2010 American Control Conference
Marriott Waterfront, Baltimore, MD, USA
June 30-July 02, 2010

A Gossip Algorithm for Convex Consensus Optimization over Networks

Jie Lu, Choon Yik Tang, Paul R. Regier, and Travis D. Bow

Abstract— **In many applications, nodes in a network wish to achieve not only a consensus, but an optimal one. To date, a family of subgradient algorithms have been proposed to solve this problem under general convexity assumptions. This paper shows that, with a few additional mild assumptions, a fundamentally different, non-gradient-based algorithm with appealing features can be constructed. Specifically, we develop *Pairwise Equalizing* (PE), a gossip-style, distributed asynchronous iterative algorithm for achieving unconstrained, separable, convex consensus optimization over undirected networks with time-varying topologies, where each component function is strictly convex, continuously differentiable, and has a minimizer. We show that PE is easy to implement, bypasses limitations facing the subgradient algorithms, and produces a switched, nonlinear, networked dynamical system that is deterministically and stochastically asymptotically convergent. Moreover, we show that PE admits a common Lyapunov function and reduces to the well-studied Pairwise Averaging and Randomized Gossip Algorithm in a special case.**

I. Introduction

Consider an N-node multi-hop network, where each node i observes a convex function f_i, and all the N nodes wish to determine an optimal consensus x^*, which minimizes the sum of the f_i's:

$$x^* \in \arg\min_x \sum_{i=1}^{N} f_i(x). \tag{1}$$

Since each node i knows only its own f_i, the nodes cannot individually compute the optimal consensus x^* and, thus, must collaborate to do so. This problem of achieving unconstrained, separable, convex consensus optimization has many applications in multi-agent systems and wired/wireless/social networks. For instance, the problems of least-squares and robust estimation [1], energy-based source localization and clustering/density estimation [2] can all be cast into the form of (1). As another example, in a social network, $f_i(x)$ may represent an individual i's level of dissatisfaction if the network takes decision x, so that finding an optimal decision x^* means minimizing the total dissatisfaction across the network, where everyone's voice is heard.

The current literature offers a large body of work on distributed consensus (e.g., [3]–[16]), including a line of research that focuses on solving problem (1) for an optimal consensus x^* [1], [2], [17]–[28]. This line of work has resulted in a family of discrete-time subgradient algorithms, including the *incremental* subgradient algorithms [1], [2], [17]–[21], [23], [28], whereby an estimate of x^* is passed around the network, and the *non-incremental* ones [22], [24]–[27], whereby each node maintains an estimate of x^* and updates it iteratively by exchanging information with neighbors. Regardless of the categories, these algorithms rely on the notion of a stepsize to execute. Furthermore, the algorithms mostly assume delay- and error-free communications with no quantization over directed graphs, although there are a few exceptions: [18] allowed the presence of bounded time delays, [28] considered random additive errors in subgradient evaluations, [20], [26] studied the effects of quantization, and [24], [27] assumed undirected graphs.

Although the aforementioned subgradient algorithms are capable of solving problem (1) under weak assumptions, they suffer from one or more of the following limitations:

L1) *Stepsizes*: To implement the subgradient algorithms, it is necessary to select stepsizes, which may be constant, diminishing, or dynamic. In general, constant stepsizes ensure only convergence to neighborhoods of x^*, rather than to x^* itself. Moreover, they present an inevitable trade-off: larger stepsizes tend to yield larger convergence neighborhoods, while smaller ones tend to yield slower convergence, both of which are undesirable. Diminishing stepsizes, on the other hand, avoid the issue with lack of asymptotic convergence. However, they may lead to very slow convergence, since the stepsizes may diminish too quickly. Finally, dynamic stepsizes provide an interesting alternative for shaping convergence behavior [17], [19]. Unfortunately, their dynamics depend on global information that is often unavailable, or costly to obtain, limiting their applicability. Hence, selecting appropriate stepsizes is not a trivial task, and inappropriate choices can severely hamper algorithm performance.

L2) *Hamiltonian cycle*: Most of the incremental subgradient algorithms [1], [2], [17]–[20], [23], [28] require the network to contain a Hamiltonian cycle, i.e., a closed path that visits every node exactly once. Such a cycle, however, does not exist in many graphs [29]. In fact, determining its existence in a given graph, and finding it when it does exist, are both NP-complete problems [29]. Further compounding the complexity are the needs to maintain the cycle when the graph changes (but does not destroy its existence), inform each node of its predecessor and successor along the latest cycle, and do all of these possibly in a decentralized, leaderless fashion.

L3) *Multi-hop transmissions*: Some of the incremental subgradient algorithms [17]–[19] require the node that has the latest estimate of x^* to pass it on to a randomly and equiprobably chosen node in the network. This implies that every node must be aware of all the nodes in the network, and the algorithms must run alongside a routing protocol that enables passing of the estimate of x^*, which may not always be the case. The fact that the chosen node is typically multiple hops away also implies that these algorithms are communication inefficient, requiring plenty of transmissions (up to the network diameter) just to complete a single iteration.

L4) *Lack of asymptotic convergence*: A variety of convergence properties have been established for the subgradient algorithms in [1], [2], [17]–[28], including error bounds, convergence in expectations, convergence in the sense of limit inferiors, convergence rates, etc. In contrast, few asymptotic convergence results have been reported, except for the incremental subgradient algorithms with diminishing or

J. Lu, C. Y. Tang, and P. R. Regier are with the School of Electrical and Computer Engineering, University of Oklahoma, Norman, OK 73019, USA (e-mail: {jie.lu-1,cytang,paulregier}@ou.edu).

T. D. Bow is with the Department of Mechanical Engineering, Oklahoma Christian University, Edmond, OK 73013, USA (e-mail: travis.bow@student.oc.edu).

This work was supported by the National Science Foundation under grant CMMI-0900806.

P. R. Regier and T. D. Bow were supported by the National Science Foundation Research Experiences for Undergraduates program under grant EEC-07555011.

dynamic stepsizes in [17]–[19], [23], [28].

Limitations L1–L4 facing the existing subgradient algorithms raise the question of whether it is possible to devise an algorithm, which requires neither the notion of a stepsize, the existence of a Hamiltonian cycle, nor the use of a routing protocol for multi-hop transmissions, and yet guarantees asymptotic convergence, bypassing L1–L4. In this paper, we show that, by imposing several mild assumptions on the network and the problem, such an algorithm can be constructed. Specifically, instead of assuming that the network is directed, we assume that it is undirected, with possibly a time-varying topology unknown to any of the nodes. In addition, instead of assuming that each f_i in (1) is convex but not necessarily differentiable, we assume that it is strictly convex, continuously differentiable, and has a minimizer. Based on these assumptions, we develop a gossip-style, distributed asynchronous iterative algorithm, referred to as *Pairwise Equalizing* (PE), which not only solves problem (1) and circumvents limitations L1–L4, but also is rather easy to implement, making it an attractive alternative to the subgradient algorithms in many applications.

As will be shown in the paper, PE exhibits a number of notable features. First, it produces a switched, nonlinear, networked dynamical system whose state evolves along an invariant manifold whenever a pair of adjacent nodes gossip with each other. The switched system is proved, using Lyapunov stability theory, to be deterministically and stochastically asymptotically convergent, as long as the gossiping pattern is sufficiently rich. In particular, we show that the first-order convexity condition [30] can be used to form a common Lyapunov function, as well as to characterize drops in its value after every gossip. Second, PE does not belong to the family of subgradient algorithms as it utilizes a fundamentally different, non-gradient-based update rule that involves no stepsize. This update rule is synthesized from a blend of three simple ideas—namely, *conservation*, *dissipation*, and *equalizing*—which is somewhat similar to how *Pairwise Averaging* [31] was conceived back in the 1980s. Indeed, we show that PE reduces to Pairwise Averaging [31], *Randomized Gossip Algorithm* [32], and *Anti-Entropy Aggregation* [33], [34] when problem (1) specializes to an averaging problem.

The outline of this paper is as follows: Section II models the network and formulates the problem. Section III describes the proposed algorithm PE, while Section IV analyzes its convergence properties. Section V illustrates the effectiveness of PE through an example. Finally, Section VI concludes the paper. Due to space limitation, all proofs are omitted and can be found in [35]. Throughout the paper, let $\mathbb{N}$ and $\mathbb{P}$ denote, respectively, the sets of nonnegative and positive integers.

II. Problem Formulation

Consider a multi-hop network consisting of $N \geq 2$ nodes, connected by bidirectional links in a time-varying topology. The network is modeled as an undirected graph $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$, where $k \in \mathbb{N}$ denotes time, $\mathcal{V} = \{1, 2, \ldots, N\}$ represents the set of N nodes (vertices), and $\mathcal{E}(k) \subset \{\{i, j\} : i, j \in \mathcal{V}, i \neq j\}$ represents the nonempty set of links (edges) at time k. The graph $\mathcal{G}(k)$ is allowed to vary in order to reflect node mobility and changing channel conditions, and the variations are assumed to be exogenous, beyond control of the nodes. Any two nodes $i, j \in \mathcal{V}$ are one-hop neighbors and can communicate at time $k \in \mathbb{N}$ if and only if $\{i, j\} \in \mathcal{E}(k)$, and the communications are assumed to be delay- and error-free, with no quantization.

Suppose, at time $k = 0$, each node $i \in \mathcal{V}$ observes a function $f_i : \mathcal{X} \to \mathbb{R}$, which maps a nonempty open interval $\mathcal{X} \subset \mathbb{R}$ to $\mathbb{R}$, and which satisfies the following assumption:

Assumption 1. For each $i \in \mathcal{V}$, the function f_i is strictly convex, continuously differentiable, and has a minimizer $x_i^* \in \mathcal{X}$.

Note that the conditions in Assumption 1 are not redundant, as strict convexity alone does not imply continuous differentiability (e.g., with $\mathcal{X} = \mathbb{R}$, $f_i(x) = e^{|x|}$ is strictly convex but not differentiable at $x = 0$), and strict convexity and continuous differentiability together do not imply the existence of a minimizer in $\mathcal{X}$ (e.g., with $\mathcal{X} = (0, 1)$, $f_i(x) = e^{-x}$ is strictly convex and continuously differentiable but has no minimizer in $\mathcal{X}$). On the other hand, strict convexity and the existence of a minimizer $x_i^* \in \mathcal{X}$ do ensure that the minimizer x_i^* is unique.

Suppose, upon observing the f_i's, all the N nodes wish to solve the following unconstrained, separable, convex optimization problem:

$$\min_{x \in \mathcal{X}} F(x), \tag{2}$$

where the function $F : \mathcal{X} \to \mathbb{R}$ is defined as

$$F(x) = \sum_{i \in \mathcal{V}} f_i(x). \tag{3}$$

Notice that F in (3) is strictly convex and continuously differentiable, as these properties are preserved under summation. To show that F has a unique minimizer in $\mathcal{X}$ so that problem (2) is well-posed, let $f_i' : \mathcal{X} \to \mathbb{R}$ and $F' : \mathcal{X} \to \mathbb{R}$ denote the derivatives of f_i and F, respectively, and consider the following proposition:

Proposition 1. *With Assumption 1, there exists a unique $x^* \in \mathcal{X}$, which satisfies $F'(x^*) = 0$, minimizes F over $\mathcal{X}$, and solves problem* (2)*, i.e.,*

$$x^* = \arg\min_{x \in \mathcal{X}} F(x). \tag{4}$$

Given the above network and problem, the goal of this paper is to construct a distributed asynchronous algorithm, with which each node $i \in \mathcal{V}$ repeatedly communicates with its one-hop neighbors, iteratively updates its estimate $\hat{x}_i$ of the unknown optimizer x^* in (4), and asymptotically drives $\hat{x}_i$ to x^*. The algorithm should be easy to implement and free of limitations L1–L4 discussed in Section I.

III. Pairwise Equalizing

In this section, we develop a gossip algorithm having the aforementioned features.

Suppose, at time $k = 0$, each node $i \in \mathcal{V}$ creates a state variable $\hat{x}_i \in \mathcal{X}$ in its local memory, which represents its estimate of the unknown optimizer x^* in (4). Also suppose, at each subsequent time $k \in \mathbb{P}$, an iteration involving a subset of the N nodes, referred to as *iteration* k, takes place. Let $\hat{x}_i(0)$ represent the initial value of $\hat{x}_i$, and $\hat{x}_i(k)$ its value upon completing each iteration $k \in \mathbb{P}$. With this setup, the goal of asymptotically driving all the $\hat{x}_i(k)$'s to x^* may be stated as

$$\lim_{k \to \infty} \hat{x}_i(k) = x^*, \quad \forall i \in \mathcal{V}. \tag{5}$$

To design an algorithm that guarantees (5), consider a *conservation condition*

$$\sum_{i \in \mathcal{V}} f_i'(\hat{x}_i(k)) = 0, \quad \forall k \in \mathbb{N}, \tag{6}$$

which says that the state variables $\hat{x}_i(k)$'s evolve in such a manner that the sum of the derivatives f_i''s, evaluated respectively at the $\hat{x}_i(k)$'s, is always conserved at zero. Moreover, consider a *dissipation condition*

$$\lim_{k\to\infty} \hat{x}_i(k) = \tilde{x}, \quad \forall i \in \mathcal{V}, \text{ for some } \tilde{x} \in \mathcal{X}, \tag{7}$$

which says that the $\hat{x}_i(k)$'s gradually dissipate their differences and asymptotically achieve some arbitrary consensus $\tilde{x} \in \mathcal{X}$. Note that if the conservation condition (6) is met, then

$$\lim_{k\to\infty} \sum_{i\in\mathcal{V}} f_i'(\hat{x}_i(k)) = \lim_{k\to\infty} 0 = 0. \tag{8}$$

If, in addition, the dissipation condition (7) is met, then due to the continuity of every f_i' and (3),

$$\sum_{i\in\mathcal{V}} \lim_{k\to\infty} f_i'(\hat{x}_i(k)) = \sum_{i\in\mathcal{V}} f_i'(\lim_{k\to\infty} \hat{x}_i(k)) = \sum_{i\in\mathcal{V}} f_i'(\tilde{x}) = F'(\tilde{x}). \tag{9}$$

Because $\lim_{k\to\infty} f_i'(\hat{x}_i(k))$ exists $\forall i \in \mathcal{V}$, we can write

$$\lim_{k\to\infty} \sum_{i\in\mathcal{V}} f_i'(\hat{x}_i(k)) = \sum_{i\in\mathcal{V}} \lim_{k\to\infty} f_i'(\hat{x}_i(k)). \tag{10}$$

Combining (8), (9), and (10), we obtain $F'(\tilde{x}) = 0$. From Proposition 1, we see that the arbitrary consensus $\tilde{x}$ must be the unknown optimizer x^*, i.e., $\tilde{x} = x^*$, so that (5) holds. Therefore, to design an algorithm that ensures (5)—where x^* explicitly appears, it suffices to make the algorithm satisfy both the conservation and dissipation conditions (6) and (7)—where x^* is implicitly encoded.

To come up with such an algorithm, observe that the conservation condition (6) holds if and only if the initial values $\hat{x}_i(0)$'s are such that

$$\sum_{i\in\mathcal{V}} f_i'(\hat{x}_i(0)) = 0, \tag{11}$$

and the values $\hat{x}_i(k)$'s upon completing each iteration $k \in \mathbb{P}$ are related to the $\hat{x}_i(k-1)$'s prior to the iteration through

$$\sum_{i\in\mathcal{V}} f_i'(\hat{x}_i(k)) = \sum_{i\in\mathcal{V}} f_i'(\hat{x}_i(k-1)), \quad \forall k \in \mathbb{P}. \tag{12}$$

To satisfy (11), recall that every node $i \in \mathcal{V}$ knows the function f_i and knows that f_i has a unique minimizer $x_i^* \in \mathcal{X}$, which yields $f_i'(x_i^*) = 0$. Thus, (11) can be met by having every node $i \in \mathcal{V}$ compute x_i^* on its own and then initialize $\hat{x}_i(0)$ to x_i^*, i.e.,

$$\hat{x}_i(0) = x_i^*, \quad \forall i \in \mathcal{V}. \tag{13}$$

On the other hand, to satisfy (12), consider a gossip algorithm, whereby at each iteration $k \in \mathbb{P}$, a pair $u(k) = \{u_1(k), u_2(k)\} \in \mathcal{E}(k)$ of one-hop neighbors $u_1(k)$ and $u_2(k)$ communicate with each other and update their $\hat{x}_{u_1(k)}(k)$ and $\hat{x}_{u_2(k)}(k)$, while the rest of the N nodes stay idle and experience no change in their $\hat{x}_i(k)$'s, i.e.,

$$\hat{x}_i(k) = \hat{x}_i(k-1), \quad \forall k \in \mathbb{P}, \ \forall i \in \mathcal{V} - u(k). \tag{14}$$

Notice that with (14), equation (12) simplifies to

$$f'_{u_1(k)}(\hat{x}_{u_1(k)}(k)) + f'_{u_2(k)}(\hat{x}_{u_2(k)}(k)) = f'_{u_1(k)}(\hat{x}_{u_1(k)}(k-1)) + f'_{u_2(k)}(\hat{x}_{u_2(k)}(k-1)), \quad \forall k \in \mathbb{P}. \tag{15}$$

Also note that the entire expression (15) is known to nodes $u_1(k)$ and $u_2(k)$: $f'_{u_1(k)}$ and $f'_{u_2(k)}$ are derivatives of $f_{u_1(k)}$ and $f_{u_2(k)}$ they observe, $\hat{x}_{u_1(k)}(k-1)$ and $\hat{x}_{u_2(k)}(k-1)$ are "old" values of the state variables they maintain, and $\hat{x}_{u_1(k)}(k)$ and $\hat{x}_{u_2(k)}(k)$ are "new" values they seek to jointly determine, respectively. Hence, all that is needed for (12) to hold is a gossip between nodes $u_1(k)$ and $u_2(k)$ to share their $f_{u_1(k)}$, $f_{u_2(k)}$, $\hat{x}_{u_1(k)}(k-1)$, and $\hat{x}_{u_2(k)}(k-1)$, followed by a joint update of their $\hat{x}_{u_1(k)}(k)$ and $\hat{x}_{u_2(k)}(k)$, which ensures (15).

Obviously, (15) alone does not uniquely determine $\hat{x}_{u_1(k)}(k)$ and $\hat{x}_{u_2(k)}(k)$, since there are two variables but only one equation. This suggests that the available degree of freedom may be used to account for the dissipation condition (7), which has yet to be addressed. Unlike the conservation condition (6), however, the dissipation condition (7) is not about how the state variables $\hat{x}_i(k)$'s should evolve for every finite k. Instead, it is about where the $\hat{x}_i(k)$'s should approach as k goes to infinity, which nodes $u_1(k)$ and $u_2(k)$ cannot guarantee themselves since they are only responsible for two of the N $\hat{x}_i(k)$'s. Nevertheless, given that all the N $\hat{x}_i(k)$'s should approach the *same* limit, nodes $u_1(k)$ and $u_2(k)$ can help make this happen by imposing an *equalizing condition*, forcing $\hat{x}_{u_1(k)}(k)$ and $\hat{x}_{u_2(k)}(k)$ to be equal, i.e.,

$$\hat{x}_{u_1(k)}(k) = \hat{x}_{u_2(k)}(k), \quad \forall k \in \mathbb{P}. \tag{16}$$

With the equalizing condition (16) added, there are now two equations with two variables, providing nodes $u_1(k)$ and $u_2(k)$ a chance to uniquely determine $\hat{x}_{u_1(k)}(k)$ and $\hat{x}_{u_2(k)}(k)$ from (15) and (16).

The following proposition asserts that (15) and (16) always have a unique solution, so that the $\hat{x}_i(k)$'s are well-defined. To prove this assertion, the following lemma is useful:

Lemma 1. *Consider the network modeled in Section II. Suppose Assumption 1 holds. Then, for any $\mathcal{V}_s \subset \mathcal{V}$ with $\mathcal{V}_s \neq \emptyset$ and any $z_i \in \mathcal{X}$ for each $i \in \mathcal{V}_s$, there exists a unique $z \in \mathcal{X}$ such that $\sum_{i\in\mathcal{V}_s} f_i'(z) = \sum_{i\in\mathcal{V}_s} f_i'(z_i)$. Moreover, $z \in [\min_{i\in\mathcal{V}_s} z_i, \max_{i\in\mathcal{V}_s} z_i]$.*

Proposition 2. *With Assumption 1,* (13)*,* (14)*,* (15)*, and* (16)*, $\hat{x}_i(k)$ $\forall k \in \mathbb{N}$ $\forall i \in \mathcal{V}$ are well-defined, i.e., they are unambiguous and in $\mathcal{X}$. Furthermore,*

$$[\min_{i\in\mathcal{V}} \hat{x}_i(k), \max_{i\in\mathcal{V}} \hat{x}_i(k)] \subset [\min_{i\in\mathcal{V}} \hat{x}_i(k-1), \max_{i\in\mathcal{V}} \hat{x}_i(k-1)], \quad \forall k \in \mathbb{P}. \tag{17}$$

Lemma 1 and Proposition 2 call for a few remarks. First, Proposition 2 says that the $\hat{x}_i(k)$'s, besides being well-defined, must lie in a closed interval $[\min_{i\in\mathcal{V}} \hat{x}_i(k), \max_{i\in\mathcal{V}} \hat{x}_i(k)]$ that can only shrink or remain unchanged, as opposed to grow or drift, over time k. While this attribute does not guarantee the dissipation condition (7), it shows that the $\hat{x}_i(k)$'s are "trying" to converge to the same limit and are, at the very least, bounded even if $\mathcal{X}$ is not (e.g., $\mathcal{X} = \mathbb{R}$). Second, Lemma 1 implies that there is a unique $z \in \mathcal{X}$ such that

$$f'_{u_1(k)}(z) + f'_{u_2(k)}(z) = f'_{u_1(k)}(\hat{x}_{u_1(k)}(k-1)) + f'_{u_2(k)}(\hat{x}_{u_2(k)}(k-1)), \tag{18}$$

which turns out to satisfy

$$z \in [\min_{i\in u(k)} \hat{x}_i(k-1), \max_{i\in u(k)} \hat{x}_i(k-1)]. \tag{19}$$

Setting

$$\hat{x}_{u_1(k)}(k) = \hat{x}_{u_2(k)}(k) = z, \tag{20}$$

we see that (20) is a solution to (15) and (16). This suggests a simple, practical procedure for nodes $u_1(k)$ and $u_2(k)$ to solve (15) and (16) for $(\hat{x}_{u_1(k)}(k), \hat{x}_{u_2(k)}(k))$: apply a numerical *root-finding method*, such as the *bisection method* with an initial bracket provided in (19), to solve (18) for the unique z and then set both $\hat{x}_{u_1(k)}(k)$ and $\hat{x}_{u_2(k)}(k)$ to z as indicated in (20). Third, (19) and (20) also suggest that whenever the old $\hat{x}_{u_1(k)}(k-1)$ and $\hat{x}_{u_2(k)}(k-1)$ are equal, the new $\hat{x}_{u_1(k)}(k)$ and $\hat{x}_{u_2(k)}(k)$ must be equal to them, resulting in no change. Finally, since (18) always has a unique solution z, we may combine it with (20) and write

$$\hat{x}_{u_1(k)}(k) = \hat{x}_{u_2(k)}(k) = (f'_{u_1(k)} + f'_{u_2(k)})^{-1}(f'_{u_1(k)}(\hat{x}_{u_1(k)}(k-1)) + f'_{u_2(k)}(\hat{x}_{u_2(k)}(k-1))), \quad \forall k \in \mathbb{P}, \tag{21}$$

eliminating the intermediate variable z and stating the new $\hat{x}_{u_1(k)}(k)$ and $\hat{x}_{u_2(k)}(k)$ directly in terms of the old $\hat{x}_{u_1(k)}(k-1)$ and $\hat{x}_{u_2(k)}(k-1)$ and the function $(f'_i + f'_j)^{-1} : (f'_i + f'_j)(\mathcal{X}) \to \mathcal{X}$, which denotes the inverse of the injective function $f'_i + f'_j$ with its codomain restricted to its range.

Expressions (13), (14), and (21) collectively define a gossip-style, distributed asynchronous iterative algorithm, the operation of which leads to a switched, nonlinear, networked dynamical system

$$\hat{x}_i(k) = \begin{cases} (\sum_{j \in u(k)} f'_j)^{-1}(\sum_{j \in u(k)} f'_j(\hat{x}_j(k-1))), & \text{if } i \in u(k), \\ \hat{x}_i(k-1), & \text{otherwise,} \end{cases} \quad \forall k \in \mathbb{P},\ \forall i \in \mathcal{V}, \tag{22}$$

with initial condition (13), and with $(u(k))_{k=1}^{\infty}$ representing the sequence of gossiping nodes, which trigger the switchings. As this algorithm ensures the conservation condition (6), the state trajectory $(\hat{x}_1(k), \hat{x}_2(k), \ldots, \hat{x}_N(k))$ of the system (22) must remain on an $(N-1)$-dimensional manifold $\mathcal{M} \subset \mathcal{X}^N \subset \mathbb{R}^N$ defined as

$$\mathcal{M} = \{(x_1, x_2, \ldots, x_N) \in \mathcal{X}^N : \sum_{i \in \mathcal{V}} f'_i(x_i) = 0\}, \tag{23}$$

making $\mathcal{M}$ an invariant set. Motivated by the fact that the algorithm involves repeated, pairwise equalizing of the state variables, we refer to it as *Pairwise Equalizing* (PE). PE may be expressed in a compact algorithmic form capturing its communication and computational aspects as follows:

Algorithm 1 (Pairwise Equalizing)**.**
Initialization:
1) Each node $i \in \mathcal{V}$ computes $x_i^* \in \mathcal{X}$.
2) Each node $i \in \mathcal{V}$ creates a variable $\hat{x}_i \in \mathcal{X}$ and initializes it: $\hat{x}_i \leftarrow x_i^*$.

Operation: At each iteration:
3) A node with one or more one-hop neighbors, say, node i, initiates the iteration and selects a one-hop neighbor, say, node j, to gossip.
4) Nodes i and j select one of two ways to gossip by labeling themselves as either nodes a and b, or nodes b and a, respectively, where $\{a, b\} = \{i, j\}$.
5) If node b does not know f_a, then node a transmits f_a to node b.
6) Node a transmits $\hat{x}_a$ to node b.
7) Node b updates $\hat{x}_b$: $\hat{x}_b \leftarrow (f'_a + f'_b)^{-1}(f'_a(\hat{x}_a) + f'_b(\hat{x}_b))$.
8) Node b transmits $\hat{x}_b$ to node a.
9) Node a updates $\hat{x}_a$: $\hat{x}_a \leftarrow \hat{x}_b$. ■

Algorithm 1, or PE, consists of an *initialization* part that is executed once, and an *operation* part that is executed iteratively. Step 1 may be accomplished by letting every node $i \in \mathcal{V}$ calculate the root x_i^* of $f'_i(x_i^*) = 0$ analytically whenever possible (e.g., when $f_i(x) = x^2 + 2x + 3$), and numerically via a root-finding method whenever not (e.g., when $f_i(x) = x^2 + 2e^{-x} + 3e^x$). In the latter case, a suitable choice is the bisection method, which can also be used to carry out Step 7. Step 2 is intended to create the node estimates, or state variables, and initialize them using the result of Step 1. Step 3 may be realized either deterministically (e.g., each node periodically initiates an iteration and cyclically picks a neighbor) or stochastically (e.g., each node initiates an iteration according to some Poisson process and equiprobably picks a neighbor), depending on applications.

Step 4 is intended to let nodes i and j pick one of two ways to gossip that are equivalent mathematically, but different communicatively and computationally: notice from Steps 5–9 that the node that labels itself as node a has little to compute but has to communicate the function f_a once in Step 5, which consumes bandwidth and transmission power. In contrast, the node that labels itself as node b has not much to communicate but has to compute the update in Step 7, which demands processor time and effort. Thus, Step 4 offers nodes i and j an opportunity to take advantage of the asymmetry in their actions, to better utilize their communication and computational resources. For instance, if f_i requires fewer data symbols to represent—and, hence, less bandwidth and power to transmit—than f_j, or if node i's processor is slower or busier than node j's, then nodes i and j might want to label themselves as nodes a and b, as opposed to nodes b and a, respectively.

Steps 5 and 6 are introduced so that node b can perform Step 7, whereas Step 8 is introduced so that node a can perform Step 9. Note that Step 5 is a conditional step that is carried out if and only if the condition "node b does not know f_a" is true. For a wired network, this condition is true if and only if nodes i and j are gossiping or alternating their a-b labels for the first time, since the function f_a, upon reception by node b, could be stored in its local memory for later use. However, for a wireless network, this condition may be false even if nodes i and j are gossiping or alternating their a-b labels for the first time, since node b may have quietly learned about f_a by overhearing the wireless transmission of f_a from node a to another neighbor during a previous iteration. Observe that whenever the condition is false (which it almost always is), only *two* real-number transmissions are needed per iteration, in Steps 6 and 8.

Finally, notice that PE does not rely on a stepsize parameter to execute, nor does it require the existence and construction of a Hamiltonian cycle, as well as the concurrent use of a routing protocol for multi-hop transmissions. Indeed, all it essentially needs is that every node is capable of applying a root-finding method, maintaining a list of its one-hop neighbors, and remembering the functions it learns along the way. Therefore, PE successfully overcomes limitations L1–L3 facing the existing subgradient algorithms [1], [2], [17]–[28], while being rather easy to implement. A question that remains is whether it also circumvents L4, achieving asymptotic convergence.

Before answering this question, we point out that PE may be viewed as a natural generalization of three existing distributed averaging algorithms—namely, *Pairwise Averaging* [31], *Randomized Gossip Algorithm* [32], and *Anti-Entropy Aggregation* [33], [34]—to the convex optimization problem (2). To see this, consider a special case where each node $i \in \mathcal{V}$ observes not an arbitrary function f_i, but a quadratic

one of the form

$$f_i(x) = \frac{1}{2}(x - y_i)^2 + c_i, \tag{24}$$

with $\mathcal{X} = \mathbb{R}$ being its domain, and $y_i, c_i \in \mathbb{R}$ its parameters. Note from (24) that $f_i'(x) = x - y_i \ \forall i \in \mathcal{V}$; from the property $f_i'(x_i^*) = 0 \ \forall i \in \mathcal{V}$ that $x_i^* = y_i \ \forall i \in \mathcal{V}$; from (3) and (24) that $F'(x) = \sum_{i\in\mathcal{V}}(x - y_i)$; from the property $F'(x^*) = 0$ that $x^* = \frac{1}{N}\sum_{i\in\mathcal{V}} y_i$; from (13) that $\hat{x}_i(0) = y_i \ \forall i \in \mathcal{V}$; from (15) that $\hat{x}_{u_1(k)}(k) + \hat{x}_{u_2(k)}(k) = \hat{x}_{u_1(k)}(k-1) + \hat{x}_{u_2(k)}(k-1) \ \forall k \in \mathbb{P}$; and from (16) that $\hat{x}_{u_1(k)}(k) = \hat{x}_{u_2(k)}(k) = \frac{1}{2}(\hat{x}_{u_1(k)}(k-1) + \hat{x}_{u_2(k)}(k-1)) \ \forall k \in \mathbb{P}$. Thus, when the f_i's are given by (24), finding the unknown optimizer x^* amounts to calculating the network-wide average $\frac{1}{N}\sum_{i\in\mathcal{V}} y_i$ of the node "observations" y_i's, so that problem (2) becomes an averaging problem. Moreover, initializing the node estimates $\hat{x}_i(0)$'s simply means setting them to the y_i's, and equalizing $\hat{x}_{u_1(k)}(k)$ and $\hat{x}_{u_2(k)}(k)$ simply means averaging them, so that PE reduces to the three aforementioned algorithms, which share the same initialization and update rules. Furthermore, in this special case, the invariant manifold $\mathcal{M}$ in (23) becomes the invariant hyperplane $\mathcal{M} = \{(x_1, x_2, \ldots, x_N) \in \mathbb{R}^N : \sum_{i\in\mathcal{V}} x_i = \sum_{i\in\mathcal{V}} y_i\}$ that plays an important role in the study of distributed averaging [31]–[34], [36]–[39].

IV. Convergence Analysis

In Section III, we showed that PE ensures the conservation condition (6) and *attempts* to satisfy the dissipation condition (7). In this section, using Lyapunov stability theory, we show that PE does guarantee the latter, thereby assuring (5), as long as the gossiping pattern is sufficiently rich.

For convenience, let $\mathbf{x}^*$ and $\mathbf{x}(k) \ \forall k \in \mathbb{N}$ denote, respectively, the vectors obtained by stacking N copies of x^* and all the $\hat{x}_i(k)$'s, i.e., $\mathbf{x}^* = (x^*, x^*, \ldots, x^*)$ and $\mathbf{x}(k) = (\hat{x}_1(k), \hat{x}_2(k), \ldots, \hat{x}_N(k))$. Then, from Propositions 1 and 2, $\mathbf{x}^* \in \mathcal{X}^N$ and $\mathbf{x}(k) \in \mathcal{X}^N \ \forall k \in \mathbb{N}$. In addition, due to (22), if $\mathbf{x}(k) = \mathbf{x}^*$ for some $k \in \mathbb{N}$, then $\mathbf{x}(\ell) = \mathbf{x}^* \ \forall \ell > k$. Hence, $\mathbf{x}^*$ is an equilibrium point of the system (22). To show that $\mathbf{x}(k)$ would asymptotically converge to the equilibrium point $\mathbf{x}^*$, i.e., (5) holds, we seek to construct a Lyapunov function. To this end, recall that for any strictly convex and differentiable function $f : \mathcal{X} \to \mathbb{R}$, the first-order convexity condition [30] says that

$$f(y) \geq f(x) + f'(x)(y - x), \quad \forall x, y \in \mathcal{X}, \tag{25}$$

where the equality holds if and only if $x = y$. This suggests the following Lyapunov function candidate $V : \mathcal{X}^N \subset \mathbb{R}^N \to \mathbb{R}$, which exploits the convexity of the f_i's:

$$V(\mathbf{x}(k)) = \sum_{i\in\mathcal{V}} f_i(x^*) - f_i(\hat{x}_i(k)) - f_i'(\hat{x}_i(k))(x^* - \hat{x}_i(k)). \tag{26}$$

Notice that V in (26) is well-defined because $\mathbf{x}^* \in \mathcal{X}^N$, $\mathbf{x}(k) \in \mathcal{X}^N \ \forall k \in \mathbb{N}$, and f_i' is well-defined $\forall i \in \mathcal{V}$. Moreover, because of Assumption 1 and the first-order convexity condition (25), V is continuous and positive definite with respect to $\mathbf{x}^*$, i.e.,

$$V(\mathbf{x}(k)) \geq 0, \quad \forall \mathbf{x}(k) \in \mathcal{X}^N, \tag{27}$$

where the equality holds if and only if $\mathbf{x}(k) = \mathbf{x}^*$. Therefore, to prove (5), it suffices to show that

$$\lim_{k\to\infty} V(\mathbf{x}(k)) = 0. \tag{28}$$

The following lemma represents the first step toward establishing (28):

Lemma 2. *Consider the network modeled in Section II and the use of PE described in Algorithm 1. Suppose Assumption 1 holds. Then, for any given sequence $(u(k))_{k=1}^{\infty}$, the sequence $(V(\mathbf{x}(k)))_{k=0}^{\infty}$ is non-increasing and satisfies*

$$V(\mathbf{x}(k)) - V(\mathbf{x}(k-1)) = -\sum_{i\in u(k)} f_i(\hat{x}_i(k)) - f_i(\hat{x}_i(k-1)) - f_i'(\hat{x}_i(k-1))(\hat{x}_i(k) - \hat{x}_i(k-1)), \quad \forall k \in \mathbb{P}. \tag{29}$$

Lemma 2 has several implications. First, according to (22), (25), and (29), upon completing each iteration $k \in \mathbb{P}$ by any two nodes $u_1(k)$ and $u_2(k)$, the value of V must either decrease from $V(\mathbf{x}(k-1))$ to $V(\mathbf{x}(k))$ or, at worst, stay the same with $V(\mathbf{x}(k)) = V(\mathbf{x}(k-1))$, where the latter occurs if and only if $\hat{x}_{u_1(k)}(k-1) = \hat{x}_{u_2(k)}(k-1)$. Therefore, with PE, every node may freely decide when to initiate an iteration and who to gossip with, knowing that no matter what it does, the value of V cannot increase. Second, since $(V(\mathbf{x}(k)))_{k=0}^{\infty}$ is non-increasing irrespective of $(u(k))_{k=1}^{\infty}$, the function V in (26) may be regarded as a *common* Lyapunov function for the nonlinear switched system (22), which has as many as $\frac{N(N-1)}{2}$ different dynamics, corresponding to the $\frac{N(N-1)}{2}$ possible gossiping pairs in the network. Finally, Lemma 2 suggests that the first-order convexity condition (25) can be used not only to form the common Lyapunov function V in (26), but also to characterize drops in its value in (29) after every gossip. This is akin to how quadratic functions may be used to form a common Lyapunov function $V(k) = x^T(k)Px(k)$ for a linear switched system $x(k+1) = A(k)x(k)$, $A(k) \in \{A_1, A_2, \ldots, A_M\}$, as well as to characterize drops in $V(k)$ via $V(k+1) - V(k) = x^T(k)(A_i^T P A_i - P)x(k) = -x^T(k)Q_i x(k)$. Indeed, as we will show later, when problem (2) specializes to an averaging problem, for which the nonlinear switched system (22) reduces to a linear one, both V in (26) and its drop in (29) become quadratic functions.

As it follows from (27) and Lemma 2, the sequence $(V(\mathbf{x}(k)))_{k=0}^{\infty}$ is nonnegative and non-increasing, implying that $\lim_{k\to\infty} V(\mathbf{x}(k))$ exists and is nonnegative. This, however, is insufficient for us to conclude that $\lim_{k\to\infty} V(\mathbf{x}(k))$ is zero, since, for some pathological gossiping patterns, $\lim_{k\to\infty} V(\mathbf{x}(k))$ can be positive. To see this, suppose the set $\mathcal{V}$ of nodes can be partitioned into two nonempty subsets, such that the nodes in one subset never gossip with those in the other—either by force (e.g., $\mathcal{V} = \{1,2,3,4\}$ and $\mathcal{E}(k) \equiv \{\{1,2\},\{3,4\}\}$, so that $u(k)$ is forced to be $\{1,2\}$ or $\{3,4\}$) or by choice (e.g., $\mathcal{V} = \{1,2,3,4\}$ and $\mathcal{E}(k) \equiv \{\{1,2\},\{2,3\},\{3,4\}\}$, but $u(k)$ is chosen to be $\{1,2\}$ or $\{3,4\}$). Then, $V(\mathbf{x}(k))$ in general would be bounded away from zero by a positive constant, since x^* in (4) depends on all the f_i's, but information never flows between the subsets. Thus, some restrictions must be imposed on the gossiping pattern, in order to establish (28).

Given that PE—or, specifically, its Step 3—may be realized either *deterministically* or *stochastically*, we will introduce restrictions on the gossiping pattern in both of these frameworks. Moreover, since the sequence $(\mathcal{E}(k))_{k=0}^{\infty}$ was assumed in Section II to be exogenous, below we will treat $(\mathcal{E}(k))_{k=0}^{\infty}$ simply as given, regardless of the frameworks.

In the *deterministic* framework, suppose each node initiates an iteration and picks a neighbor to gossip according to some deterministic policy, resulting in a deterministic sequence $(u(k))_{k=1}^{\infty}$, which must satisfy $u(k) \in \mathcal{E}(k) \ \forall k \in$

$\mathbb{P}$. For any given $(u(k))_{k=1}^{\infty}$, define the set $\mathcal{E}_\infty \subset \{\{i,j\} : i,j \in \mathcal{V}, i \neq j\}$ as

$$\mathcal{E}_\infty = \{\{i,j\} : u(k) = \{i,j\} \text{ for infinitely many } k \in \mathbb{P}\}. \quad (30)$$

Equation (30) says that a link $\{i,j\}$ is in $\mathcal{E}_\infty$ if and only if nodes i and j gossip with each other infinitely often. With $\mathcal{E}_\infty$ defined as such, we may state the following restriction on the gossiping pattern, which is similar to the connectivity assumption adopted in [22], [27]:

Assumption 2. The deterministic sequence $(u(k))_{k=1}^{\infty}$ is such that the graph $(\mathcal{V}, \mathcal{E}_\infty)$ is connected.

Assumption 2 is not difficult to satisfy in practice, provided that the network is "connected in the long run." To justify this claim, consider the exogenous sequence $(\mathcal{E}(k))_{k=0}^{\infty}$ and let $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_M$ represent the sets of links that occur infinitely often in $(\mathcal{E}(k))_{k=0}^{\infty}$, i.e., for each $\ell \in \{1,2,\ldots,M\}$, $\mathcal{E}(k) = \mathcal{E}_\ell$ for infinitely many k's. Note that if the graph $(\mathcal{V}, \cup_{\ell=1}^{M} \mathcal{E}_\ell)$ is not connected, it means that the set $\mathcal{V}$ of nodes can be partitioned into two nonempty subsets $\mathcal{V}_1$ and $\mathcal{V}_2$, such that after some finite time, the nodes in $\mathcal{V}_1$ can no longer gossip with those in $\mathcal{V}_2$, even if they want to. Thus, we may say that the network is *connected in the long run* if and only if the graph $(\mathcal{V}, \cup_{\ell=1}^{M} \mathcal{E}_\ell)$ is connected. Now suppose the graph $(\mathcal{V}, \cup_{\ell=1}^{M} \mathcal{E}_\ell)$ is connected. Also suppose $\mathcal{E}(k)$ is slowly varying, in the sense that it is constant for many consecutive k's. This assumption is reasonable because the topology of a network typically changes at a rate that is much slower compared to the rate at which iterations can occur (e.g., in a wireless network, although path losses and shadowing may cause a link to fail or recover, such a change occurs at a much slower time scale compared to the propagation of electromagnetic waves). Since the graph $(\mathcal{V}, \cup_{\ell=1}^{M} \mathcal{E}_\ell)$ is connected and $\mathcal{E}(k)$ is slowly varying, if we simply let every possible pair of one-hop neighbors gossip frequently enough—at least once per change in $\mathcal{E}(k)$—then $\mathcal{E}_\infty = \cup_{\ell=1}^{M} \mathcal{E}_\ell$, so that Assumption 2 holds. Therefore, as long as the network is connected in the long run, Assumption 2 can be easily met.

In the previous paragraph, if the graph $(\mathcal{V}, \cup_{\ell=1}^{M} \mathcal{E}_\ell)$ is not connected, then for every $i \in \mathcal{V}_1$ and $j \in \mathcal{V}_2$, we have $\{i,j\} \notin \mathcal{E}_\infty$. This implies that the graph $(\mathcal{V}, \mathcal{E}_\infty)$ is also not connected, so that Assumption 2 fails. In this case, PE generally would fail to asymptotically converge, but so would most distributed iterative algorithms, including the consensus algorithms in [3], [4], [6], [8]–[10], [12], [14], [15], as well as the averaging algorithms in [31]–[34], [36]–[39].

Based on Assumption 2, the following theorem can be established:

Theorem 1. *Consider the network modeled in Section II and the use of PE described in Algorithm 1. Suppose Assumptions 1 and 2 hold. Then,* (28) *and* (5) *hold.*

Theorem 1 says that, under Assumption 2 on the gossiping pattern, PE ensures asymptotic convergence of all the $\hat{x}_i(k)$'s to x^*, circumventing limitation L4 facing many of the existing subgradient algorithms.

Next, in the *stochastic* framework, suppose each node initiates an iteration and picks a neighbor to gossip according to some random strategy, resulting in a random sequence $(u(k))_{k=1}^{\infty}$, which satisfies $u(k) \in \mathcal{E}(k)\ \forall k \in \mathbb{P}$, and which is independent, but not necessarily identically distributed, over time k. For each $k \in \mathbb{P}$ and each $\{i,j\} \in \mathcal{E}(k)$, let $p_{\{i,j\}}(k) \in [0,1]$ denote the probability of $u(k)$ being $\{i,j\}$. In addition, for each $\{i,j\} \notin \mathcal{E}(k)$, let $p_{\{i,j\}}(k)$ be undefined since the event $u(k) = \{i,j\}$ cannot happen. For any given $p_{\{i,j\}}(k)\ \forall k \in \mathbb{P}\ \forall \{i,j\} \in \mathcal{E}(k)$, define the set $\tilde{\mathcal{E}}_\infty \subset \{\{i,j\} : i,j \in \mathcal{V}, i \neq j\}$ as

$$\tilde{\mathcal{E}}_\infty = \{\{i,j\} : \exists \varepsilon > 0 \text{ such that } \forall k \in \mathbb{P}, \; p_{\{i,j\}}(\ell) \geq \varepsilon \text{ for some } \ell > k\}. \quad (31)$$

Expression (31) says that a link $\{i,j\}$ is in $\tilde{\mathcal{E}}_\infty$ if and only if the probability with which nodes i and j gossip with each other is no less than a positive constant ε for infinitely many iterations. In other words, $\{i,j\} \in \tilde{\mathcal{E}}_\infty$ if and only if the sequence $(p_{\{i,j\}}(k))_{k=1}^{\infty}$ has a subsequence whose elements are no less than ε. For instance, if $p_{\{i,j\}}(k) = \frac{1}{k}\ \forall k \in \mathbb{P}$, then $\{i,j\} \notin \tilde{\mathcal{E}}_\infty$. In contrast, if

$$(p_{\{i,j\}}(k))_{k=1}^{\infty} = (0.1, \underbrace{\#, \ldots, \#}_{10 \text{ times}}, 0.1, \underbrace{\#, \ldots, \#}_{100 \text{ times}}, 0.1, \underbrace{\#, \ldots, \#}_{1000 \text{ times}}, \ldots),$$

where $\#$ represents either zero or "undefined," then $\{i,j\} \in \tilde{\mathcal{E}}_\infty$. Based on this definition of $\tilde{\mathcal{E}}_\infty$, we may introduce the following restriction on the random gossiping pattern:

Assumption 3. The random sequence $(u(k))_{k=1}^{\infty}$ is such that the graph $(\mathcal{V}, \tilde{\mathcal{E}}_\infty)$ is connected.

Similar to Assumption 2, it is not difficult to satisfy Assumption 3, so long that the network is connected in the long run. To explain this, suppose the graph $(\mathcal{V}, \cup_{\ell=1}^{M} \mathcal{E}_\ell)$ is connected. Note that at each time $k \in \mathbb{P}$ and for each node $i \in \mathcal{V}$ who has one or more one-hop neighbors at time k, if we simply let the probabilities $\mathrm{P}\{\text{node } i \text{ initiates iteration } k\}$ be no less than some $\varepsilon_1 > 0$ and $\mathrm{P}\{\text{node } i \text{ picks node } j \text{ to gossip} \,|\, \text{node } i \text{ initiates iteration } k\}$ be no less than some $\varepsilon_2 > 0$, then $p_{\{i,j\}}(k) \geq 2\varepsilon_1\varepsilon_2\ \forall k \in \mathbb{P}\ \forall \{i,j\} \in \mathcal{E}(k)$. This implies that $\tilde{\mathcal{E}}_\infty = \cup_{\ell=1}^{M} \mathcal{E}_\ell$, so that Assumption 3 is met, explaining the argument.

With Assumption 3, the following stochastic version of Theorem 1 can be stated:

Theorem 2. *Consider the network modeled in Section II and the use of PE described in Algorithm 1. Suppose Assumptions 1 and 3 hold. Then, with probability* 1*,* (28) *and* (5) *hold.*

Theorem 2 shows that, under Assumption 3 on the random gossiping pattern, PE is almost surely asymptotically convergent, again overcoming limitation L4.

Finally, reconsider the special case where the f_i's are as in (24), i.e., where problem (2) is an averaging problem. In this case, the common Lyapunov function V in (26) takes a quadratic form:

$$V(\mathbf{x}(k)) = \frac{1}{2}\sum_{i \in \mathcal{V}} (\hat{x}_i(k) - x^*)^2 = \frac{1}{2}(\mathbf{x}(k) - \mathbf{x}^*)^T P (\mathbf{x}(k) - \mathbf{x}^*), \quad (32)$$

where x^* is the network-wide average $\frac{1}{N}\sum_{i \in \mathcal{V}} y_i$, and $P \in \mathbb{R}^{N \times N}$ is the identity matrix. In addition, because of the linear, pairwise averaging update $\hat{x}_{u_1(k)}(k) = \hat{x}_{u_2(k)}(k) = \frac{1}{2}(\hat{x}_{u_1(k)}(k-1) + \hat{x}_{u_2(k)}(k-1))\ \forall k \in \mathbb{P}$, the drop $V(\mathbf{x}(k)) -$

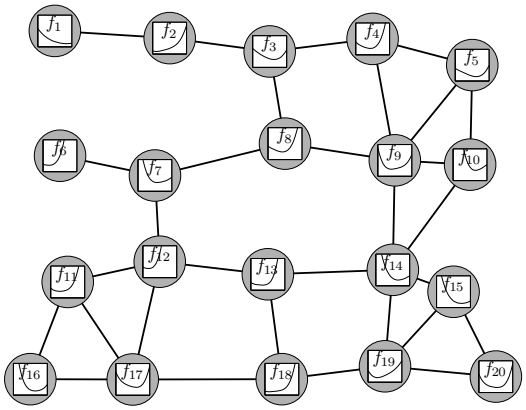

Fig. 1. A 20-node, 30-link network with each node i observing f_i.

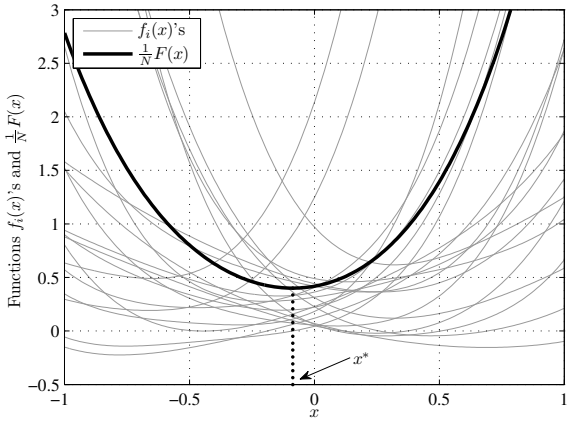

Fig. 2. Graphs of the functions f_i's and $\frac{1}{N}F$, along with the unknown optimizer x^*.

$V(\mathbf{x}(k-1))$ in (29) also takes a quadratic form:

$$V(\mathbf{x}(k))-V(\mathbf{x}(k-1))=-\frac{1}{4}(\hat{x}_{u_1(k)}(k-1)-\hat{x}_{u_2(k)}(k-1))^2$$
$$=-\frac{1}{2}\mathbf{x}^T(k-1)Q_{u(k)}\mathbf{x}(k-1), \quad \forall k\in\mathbb{P},$$

where $Q_{\{i,j\}}\in\mathbb{R}^{N\times N}$ is a symmetric positive semidefinite matrix whose ii and jj entries are $\frac{1}{2}$, ij and ji entries are $-\frac{1}{2}$, and all other entries are zero. Therefore, the first-order-convexity-condition-based Lyapunov function (26) may be viewed as a natural generalization of the quadratic Lyapunov function (32) for distributed averaging to the convex optimization problem (2).

V. Illustrative Example

In this section, we illustrate the effectiveness of PE via a simple example.

Consider a network of 20 nodes, connected by 30 links in a fixed topology, as shown in Figure 1. Suppose each node i observes a function $f_i:\mathbb{R}\to\mathbb{R}$, given by

$$f_i(x)=a_ix+b_i(x-c_i)^2+d_i(x-e_i)^4, \qquad (33)$$

where a_i,b_i,c_i,d_i,e_i are parameters of f_i, whose values are randomly chosen from the intervals $(-1,1),(0,1),(-1,1),(0,2),(-1,1)$. The f_i's in (33) fulfill Assumption 1 because the b_i's and d_i's are positive. To visualize these f_i's, their graphs are displayed as thumbnails in Figure 1 and superimposed in Figure 2. Also depicted in Figure 2 are the graph of the function F, scaled by $\frac{1}{N}$ so that it fits into the figure, and the unknown optimizer x^* of F, that all the nodes wish to determine.

Suppose the nodes apply PE and carry out its Step 3 stochastically, such that every pair of one-hop neighbors has equal probability (i.e., $\frac{1}{30}$) of being the pair $u(k)$ that gossips at iteration k, for every k. By simulating PE for 1200 iterations, a realization of the random sequence $(u(k))_{k=1}^{1200}$ of gossiping pairs has been obtained (not shown here). Figure 3 shows, on a logarithmic scale, the value $V(\mathbf{x}(k))$ of the common Lyapunov function along the state trajectory $\mathbf{x}(k)$ of the system. Note that $V(\mathbf{x}(k))$ is indeed non-increasing, agreeing with Lemma 2. Moreover, it is converging to zero, at a rate that is roughly exponential. Figure 4 shows the individual components $\hat{x}_i(k)$'s of $\mathbf{x}(k)$, which represent the estimates of x^*. Observe that the $\hat{x}_i(k)$'s gradually approach x^*, converging to $x^*\pm 0.005$ after 1008 iterations. Furthermore,

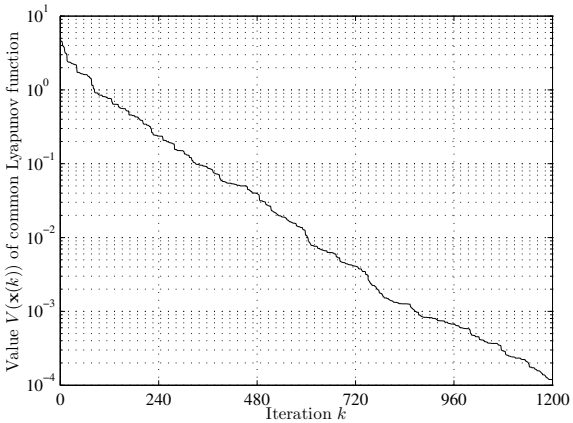

Fig. 3. Convergence of the value $V(\mathbf{x}(k))$ of the common Lyapunov function to zero.

the closed interval $[\min_i \hat{x}_i(k), \max_i \hat{x}_i(k)]$ indeed can only shrink or remain unchanged, concurring with Proposition 2.

Notice that the network in Figure 1 contains no Hamiltonian cycle. Hence, it may be difficult to apply the subgradient algorithms mentioned in L2. Also, if the nodes are not fully aware of one another, or if they do not have a routing protocol, then the same can be said about the subgradient algorithms mentioned in L3, since it may be difficult to randomly and equiprobably pass the latest estimate of x^* among the nodes. In fact, even if such passing can be realized, each pass requires, on average, 2.98 real-number transmissions (or hops) to complete if shortest-path routing is used, and a higher number if it is not, or if the network diameter were larger. In comparison, although PE requires, in its Step 5, one-time transmissions of the f_i's as communication overhead, it requires only 2 real-number transmissions per iteration, regardless of the network size and topology.

VI. Conclusion

In this paper, we have addressed the problem of achieving unconstrained, separable, convex consensus optimization over undirected networks with time-varying topologies, where each observed component function is strictly convex, continuously differentiable, and has a minimizer. Based on the ideas of conservation, dissipation, and equalizing, we have developed *Pairwise Equalizing* (PE), a gossip-style, distributed asynchronous iterative algorithm, which enables

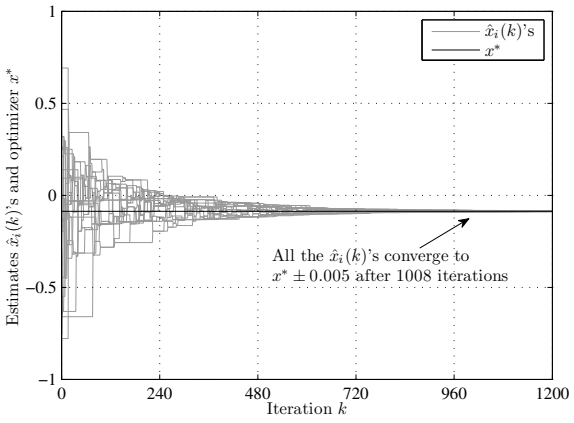

Fig. 4. Convergence of the estimates $\hat{x}_i(k)$'s to the optimizer x^*.

nodes to cooperatively solve the problem, in a way that is fundamentally different from the subgradient algorithms. Using Lyapunov stability theory, we have shown that the switched, nonlinear, networked dynamical system induced by PE is deterministically and stochastically asymptotically convergent, provided that the gossiping pattern is rich enough. In the analysis, we have utilized the first-order convexity condition to construct a common Lyapunov function and characterize drops in its value. We have also shown that PE is easy to implement and bypasses limitations facing the subgradient algorithms. Finally, we have shown that PE reduces to Pairwise Averaging and Randomized Gossip Algorithm in a special case. Given these appealing features of PE, it may be recommended for a wide range of applications.

REFERENCES

[1] S.-H. Son, M. Chiang, S. R. Kulkarni, and S. C. Schwartz, "The value of clustering in distributed estimation for sensor networks," in *Proc. International Conference on Wireless Networks, Communications and Mobile Computing*, Maui, HI, 2005, pp. 969–974.

[2] M. G. Rabbat and R. D. Nowak, "Distributed optimization in sensor networks," in *Proc. International Symposium on Information Processing in Sensor Networks*, Berkeley, CA, 2004, pp. 20–27.

[3] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.

[4] J. N. Tsitsiklis and M. Athans, "Convergence and asymptotic agreement in distributed decision problems," *IEEE Transactions on Automatic Control*, vol. 29, no. 1, pp. 42–50, 1984.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[6] ——, "Some aspects of parallel and distributed iterative algorithms—a survey," *Automatica*, vol. 27, no. 1, pp. 3–21, 1991.

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

[8] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[9] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[10] Y. Hatano and M. Mesbahi, "Agreement over random networks," *IEEE Transactions on Automatic Control*, vol. 50, no. 11, pp. 1867–1872, 2005.

[11] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.

[12] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[13] J. Cortés, "Finite-time convergent gradient flows with applications to network consensus," *Automatica*, vol. 42, no. 11, pp. 1993–2000, 2006.

[14] L. Fang and P. J. Antsaklis, "On communication requirements for multi-agent consensus seeking," in *Networked Embedded Sensing and Control*, ser. Lecture Notes in Control and Information Sciences, P. J. Antsaklis and P. Tabuada, Eds. Berlin, Germany: Springer-Verlag, 2006, vol. 331, pp. 53–67.

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

[16] S. Sundaram and C. N. Hadjicostis, "Finite-time distributed consensus in graphs with time-invariant topologies," in *Proc. American Control Conference*, New York, NY, 2007, pp. 711–716.

[17] A. Nedić and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 109–138, 2001.

[18] A. Nedić, D. P. Bertsekas, and V. S. Borkar, "Distributed asynchronous incremental subgradient methods," in *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*, D. Butnariu, Y. Censor, and S. Reich, Eds. Amsterdam, Holland: Elsevier, 2001, pp. 381–407.

[19] A. Nedić and D. P. Bertsekas, "Convergence rate of incremental subgradient algorithms," in *Stochastic Optimization: Algorithms and Applications*, S. P. Uryasev and P. M. Pardalos, Eds. Norwell, MA: Kluwer Academic Publishers, 2001, pp. 223–264.

[20] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 798–808, 2005.

[21] B. Johansson, M. Rabi, and M. Johansson, "A simple peer-to-peer algorithm for distributed optimization in sensor networks," in *Proc. IEEE Conference on Decision and Control*, New Orleans, LA, 2007, pp. 4705–4710.

[22] A. Nedić and A. Ozdaglar, "On the rate of convergence of distributed subgradient methods for multi-agent optimization," in *Proc. IEEE Conference on Decision and Control*, New Orleans, LA, 2007, pp. 4711–4716.

[23] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Stochastic incremental gradient descent for estimation in sensor networks," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, 2007, pp. 582–586.

[24] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson, "Subgradient methods and consensus algorithms for solving convex optimization problems," in *Proc. IEEE Conference on Decision and Control*, Cancun, Mexico, 2008, pp. 4185–4190.

[25] I. Lobel and A. Ozdaglar, "Convergence analysis of distributed subgradient methods over random networks," in *Proc. Allerton Conference on Communication, Control, and Computing*, Monticello, IL, 2008, pp. 353–360.

[26] A. Nedić, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "Distributed subgradient methods and quantization effects," in *Proc. IEEE Conference on Decision and Control*, Cancun, Mexico, 2008, pp. 4177–4184.

[27] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[28] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Incremental stochastic subgradient algorithms for convex optimization," *SIAM Journal on Optimization*, vol. 20, no. 2, pp. 691–717, 2009.

[29] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY: W. H. Freeman, 1979.

[30] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY: Cambridge University Press, 2004.

[31] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, 1984.

[32] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[33] M. Jelasity and A. Montresor, "Epidemic-style proactive aggregation in large overlay networks," in *Proc. IEEE International Conference on Distributed Computing Systems*, Tokyo, Japan, 2004, pp. 102–109.

[34] A. Montresor, M. Jelasity, and O. Babaoglu, "Robust aggregation protocols for large-scale overlay networks," in *Proc. IEEE/IFIP International Conference on Dependable Systems and Networks*, Florence, Italy, 2004, pp. 19–28.

[35] J. Lu, C. Y. Tang, P. R. Regier, and T. D. Bow, "A gossip algorithm for convex consensus optimization over networks," submitted for publication in *IEEE Transactions on Automatic Control*, 2010.

[36] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[37] J.-Y. Chen, G. Pandurangan, and D. Xu, "Robust computation of aggregates in wireless sensor networks: Distributed randomized algorithms and analysis," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 9, pp. 987–1000, 2006.

[38] M. Mehyar, D. Spanos, J. Pongsajapan, S. H. Low, and R. M. Murray, "Asynchronous distributed averaging on communication networks," *IEEE/ACM Transactions on Networking*, vol. 15, no. 3, pp. 512–520, 2007.

[39] C. Y. Tang and J. Lu, "Controlled hopwise averaging: Bandwidth/energy-efficient asynchronous distributed averaging for wireless networks," in *Proc. American Control Conference*, St. Louis, MO, 2009, pp. 1561–1568.