D}^T \dot{q} = -\mathcal{D}^T (W\mathcal{D}\eta + W\mathcal{N}) = -\mathcal{D}^T (W\mathcal{D}\mathcal{D}^T q + W\mathcal{N}) \tag{16}$$

This then gives us a update law for q as follows:

$$\dot{q} = W(-\mathcal{L}q + \mathcal{N}) \tag{17}$$

Clearly, we have $\dot{q} = 0$ when $\mathcal{L}q = \mathcal{N}$. The above update law defined for q is not always stable as $\mathcal{N}$ need not lie in the **range** $\mathcal{L}$. We can achive stabilization by simply projecting $\mathcal{N}$ into the **range** $\mathcal{L}$ as follows:

$$\begin{aligned} &\dot{q} = W(-\mathcal{L}q + \mathcal{N} - s) \\ &\text{where } \dot{s} = -\mathcal{L}Ws \text{ and } s(0) = \mathcal{N} \end{aligned} \tag{18}$$

The update on $s \in \mathbb{R}^N$ is defined such that s converges to $\alpha W^{-1}\mathbf{1}$ where $\alpha = \frac{\mathbf{1}^T\mathcal{N}}{\mathbf{1}^T W^{-1}\mathbf{1}}$. For more details on the convergence properties of ODEs involving graph Laplacian, see Mesbahi and Egerstedt [2010]. So, when $\dot{q} = 0$, we obtain $p = \mathcal{L}q = \mathcal{N} - \alpha W^{-1}\mathbf{1}$ as expected.

We can now determine the information required by individual prosumers by inspecting (18). Both the q-update law and the s-update law requires only that we know the state pair (q_i,s_i) of the neighbouring prosumers (encoded by the $\mathcal{L}$ term). So, the update law given by (15) requires a one-hop information network as opposed to (15), but it requires that each prosumer maintain two states, s and q, instead of one.

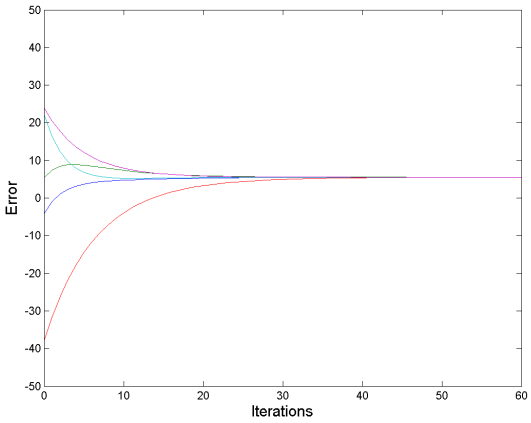
Also, the update law (15) typically converges to the optimal solution faster than (18). This is because the convergence rate of (15) is proportional by λ_2^2 and the convergence of (18) is proportional by λ_2 where λ_2 is the second smallest eigenvalue of the Laplacian matrix.

7. EXAMPLE - A 100 PROSUMER SYSTEM

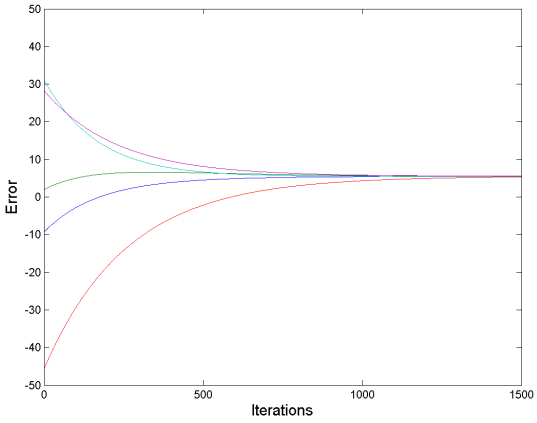
In this section, we compare the performance of the controllers given by (15) and (18) on a randomly generated prosumer network with two different weight functions.

A prosumer graph $\mathcal{G}$ with 100 prosumers was generated randomly using a Erdos-Renyi random graph model and the corresponding graph Laplacian $\mathcal{L}$ was constructed. For a discussion on random graphs, see Erdős and Rényi [1960]. The smallest non-zero eigen value of the generated graph, which determines the rate of convergence, was 34.6389. The power need vector $\mathcal{N}$ was also generated randomly with values ranging in between -70 to 70 watts. The average surplus in the network, given by $\frac{1}{100}\mathbf{1}^T\mathcal{N}$, is 5.5173 . The potentials q_1 and q_2 were randomly initialized and are modified according to the update law given by (15) and (18), respectively.

The weight function W_1 is the identity matrix $I_{100} \in \mathbb{R}^{100\times100}$. The power p converges to $\mathcal{N} - \alpha\mathbf{1}$ where $\alpha = \frac{1}{100}\mathbf{1}^T\mathcal{N} = 5.5173$. Figure 2(a) tracks the evolution of the error $\mathcal{N} - Lq_1$ for the first five nodes where q_1 is updated according to (18) while Figure 2(b) tracks $\mathcal{N} - Lq_2$ for the same set of nodes, where q_2 is updated according to (15). Note that the error does converge to 5.5173, average of the power needs $\mathcal{N}$. Also, the rate of the convergence of the update law (15) is much higher than that of the update law given by (18). This is due to the fact that the convergence rate of (15) is proportional to the square of the smallest non-zero eigen value of the graph Laplacian $\mathcal{L}$, due to the $\mathcal{L}W\mathcal{L}$ term. The rate of convergence of the update law given by (18) is determined by the smallest non-zero eigen value of $\mathcal{L}$ which is approximately 1120. As a result Lq_1 converges to the minimum in less that 50 iterations while Lq_2 takes almost 1500 iterations to converge.



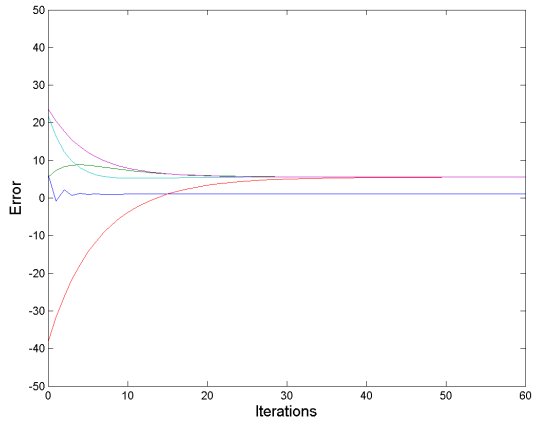
(a) Convergence of error at nodes 1 through 5 using update law (15)



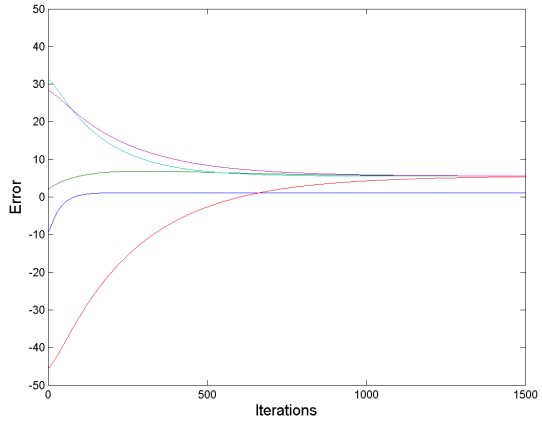
(b) Convergence of error at nodes 1 through 5 using update law (18)

Fig. 2. (a) tracks the error $\mathcal{N} - \mathcal{L}q_1$ while (b) tracks the error $\mathcal{N} - \mathcal{L}q_2$. q_1 is updated according to (15) while q_2 is updated according to (18). The weights are chosen to be the identity matrix.

The weight function W_2 is the matrix where $W_{2ii} = 1$ if $i \neq 1$ and $W_{211} = 5$. All the other entries, W_{2ij} where $i \neq j$, is set to 0. In short, prosumer 1 is weighted in such a fashion that it is considered 5 times more important the rest of the prosumers and will suffer 5 times less error than the others. Figure 3(a) tracks $\mathcal{N} - Lq_1$ where q_1 is updated according to (15) while Figure 3(b) tracks $\mathcal{N} - Lq_2$ where



(a) Error convergence at nodes 1 through 5 using update law (15)



(b) Error convergence at nodes 1 through 5 using update law (18)

Fig. 3. (a) tracks the error $\mathcal{N} - \mathcal{L}q_1$ while (b) tracks the error $\mathcal{N} - \mathcal{L}q_2$. q_1 is updated according to (15) while q_2 is updated according to (18). Agent 1(blue) suffers lower error than the others as it was weighted much higher than the other agents

q_2 is updated according to (15). The error suffered by prosumer 1(blue) is just 1.1125, while the other prosumers bear $5 \times 1.1125 = 5.5623$. The performance of both the update law is similar to the unweighted case for the same reasons.

8. CONCLUSIONS

We have utilized a prosumer based abstraction which allows us to reason about agents in a producer-consumer hybrid environments. We have solved the power allocation problem in a distributed mannner, which is a protoypical problem in such environments. Finally, we have demonstrated how different solutions have structural implications for the information exchage network needed to accompany the physical power transmission network.

REFERENCES

Erdős, P. and Rényi, A. (1960). *On the evolution of random graphs*. Akad. Kiadó.

Grijalva, S. and Tariq, M. (2011). Prosumer-based smart grid architecture enables a flat, sustainable electricity industry. In *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, 1–6. IEEE.

Ipakchi, A. and Albuyeh, F. (2009). Grid of the future. *Power and Energy Magazine, IEEE*, 7(2), 52–62.

Luenberger, D. (1997). *Optimization by vector space methods*. Wiley-Interscience.

Mesbahi, M. and Egerstedt, M. (2010). *Graph theoretic methods in multiagent networks*. Princeton Univ Pr.

Nedic, A. and Ozdaglar, A. (2007). On the rate of convergence of distributed subgradient methods for multi-agent optimization. In *Decision and Control, 2007 46th IEEE Conference on*, 4711–4716. IEEE.

Ozdaglar, A. (2007). Distributed subgradient methods for multi-agent optimization.

Terelius, H., Topcu, U., and Murray, R. (2011). Decentralized multi-agent optimization via dual decomposition. In *18th World Congress of the International Federation of Automatic Control (IFAC), Milano, Italy*.

Willis, H. and Scott, W. (2000). *Distributed power generation: planning and evaluation*, volume 10. CRC Press.

Zelazo, D., Rahmani, A., and Mesbahi, M. (2007). Agreement via the edge laplacian. In *Decision and Control, 2007 46th IEEE Conference on*, 2309–2314. IEEE.