Adaptive Penalty-Based Distributed Stochastic Convex Optimization

Zaid J. Towfic, *Student Member, IEEE*, and Ali H. Sayed, *Fellow, IEEE*

***Abstract*—In this work, we study the task of distributed optimization over a network of learners in which each learner possesses a convex cost function, a set of affine equality constraints, and a set of convex inequality constraints. We propose a fully distributed adaptive diffusion algorithm based on penalty methods that allows the network to cooperatively optimize the global cost function, which is defined as the sum of the individual costs over the network, subject to all constraints. We show that when small constant step-sizes are employed, the expected distance between the optimal solution vector and that obtained at each node in the network can be made arbitrarily small. Two distinguishing features of the proposed solution relative to other approaches is that the developed strategy does not require the use of projections and is able to track drifts in the location of the minimizer due to changes in the constraints or in the aggregate cost itself. The proposed strategy is able to cope with changing network topology, is robust to network disruptions, and does not require global information or rely on central processors.**

***Index Terms*—Adaptation and learning, consensus strategies, constrained optimization, diffusion strategies, distributed processing, penalty method.**

I. INTRODUCTION

DISTRIBUTED convex optimization refers to the task of minimizing the aggregate sum of convex cost functions, each available at an agent of a connected network, subject to convex constraints that are also distributed across the agents. The key challenge in such problems is that each agent is only aware of its cost function and its constraints. This article proposes a *fully* decentralized solution that is able to minimize the aggregate cost function while satisfying all distributed constraints. The solution method is based solely on local cooperation among neighboring nodes and does not rely on the use of projection constructions. Furthermore, the individual nodes do not need to know any of the constraints besides their own.

There have been several useful studies on distributed convex optimization techniques in the literature [2]–[21]. Most existing techniques are suitable for the solution of *static* optimization problems, where the objective is to determine the location of a fixed optimal parameter. The available solution methods tend to employ constructions that become problematic in the context of adaptation and learning over networks. This is because they often rely on the use of decaying step-sizes in their stochastic gradient updates [4], [5], [20], [21]. Such decaying step-sizes are a hindrance to adaptation when it is desired to develop *dynamic* or adaptive solutions that are able to track *drifts* in the location of the optimal parameter; these drifts can result from changes in the constraint conditions or in the cost functions themselves. For this reason, in this work, we employ *constant* step-sizes in order to enable continuous adaptation and learning [22]–[24].

When constant step-sizes are used, the dynamics of the distributed algorithm is changed in a nontrivial manner and its convergence analysis becomes more demanding because, as we are going to see, the gradient update term does not die out anymore with time as happens with decaying step-size implementations. In the constant step-size case, gradient noise will always be present and will seep into the update equations. Nevertheless, we will show that the proposed distributed strategy can still ensure approximation errors of the order of the step-size so that arbitrarily small levels of accuracy can be attained by using sufficiently small step-sizes for slowly changing optimizers (see Theorem 2). It is well-known that there is an inherent trade-off between tracking ability and accuracy in the estimation process [25, pp. 280–283], namely, smaller step-sizes improve estimation accuracy and larger step-sizes enhance tracking. The main conclusion though is that if the optimizer is changing slowly over time, then a stochastic-gradient algorithm that utilizes small constant step-sizes can both track well and estimate the optimizer with good accuracy.

Most available distributed solutions for convex optimization problems also tend to rely on the use of projection operators in order to ensure that the successive estimates at the nodes satisfy the convex constraints [4], [19]–[21], [26]. In some of the methods [20], [21], each node is required to know all the constraints across the entire network in order to compute the necessary projections. This requirement defeats the purpose of a distributed solution since it requires the nodes to have access to global information. The works [4], [26] develop useful distributed solutions where nodes are only required to know their own constraints. However, the constraint conditions still need to be relatively simple in order for the distributed algorithm to be able to compute the necessary projections analytically (such as projecting onto the nonnegative orthant). In cases when the constraints are more complex so that the necessary projections are

Manuscript received December 13, 2013; revised April 09, 2014; accepted May 25, 2014. Date of publication June 18, 2014; date of current version July 10, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Shuguang (Robert) Cui. This work was supported in part by NSF grant CCF-1011918. A short version of this work appears in the Allerton Conference, Allerton, IL, USA, October 2013 [1].

The authors are with Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: ztowfic@ucla.edu; sayed@ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2014.2331615

not easily computed, then several of the existing techniques tend to implement an *offline* optimization routine that is guaranteed to converge only asymptotically, and not in a finite number of steps, as explained in [27], [28]. The analysis for these methods generally assumes that the projection step is implemented ideally even though the offline iterations are in fact truncated in practice and the truncation errors interfere with the accuracy of the distributed solution.

Motivated by the above considerations, in this work, we propose a distributed solution that employs constant step-sizes and does not require projections. The solution relies on the use of suitably chosen *penalty* functions and replaces the projection step by a stochastic approximation update that runs *simultaneously* with the optimization step. The challenge is to show that the use of penalty functions in the stochastic gradient update step still leads to reasonable solutions. The analysis in the article establishes that this is indeed possible. In particular, we show following Theorem 2 further ahead how to select the parameters of the proposed algorithm in order to ensure desirable convergence properties with small approximation errors. Moreover, in the proposed solution, the nodes are only required to interact locally and to have access to local estimates from their neighbors; there is no need for the nodes to know any of the constraints besides their own.

One important issue that is useful to mention is that some solution methods (e.g., [21]) require a *feasible* initial condition for their distributed algorithm. When the constraint set is distributed across the agents, it is not possible to find such feasible initial conditions without a substantial amount of in-network communication. We therefore take a different approach. By relying on suitably defined stochastic approximation steps, we show how the weight estimates constructed by the various nodes will approach the optimal feasible solution with arbitrarily good precision.

The technique used in this work relies on the use of diffusion strategies, which have been proven to have useful convergence and learning properties [23]–[25], [29]–[31]. The algorithm is comprised of three steps: 1) an adaptation step that updates the current solution using the local stochastic gradient available at the current iteration; 2) a constraint penalty step that penalizes directions that are not feasible according to the local constraint set; 3) and an aggregation step in which each agent combines its solution estimate with that of its *network neighbors*. In this way, the only communication that takes place in the algorithm is in-network and relatively low-power since neighbors are usually (but not necessarily) chosen according to physical proximity.

Notation: Throughout the manuscript, random quantities are denoted in boldface. Matrices are denoted in capital letters while vectors and scalars are denoted in small-case letters. The operator $\preceq$ denotes an element-wise inequality; i.e., $a \preceq b$ implies that each pair of elements of the vectors a and b satisfy $a_i \leq b_i$.

II. Background: Augmentation Methods

In this section, we briefly review a basic technique in constrained deterministic optimization and highlight some of the issues that are relevant in the context of distributed implementations and which need attention. Specifically, we describe augmentation-based methods for constrained optimization. These methods generally fall into two categories: (1) barrier methods, also known as interior penalty methods, and (2) penalty methods, also known as exterior penalty methods. Both methods are based on a simple yet insightful technique to augment the original objective function with a "penalty" term that penalizes getting too close to the constraint from the interior of the feasible set or leaving the feasible region altogether.

Thus, consider a convex optimization problem of the form:

$$\begin{aligned} \min_{w} \quad & J(w) \\ \text{subject to} \quad & g_l(w) \leq 0, \qquad l = 1, 2, \ldots, L \end{aligned} \tag{1}$$

where $w \in \mathbb{R}^M$, $\{g_1(w), \ldots, g_L(w)\}$ is a collection of convex functions, and $J(w)$ is a strongly convex function from $\mathbb{R}^M$ to $\mathbb{R}$. Augmentation incorporates the inequality constraints into the cost function and helps transform the constrained optimization problem into an unconstrained optimization problem via a *convex* nondecreasing barrier or penalty function $\delta(\cdot) : \mathbb{R} \to \mathbb{R}$, in the following manner:

$$\min_{w} \; J(w) + \eta \sum_{l=1}^{L} \delta\left(g_l(w)\right) \tag{2}$$

where $\eta > 0$ is a scalar parameter that controls the relative importance of adhering to the constraints. One choice for $\delta(\cdot)$ that yields an equivalent problem to (1) for any finite $\eta > 0$ is the indicator function [27, pp. 562–563]:

$$\delta^{\mathrm{IF}}(x) = \begin{cases} 0, & x \leq 0 \\ \infty, & \text{otherwise} \end{cases} \tag{3}$$

Observe that the indicator function $\delta^{\mathrm{IF}}(x)$ is convex and nondecreasing. Since the indicator function is generally nondifferentiable, approximations are used in its place. The main difference between barrier methods and penalty methods is the choice of the approximating functions.

A. Barrier Method

Barrier methods set a "barrier" around the feasible region. One of the most popular smooth approximations for (3) is the logarithmic barrier function:

$$\delta^{\log}(x) = \begin{cases} -\log(-x), & x < 0 \\ \infty, & \text{otherwise} \end{cases} \tag{4}$$

In this case, the algorithm requires a strictly feasible initialization, so that the augmented cost given in (2) is finite. A gradient-descent optimization algorithm would then travel against the gradient of (2), while adjusting the step-size to ensure that the next iterate stays within the feasible region via a line-search algorithm [27, p. 464] [28, p. 288]. Barrier methods are *interior-point methods* since the iterates never leave the feasible-set. Clearly, this is an advantage since any solution obtained during the optimization process may be used as a sub-optimal approximation. Nevertheless, this advantage requires a strictly feasible initialization. When the entire constraint set $\{g_1(w), \ldots, g_L(w)\}$ is not available to an agent (as happens in *distributed* constrained optimization), then it is not possible to choose a strictly feasible initializer without sharing

TABLE I
TABLE COMPARING THE BARRIER AND PENALTY METHODS FOR DISTRIBUTED CONSTRAINED OPTIMIZATION

Method	Feasible Start	Incorporate Equality Constraints	Full Knowledge of Feasible Set	Iterates Feasible	Constant Step-size
Barrier	Required	Indirectly	Required	Guaranteed	No (Backtracking)
Penalty	Not Required	Directly	Not Required	Asymptotically	Yes

this global information with the agents. This situation creates an annoying disadvantage from the perspective of distributed optimization. Penalty methods avoid this difficulty.

B. Penalty Method

In contrast to barrier methods, penalty methods give some positive penalty to solutions that fall outside the feasible set. In this case, the inequality penalty function takes the form:

$$\delta^{\mathrm{IP}}(x) = \begin{cases} 0, & x \le 0 \\ > 0, & \text{otherwise} \end{cases} \tag{5}$$

For example, one continuous, convex, nondecreasing, and twice-differentiable choice that satisfies (5) is:

$$\delta^{\mathrm{SIP}}(x) = \max\left(0, \frac{x^3}{\sqrt{x^2+\rho^2}}\right) \tag{6}$$

for some parameter $\rho > 0$. Observe that $\delta^{\mathrm{SIP}}(x)$ does not assume unbounded values for bounded x and, therefore, penalty methods do not require a feasible solution as an initializer. While this fact implies that penalty methods are particularly well-suited for distributed optimization scenarios, it also follows that the iterates may not remain inside the feasible region in general. This property means that there is no longer a need to execute a linesearch backtracking algorithm, as we would have done had we used the barrier function method since the objective value may become infinite in that case. Therefore, a constant step-size may now be used throughout the execution of the algorithm. The use of constant step-sizes is advantageous for a couple of reasons. First, it allows us to reduce the number of free parameters in the algorithm. Second, it becomes possible to derive useful bounds on the performance of the algorithm. And, perhaps more importantly, constant step-sizes endow the resulting distributed algorithm with adaptation and learning abilities. In this way, the algorithm acquires the ability to track in real-time slow variations in the underlying constraints and in the location of the slowly changing minimizer. In comparison, diminishing step-sizes are problematic because once these step-sizes approach their zero limiting value, the algorithm stops adapting.

For penalty methods, we observe that the approximation (5) of (3) improves in quality as η increases in value [28, p. 288], [32, p. 366]. This is because the penalty on the inside of the feasible region is zero and does not increase as η is increased. While, as $\eta \to \infty$, the function $\eta \cdot \delta^{\mathrm{IP}}(x)$ approximates the ideal barrier (3). Since $J(w)$ and the penalty function, $\delta^{\mathrm{IP}}(g_l(w))$, are convex, the augmented cost is also convex and its minimizer is obtained at the optimizer of the original optimization problem as $\eta \to \infty$ [32, p. 366].

Another advantage of penalty methods, as opposed to barrier methods, is that it is possible to easily incorporate *affine* constraints as well. Thus, consider the constrained convex optimization problem:

$$\begin{aligned} \min_w \quad & J(w) \\ \text{subject to} \quad & h_u(w) = 0, \quad u = 1, 2, \ldots, U \\ & g_l(w) \le 0, \quad l = 1, 2, \ldots, L \end{aligned} \tag{7}$$

where the functions $h_u(w)$ are affine and $g_l(w)$ are convex. This problem can be approached as an unconstrained optimization problem by using penalty functions as follows:

$$\min_w \; J(w) + \eta \left[\sum_{l=1}^{L} \delta^{\mathrm{IP}}\left(g_l(w)\right) + \sum_{u=1}^{U} \delta^{\mathrm{EP}}\left(h_u(w)\right) \right] \tag{8}$$

where $\delta^{\mathrm{IP}}(\cdot)$ is described in (5) while $\delta^{\mathrm{EP}}(\cdot) : \mathbb{R} \to \mathbb{R}$ is a convex function that is described by

$$\delta^{\mathrm{EP}}(x) = \begin{cases} 0, & x = 0 \\ > 0, & x \ne 0 \end{cases} \tag{9}$$

One choice of a continuous, convex, and twice-differentiable equality penalty function that satisfies (9) is the quadratic penalty:

$$\delta^{\mathrm{SEP}}(x) = x^2 \tag{10}$$

Clearly, the penalty functions are convex. Indeed, note that $\delta^{\mathrm{SIP}}(g(w))$ is convex since $\delta^{\mathrm{SIP}}(x)$ is convex and nondecreasing and $g(w)$ is convex, while $\delta^{\mathrm{SEP}}(h(w))$ is convex since $\delta^{\mathrm{SEP}}(x)$ is convex and $h(w)$ is affine [27, pp. 79,83–84]. Combining the convexity of the penalty functions with the fact that the original objective function is strongly convex, we conclude that the augmented cost (8) is also strongly convex. Moreover, when (7) is feasible, the minimizer of (8) will tend to the optimal solution of (7) as $\eta \to \infty$ (see Theorem 1). This result shows that it is possible to tackle both equality and inequality constraints simultaneously using penalty methods. Table I compares the barrier and penalty methods for the solution of distributed optimization problems.

In the next section, we employ penalty methods to develop an adaptive distributed algorithm for the solution of constrained convex optimization problems.

III. CONSTRAINED OPTIMIZATION OVER NETWORKS

Consider a network of agents (nodes), where each node k possesses a strongly convex cost function, $J_k(w)$, and a convex set of constraints $w \in \mathbb{W}_k$ where $w \in \mathbb{R}^M$. The objective of the network is to optimize the aggregate cost across all nodes subject to all constraints, i.e.,

$$\begin{aligned} \min_w \quad & J^{\mathrm{glob}}(w) \triangleq \sum_{k=1}^{N} J_k(w) \\ \text{subject to} \quad & w \in \mathbb{W}_1, \ldots, w \in \mathbb{W}_N \end{aligned} \tag{11}$$

Each of the convex sets $\{\mathbb{W}_1, \ldots, \mathbb{W}_N\}$ is defined as the set of points w that satisfy a collection of affine equality and convex inequality constraints:

$$\mathbb{W}_k \triangleq \left\{ w : \begin{array}{ll} h_{k,u}(w) = 0, & u = 1, \ldots, U_k \\ g_{k,l}(w) \leq 0, & l = 1, \ldots, L_k \end{array} \right\} \tag{12}$$

Obviously, the original optimization problem (11) can be cast as the optimization of the aggregate cost function $J^{\text{glob}}(w)$ over the common feasible set, $\mathbb{W}_1 \cap \ldots \cap \mathbb{W}_N$:

$$\min_w \; J^{\text{glob}}(w) \quad \text{subject to } w \in \mathbb{W} \tag{13}$$

where $\mathbb{W} \triangleq \mathbb{W}_1 \cap \ldots \cap \mathbb{W}_N$ is a convex set since the intersection of convex sets is itself convex [27, p. 36]. Assuming a solution for the above deterministic optimization problem exists (i.e., $\mathbb{W} \neq \emptyset$), we will denote an optimal solution for it by $w^\star$. The optimal objective value is given by $J^{\text{glob}}(w^\star)$. Observe that since $J^{\text{glob}}(w)$ is strongly convex, then $w^\star$ is unique (see Fact 1 further ahead).

Remark 1: Although we are requiring the individual cost functions $J_k(w)$ to be strongly convex, this condition is actually unnecessary and it is sufficient to require that at least one of the individual costs is strongly convex while all other costs can simply be convex; this condition is sufficient to ensure that the aggregate cost $J^{\text{glob}}(w)$ will remain strongly convex. Most of the results in this manuscript will hold under such weaker conditions. The strong convexity of the individual costs is adopted here for three reasons. First, the more relaxed situation would require more technical arguments to pursue a generalization of Theorem 2, as shown in [33], [34] in a different context. Due to space limitations, we opt to illustrate our construction under the strong convexity condition to facilitate the exposition of the main conclusions without digressing into specialized situations. Second, strong convexity is satisfied in many applications involving adaptation and learning where it is common to incorporate regularization into the cost functions. Third, when strong convexity is not satisfied, the Hessian matrices of the individual costs can become ill-conditioned, which is known to be problematic for optimization algorithms. □

Returning to (11), using the cost-augmentation technique described in Section II, we approximate (11) by using penalty functions in a manner similar to (8). Specifically, we consider the unconstrained problem:

$$\min_w \; J_\eta^{\text{glob}}(w) \tag{14}$$

where

$$J_\eta^{\text{glob}}(w) \triangleq \sum_{k=1}^{N} J'_{k,\eta}(w) \tag{15}$$

and

$$J'_{k,\eta}(w) \triangleq J_k(w) + \eta \cdot p_k(w) \tag{16}$$

with

$$p_k(w) \triangleq \sum_{l=1}^{L_k} \delta^{\text{IP}}\left(g_{k,l}(w)\right) + \sum_{u=1}^{U_k} \delta^{\text{EP}}\left(h_{k,u}(w)\right) \tag{17}$$

The terms $\delta^{\text{IP}}(x)$ and $\delta^{\text{EP}}(x)$ denote continuous convex functions that satisfy (5) and (9), respectively. We assume that $\delta^{\text{IP}}(x)$ and $\delta^{\text{EP}}(x)$ are selected so that $\nabla_w p_k(w') = 0$ when $w' \in \mathbb{W}$ (this is the case, for example, for (6) and (10)). We stress that (14) is *not* an equivalent problem to (11), but is an approximation for it. We will see later though that the approximation improves as $\eta \to \infty$. When $J^{\text{glob}}(w)$ is strongly convex, the cost (14) will also be strongly convex and will have a unique optimizer for any $\eta > 0$ (see Fact 2 further ahead). We shall denote this optimal solution to (14) by $w^o(\eta)$, which is parameterized in terms of η. Our task is now two-fold: (1) to motivate a fully distributed algorithm to solve (14) and determine $w^o(\eta)$, and (2) to characterize the distance between $w^o(\eta)$ and the desired optimizer $w^\star$ of (11). The distributed solution that we develop will rely solely on local in-network processing with each agent having knowledge of only its own constraint set $\mathbb{W}_k$. We will establish after Theorem 2 in the sequel that by choosing the algorithm's parameters appropriately, it is possible to obtain an arbitrarily accurate approximation for $w^\star$.

A. Diffusion-Based Distributed Optimization

Consider the optimization problem given by (14). Observe that each function $J'_{k,\eta}(w)$ depends only on agent k's information: cost function $J_k(w)$ and constraint set $\mathbb{W}_k$. This situation falls within the framework of unconstrained diffusion optimization [24], [30], [35]. Following similar arguments to those employed in these references, we conclude that one way to seek the minimizer of (15) is for each node to run iterations of the following form with a constant step-size:

$$\psi_{k,i} = w_{k,i-1} - \mu \cdot \nabla_w J'_{k,\eta}(w_{k,i-1}) \tag{18a}$$

$$w_{k,i} = \sum_{\ell=1}^{N} a_{\ell k} \psi_{\ell,i} \tag{18b}$$

In (18a), (18b), the vector $w_{k,i-1}$ denotes the estimate for $w^o(\eta)$ at node k at iteration $i-1$. This iterate is first updated via the (adaptive) gradient-descent update (18a) with step-size $\mu > 0$ to the intermediate value $\psi_{k,i}$. All other nodes in the network perform a similar update simultaneously by using their gradient vectors. Subsequently, each node k uses (18b) to combine, in a convex manner, the intermediate estimates from its neighbors. This step results in the updated estimate $w_{k,i}$ and the process repeats itself. The nonnegative coefficients $\{a_{\ell k}\}$ are chosen to satisfy the conditions:

$$a_{\ell k} = 0, \quad \text{when agents } \ell \text{ and } k \text{ are not neighbors} \tag{19a}$$

$$\sum_{\ell=1}^{N} a_{\ell k} = 1, \quad k = 1, \ldots, N \tag{19b}$$

If we collect the combination coefficients into a matrix $A = [a_{\ell k}]$, then condition (19b) implies that A is left-stochastic (i.e., it satisfies $A^\top \mathbb{1}_N = \mathbb{1}_N$, where $\mathbb{1}_N \in \mathbb{R}^N$ is the vector with all entries equal to one).

Evaluating the gradient vector from (16) and substituting into (18a) we get:

$$\psi_{k,i} = w_{k,i-1} - \mu \cdot \nabla_w J_k(w_{k,i-1}) - \mu\eta \cdot \nabla_w p_k(w_{k,i-1}) \tag{20}$$

for differentiable penalty functions. Expression (20) indicates that the update from $w_{k,i-1}$ to $\psi_{k,i}$ involves two components: the original gradient vector, $\nabla_w J_k(\cdot)$, and the gradient vector of the penalty function. We can incorporate these update terms into $w_{k,i-1}$ in different orders. For example, we may split the update into two parts: first we move from $w_{k,i-1}$ to $\psi_{k,i}$ in the opposite direction of the gradient vector of $J_k(\cdot)$. Subsequently, we incorporate the correction by the penalty gradients, say, as follows:

$$\zeta_{k,i} = w_{k,i-1} - \mu \cdot \nabla_w J_k(w_{k,i-1}) \tag{21a}$$
$$\psi_{k,i} = \zeta_{k,i} - \mu\eta \cdot \nabla_w p_k(w_{k,i-1}) \tag{21b}$$

One could also incorporate the gradient of $p_k(\cdot)$ first into $w_{k,i-1}$ followed by the gradient of $J_k(\cdot)$ to arrive at $\psi_{k,i}$. We continue with the order (21a), (21b) since it is sufficient to convey the main features of our construction. Now, it is generally expected that the intermediate iterate $\zeta_{k,i}$ generated by (21a) is a better estimate for $w^o(\eta)$ than $w_{k,i-1}$. This motivates us to replace $w_{k,i-1}$ in (21b) by $\zeta_{k,i}$ to get:

$$\zeta_{k,i} = w_{k,i-1} - \mu \cdot \nabla_w J_k(w_{k,i-1}) \tag{22}$$
$$\psi_{k,i} = \zeta_{k,i} - \mu\eta \cdot \nabla_w p_k(\zeta_{k,i}) \tag{23}$$

This last substitution is reminiscent of incremental-type arguments in gradient descent algorithms [36]–[38]. We further observe from (17) that the gradient vector of the penalty function can in turn be decomposed into the sum of two gradient components: one arising from the inequality constraints and the other from the equality constraints. Thus, in principle, we can further split (23) into two steps by adding these two gradient components one at a time. We shall forgo this extension here since (22), (23) is sufficient to convey the idea behind the main construction in this article. Further splitting of the gradient updates can generally help improve the performance of the distributed algorithm; this study can be pursued using techniques similar to those used by [39].

Now, combining (22), (23) with (18b), we arrive at what we shall refer to as the *penalized* Adapt-then-Combine (ATC) diffusion algorithm shown in (24a)–(24c),

---

Algorithm 1: Adapt-then-Combine (ATC) Diffusion Strategy

$$\zeta_{k,i} = w_{k,i-1} - \mu \cdot \nabla_w J_k(w_{k,i-1}) \tag{24a}$$
$$\psi_{k,i} = \zeta_{k,i} - \mu\eta \cdot \nabla_w p_k(\zeta_{k,i}) \tag{24b}$$
$$w_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\psi_{\ell,i} \tag{24c}$$

---

where $\mathcal{N}_k$ denotes the neighborhood of node k. It is also possible to exchange the order in which steps (18a), (18b) are performed, with combination performed prior to adaptation. Following similar arguments to the above, we can motivate the alternative *penalized* Combine-then-Adapt (CTA) diffusion algorithm shown in (25a)–(25c). Observe that in both penalized ATC and CTA algorithms, there is an explicit step to move along the gradient of the penalty function. This step can be thought of as performing a single incremental "projection" step along agent k's constraints [28, pp. 20–21]. Before we move on to examine the convergence of these distributed strategies for sufficiently small step-sizes, we pause to compare their structure with other related contributions in the literature.

Algorithm 2: Combine-then-Adapt (CTA) Diffusion Strategy

$$\psi_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k} w_{\ell,i-1} \tag{25a}$$
$$\zeta_{k,i} = \psi_{k,i-1} - \mu \cdot \nabla_w J_k(\psi_{k,i-1}) \tag{25b}$$
$$w_{k,i} = \zeta_{k,i} - \mu\eta \cdot \nabla_w p_k(\zeta_{k,i}) \tag{25c}$$

B. Comparison With Other Methods

We first compare the penalized CTA algorithm (25a)–(25c) to the consensus-type algorithm used in [20] for constrained optimization, and which is reproduced below using our notation (the same structure is used in [5], but for unconstrained optimization):

$$\psi_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k} w_{\ell,i-1} \tag{26a}$$
$$\zeta_{k,i} = \psi_{k,i-1} - \mu(i) \cdot \nabla_w J_k(w_{k,i-1}) \tag{26b}$$
$$w_{k,i} = P_{\mathbb{W}_1\cap\ldots\cap\mathbb{W}_N}[\zeta_{k,i}] \tag{26c}$$

where the notation $P_{\mathbb{X}}[y]$ denotes the operation of projecting the vector y onto the set $\mathbb{X}$:

$$P_{\mathbb{X}}[y] \triangleq \arg\min_{x\in\mathbb{X}} \|x-y\| \tag{27}$$

Observe that (26c) involves an explicit projection step and that this step requires each agent k to know the constraints from across all agents in the network. Moreover, observe that the same weight estimate $\psi_{k,i-1}$ is used on the right-hand side of the diffusion update (25b), while different estimates $\{\psi_{k,i-1}, w_{k,i-1}\}$ are used on the right-hand side of the consensus update (26b). This asymmetry can cause an unbounded growth in the state of consensus networks and can lead to instability. It is explained in [24], [31] that performing the combination and adaptation steps incrementally, where the updated iterate $\psi_{k,i-1}$ is used in the gradient vector in (25b), guarantees network stability in mean-square-error optimization problems over diffusion networks, while consensus-based implementations using (26b) can become unstable.

Another distributed algorithm is developed in [4]; it relies on a structure similar to the penalized CTA diffusion form albeit with two important differences: step (25c) is replaced by the local projection step (28c) shown below and the constant step-size in step (25b) is replaced by an iteration-dependent step-size in step (28b):

$$\psi_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k} w_{\ell,i-1} \tag{28a}$$
$$\zeta_{k,i} = \psi_{k,i-1} - \mu(i) \cdot \nabla_w J_k(\psi_{k,i-1}) \tag{28b}$$
$$w_{k,i} = P_{\mathbb{W}_k}[\zeta_{k,i}] \tag{28c}$$

In this solution, each node does not need to know the global constraint set $\mathbb{W}$ and would project only onto agent k's constraint set $\mathbb{W}_k$, as indicated by (28c). However, and understandably, each constraint set $\mathbb{W}_k$ is required to be expressed as the intersection of "simple constraints" whose projections (28c) can be computed analytically, such as the projection onto the non-negative orthant. As explained earlier, the solution method we propose in this work removes the need for carrying out explicit projection steps such as (28c). Moreover, note that steps (26b) and (28b) utilize diminishing step-sizes, which limit the adaptation ability of the network in tracking slowly drifting constraints and cost functions under *dynamic* optimization scenarios. For this reason, we are setting the step-size to a constant value in (25b). By doing so, the dynamics of the algorithm changes in a significant manner. For one thing, when stochastic gradient approximations are used in place of the true gradients, the right-most gradient term in (28b) would not vanish anymore with a constant step-size and the algorithm will continue to adapt indefinitely. In this case, gradient noise will seep into the operation of the algorithm and it becomes necessary to examine whether the algorithm will still be able to approach the solution of the optimization problem with high accuracy. The main results in this paper establish that this is indeed the case.

IV. Analysis Setup and Main Assumptions

In this section, we study the performance of the penalized algorithms (24a)–(24c) and (25a)–(25c) in a unified manner. We shall not limit our analysis to deterministic optimization problems, but will consider more general stochastic gradient approximation problems where the true gradient vectors, $\nabla_w J_k(\cdot)$, are replaced by approximations, say, $\widehat{\nabla_w} J_k(\cdot)$. We model the approximate gradient direction as a randomly perturbed version of the true gradient, say, as:

$$\widehat{\nabla_w} J_k(w) \triangleq \nabla_w J_k(w) + \boldsymbol{v}_{k,i}(w) \tag{29}$$

where $\boldsymbol{v}_{k,i}(\cdot)$ is the perturbation vector (or gradient noise). Observe that once we replace $\nabla_w J_k(w)$ by $\widehat{\nabla_w} J_k(w)$, then the variables ϕ, ψ, ζ, and w in the diffusion strategies (24a)–(24c) and (25a)–(25c) become random variables due to the presence of the random perturbation $\boldsymbol{v}_{k,i}(\cdot)$.

In order to treat the two penalized diffusion algorithms (ATC and CTA) within a unified framework, we consider the following general description:

$$\boldsymbol{\phi}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\boldsymbol{w}_{\ell,i-1} \tag{30a}$$

$$\boldsymbol{\zeta}_{k,i} = \boldsymbol{\phi}_{k,i-1} - \mu \cdot \widehat{\nabla_w} J_k(\boldsymbol{\phi}_{k,i-1}) \tag{30b}$$

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\zeta}_{k,i} - \mu\eta \cdot \nabla_w p_k(\boldsymbol{\zeta}_{k,i}) \tag{30c}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\boldsymbol{\psi}_{\ell,i} \tag{30d}$$

where we introduced two sets of nonnegative convex combination coefficients $\{a_{1,\ell k}\}$ and $\{a_{2,\ell k}\}$ that form left-stochastic matrices A_1 and A_2 and satisfy:

$$a_{1,\ell k} = 0, \qquad \text{when } \ell \notin \mathcal{N}_k \tag{31}$$

$$a_{2,\ell k} = 0, \qquad \text{when } \ell \notin \mathcal{N}_k \tag{32}$$

In (30b), we already replaced the true gradient vector, $\nabla_w J_k(\cdot)$, with an approximation $\widehat{\nabla_w} J_k(\cdot)$, usually evaluated from instantaneous data realizations. For this reason, $\boldsymbol{\phi}$, $\boldsymbol{\psi}$, $\boldsymbol{\zeta}$, and $\boldsymbol{w}$ in (30a)–(30d) are denoted in boldface to highlight that they are now random variables. In order to recover the ATC algorithm, we set $A_1 = I_N$ and $A_2 = A$ and to recover the CTA algorithm we set $A_1 = A$ and $A_2 = I_N$.

Since the iterate $\boldsymbol{w}_{k,i}$ generated by (30d) is random, we shall measure performance by examining the average squared distance between $\boldsymbol{w}_{k,i}$ and $w^\star$ in steady-state, namely,

$$\limsup_{i\to\infty} \mathbb{E}\|w^\star - \boldsymbol{w}_{k,i}\|^2 \tag{33}$$

Now, using the optimal solution $w^o(\eta)$ of (14) we can write:

$$\begin{aligned}\limsup_{i\to\infty} \mathbb{E}\|w^\star - \boldsymbol{w}_{k,i}\|^2 &= \limsup_{i\to\infty} \mathbb{E}\left\|w^\star - w^o(\eta) + w^o(\eta) - \boldsymbol{w}_{k,i}\right\|^2 \\ &\le 2 \underbrace{\|w^\star - w^o(\eta)\|^2}_{\text{Approximation Error}} + 2\limsup_{i\to\infty} \mathbb{E}\left\|w^o(\eta) - \boldsymbol{w}_{k,i}\right\|^2\end{aligned} \tag{34}$$

We will see later that the approximation error $\|w^\star - w^o(\eta)\|^2$ can be driven to arbitrarily small values as $\eta \to \infty$. This agrees with the intuition from Section II-B. After we establish this fact, we shift our attention towards characterizing the second term of the upper bound in (34) in order to assess how small (33) is.

We now list the necessary assumptions for studying the performance of the diffusion strategies and explain how they arise and where they are used in the analysis. These conditions are of the same nature as assumptions regularly used in the broad stochastic optimization literature, as indicated by the references given below in the explanations.

A. Main Assumptions

Assumption 1 (Feasible Problem): Problem (11) is feasible and, therefore, a minimizer $w^\star \in \mathbb{W}$ exists. ■

This is a logical assumption and it simply states that the set $\mathbb{W}_1 \cap \ldots \cap \mathbb{W}_N$ is non-empty. This situation is common when analyzing barrier and penalty methods [27, p. 561] for solving convex optimization problems.

Assumption 2 (Individual Costs): Each cost function $J_k(w)$ has a Hessian matrix that is bounded from above, i.e., there exist $\{\lambda_{k,\max} > 0\}$ such that, for each $k = 1, \ldots, N$:

$$\nabla_w^2 J_k(w) \le \lambda_{k,\max} I_M \tag{35}$$

Furthermore, since the individual costs $J_k(w)$ are strongly convex, there exist $\lambda_{k,\min} > 0$ such that

$$\nabla_w^2 J_k(w) \ge \lambda_{k,\min} I_M \tag{36}$$

■

Observe that when (36) holds, we have the following facts.

Fact 1 (Uniqueness of $w^\star$): When Assumption 1 and (36) hold, the optimizer $w^\star$ of (11) is unique [40, p. 217]. ■

Fact 2 (Uniqueness of $w^o(\eta)$): When (36) holds, the optimizer $w^o(\eta)$ of (14) is unique for any $\eta \ge 0$. ■

Observe that Fact 2 does not require the existence of $w^\star$ (Assumption 1) in order for $w^o(\eta)$ to be unique—in this case, $w^o(\eta)$ will be infeasible in terms of $\mathbb{W}$ even as $\eta \to \infty$, and thus not meaningful. Fact 1 follows from Assumptions 1–2 since strict convexity (which is guaranteed by strong convexity) of the objective function, and the existence of an optimizer, guarantee uniqueness of the optimizer [40, p. 217]. The reason Fact 2 follows from Assumption 2 is that the aggregate cost in (14) will be strongly convex.

We also require the Hessian matrices of the penalty functions with respect to w to be bounded from above, but not necessarily from below (they are obviously positive-semidefinite since the penalty functions are convex).

Assumption 3 (Penalty Functions): The penalty function $p_k(w)$ is twice-differentiable and its Hessian matrix with respect to w is upper bounded, i.e.,

$$\nabla_w^2 p_k(w) \leq \lambda_{k,\max}^p I_M \tag{37}$$

where $\lambda_{k,\max}^p > 0$. Furthermore, since the penalty functions are convex, their Hessian matrices are positive-semidefinite. ■

Assumption 4 (Combination Matrices): The combination matrix A in the penalized ATC or CTA implementation is primitive (i.e., A has nonnegative elements and A^m has all positive elements for some $m > 0$). We also assume that A is doubly-stochastic. ■

Since in our unified framework (30a)–(30d), either A_1 or A_2 is the identity matrix, then Assumption 4 is equivalent to requiring that the product matrix $A = A_1 A_2$ is primitive and doubly-stochastic. A doubly-stochastic matrix A is one that satisfies $A^\top \mathbb{1} = \mathbb{1}$ and $A\mathbb{1} = \mathbb{1}$ so that the entries on each of its columns and on each of its rows add up to one. The widely used Metropolis weights [22], [41], [42] satisfy Assumption 4 and can be computed in a distributed manner:

$$a_{\ell k} = \begin{cases} \min\left(\frac{1}{|\mathcal{N}_\ell|}, \frac{1}{|\mathcal{N}_k|}\right), & \ell \in \mathcal{N}_k, \ell \neq k \\ 1 - \sum_{j \in \mathcal{N}_k \setminus \{k\}} a_{jk}, & \ell = k \\ 0, & \text{otherwise} \end{cases} \tag{38}$$

where the notation $|\mathcal{N}_k|$ denotes the degree of node k or the number of its neighbors. The primitive condition on A is satisfied by any connected network with at least one self-loop (i.e., at least one $a_{k,k} > 0$) [22]. This situation is common in practice where networks tend to be connected and at least one node has some level of trust in its own data.

Assumption 5 (Gradient Noise Model): We model the perturbed gradient vector as:

$$\widehat{\nabla_w} J_k(\boldsymbol{w}) = \nabla_w J_k(\boldsymbol{w}) + \boldsymbol{v}_{k,i}(\boldsymbol{w}) \tag{39}$$

where, conditioned on the past history of the iterates $\mathcal{H}_{i-1} \triangleq \{\boldsymbol{w}_{k,j} : k = 1, \ldots, N \text{ and } j \leq i-1\}$, the gradient noise $\boldsymbol{v}_{k,i}(\boldsymbol{w})$ is assumed to satisfy:

$$\mathbb{E}\left\{\boldsymbol{v}_{k,i}(\boldsymbol{w}) | \mathcal{H}_{i-1}\right\} = 0 \tag{40}$$

$$\mathbb{E}\left[\|\boldsymbol{v}_{k,i}(\boldsymbol{w})\|^2 | \mathcal{H}_{i-1}\right] \leq \alpha \|\boldsymbol{w}\|^2 + \sigma_v^2 \tag{41}$$

for some $\alpha \geq 0$, $\sigma_v^2 \geq 0$, and where $\boldsymbol{w} \in \mathcal{H}_{i-1}$. ■

Models similar to (40)–(42) are also used in the works by [5], [28] on distributed algorithms—see the explanation in [24], [35]. Taking expectations of both sides of (41), we obtain

$$\mathbb{E}\|\boldsymbol{v}_{k,i}(\boldsymbol{w})\|^2 \leq \alpha \mathbb{E}\|\boldsymbol{w}\|^2 + \sigma_v^2 \tag{42}$$

We are now ready to state our main results. We delay most of the proofs to the appendices to simplify the exposition.

V. Main Convergence Result

First, we characterize the distance between the optimizer of the augmented cost function (15), $w^o(\eta)$, and the optimizer of the original optimization problem (11), $w^\star$. This distance appears in the first term of (34). Therefore, in order to show that the right-hand-side of (34) can be made arbitrarily small, we must first show that this distance can be made arbitrarily small by choosing η appropriately. For convenience, we introduce the compact notation:

$$w^o(\infty) \triangleq \lim_{\eta \to \infty} w^o(\eta) \tag{43}$$

Theorem 1 (Approaching Optimal Solution): Under Assumptions 1, 2, it holds that:

$$\|w^\star - w^o(\infty)\| = 0 \tag{44}$$

so that $w^o(\infty)$ is feasible and optimal.

Proof: Since $J_\eta^{\text{glob}}(w)$ is strongly convex, we have that for any point $w \in \mathbb{R}^M$, the distance from the optimizer $w^o(\eta)$ is bounded by [27, p. 460]:

$$\|w^o(\eta) - w\| \leq \frac{2}{\lambda_{\min}} \left\|\nabla_w J_\eta^{\text{glob}}(w)\right\| \tag{45}$$

where $\lambda_{\min} = \min_k\{\lambda_{k,\min}\}$ as defined in Assumption 2. It is possible to obtain an upper bound in (45) that is independent of η as follows. Since we are free to pick w, we let $w = w^\star$, where $w^\star \in \mathbb{W}$ by Assumption 1 to obtain

$$\|w^o(\eta) - w^\star\| \leq \frac{2}{\lambda_{\min}} \left\|\nabla_w J_\eta^{\text{glob}}(w^\star)\right\| \tag{46}$$

Recalling (15), we have that

$$\nabla_w J_\eta^{\text{glob}}(w^\star) = \nabla_w J^{\text{glob}}(w^\star) + \eta \sum_{k=1}^N p_k(w^\star) \tag{47}$$

but since by construction, $p_k(w') = 0$ when $w' \in \mathbb{W}$, we have that

$$\nabla_w J_\eta^{\text{glob}}(w^\star) = \nabla_w J^{\text{glob}}(w^\star) \tag{48}$$

and since $\|w^o(\eta)\| \leq \|w^o(\eta) - w^\star\| + \|w^\star\|$, we obtain

$$\|w^o(\eta)\| \leq \frac{2}{\lambda_{\min}} \left\|\nabla_w J^{\text{glob}}(w^\star)\right\| + \|w^\star\| < \infty \tag{49}$$

The upper bound in (49) is independent of η and is also finite since $J^{\text{glob}}(w)$ is a continuous function in w. To obtain (44), we appeal to Theorem 9.2.2 from [32] by noting that $w^o(\eta) \in \mathbb{B}$, where $\mathbb{B} \subset \mathbb{R}^M$ is the compact set [43, p. 2–3, 188]

$$\mathbb{B} = \left\{ w : \|w\| \leq \frac{2}{\lambda_{\min}} \left\| \nabla_w J^{\text{glob}}(w^\star) \right\| + \|w^\star\| \right\} \tag{50}$$

from which we conclude (44). ■

We now turn our attention to the convergence of the distributed algorithm.

Theorem 2 (Convergence Condition): Let Assumptions 2, 3, 4, and 5 hold. Then, the diffusion strategy (30a)–(30d) converges for sufficiently small positive step-sizes, namely, for step-sizes that satisfy

$$\mu < \min_{1 \leq k \leq N} \left\{ \frac{2\lambda_{k,\max}}{\lambda_{k,\max}^2 + 2\alpha}, \frac{2\lambda_{k,\min}}{\lambda_{k,\min}^2 + 2\alpha}, \frac{2}{\eta \cdot \lambda_{k,\max}^p} \right\} \tag{51}$$

Specifically, it holds that for small μ:

$$\limsup_{i \to \infty} \ \mathbb{E} \left\| w^o(\eta) - \boldsymbol{w}_{k,i} \right\|^2 \leq c_1 \cdot \mu + c_2 \cdot (\eta \cdot \mu)^2 \tag{52}$$

for some positive constants $c_1, c_2 > 0$ so that

$$\lim_{\mu \to 0} \limsup_{i \to \infty} \ \mathbb{E} \left\| w^o(\eta) - \boldsymbol{w}_{k,i} \right\|^2 = 0. \tag{53}$$

Proof: See Appendix A. ■

Theorem 2 states that the expected squared distance between $\boldsymbol{w}_{k,i}$ at each node and $w^o(\eta)$ is on the order of μ or $(\eta \cdot \mu)^2$, whichever is larger. This implies that when the step-size is chosen to be sufficiently small, the expected error can be made arbitrarily small when we set

$$\eta \triangleq \frac{c}{\mu^\theta}, \quad 0 < \theta < 1, \quad c > 0 \tag{54}$$

so that

$$\boxed{\lim_{\mu \to 0} \limsup_{i \to \infty} \ \mathbb{E} \| w^\star - \boldsymbol{w}_{k,i} \|^2 = 0} \tag{55}$$

We conclude that the diffusion strategy (30a)–(30d) effectively solves (11) in a fully distributed manner with progressively improving estimates of the optimizer as $\mu \to 0$. In addition, the diffusion algorithm, which utilizes a constant step-size, is capable of tracking slowly varying constraint sets and will continue to track the true optimizer $w^\star$ as the convex constraint sets, $\mathbb{W}_k$, and cost functions, $J_k(w)$, slowly drift, as illustrated next.

VI. Design Example

We consider a distributed optimization problem with $N = 100$ nodes. Each node is associated with the mean-square-error cost $J_k(w) = \mathbb{E}(\boldsymbol{d}_k(i) - \boldsymbol{h}_{k,i}^\top w)^2$, where the desired signal $\boldsymbol{d}_k(i)$ is related to some unknown model $\overline{w}$ via the linear regression model:

$$\boldsymbol{d}_k(i) \triangleq \boldsymbol{h}_{k,i}^\top \overline{w} + \boldsymbol{n}_k(i) \tag{56}$$

To illustrate adaptation and tracking ability, we introduce a single moving quadratic inequality at each node

$$g_{k,i}(w) \triangleq (w - w_{c,i})^\top R_{c,k,i}^{-1} (w - w_{c,i}) - 1 \tag{57}$$

where $w_{c,i}$ and $R_{c,k,i} > 0$ are allowed to change with i. We then consider the global optimization problem:

$$\begin{aligned} \min_w \quad & \sum_{k=1}^N \mathbb{E} \left(\boldsymbol{d}_k(i) - \boldsymbol{h}_{k,i}^\top w \right)^2 \\ \text{subject to} \quad & (w - w_{c,i})^\top R_{c,k,i}^{-1} (w - w_{c,i}) \leq 1, \ \forall k \in [1, N] \end{aligned} \tag{58}$$

The centers, $\{w_{c,i}\}$, of the ellipsoids in the constraints in (58) can be node-dependent; we are setting them to be node-independent for illustration purposes and to ensure that the feasible set $\mathbb{W}_1 \cap \ldots \cap \mathbb{W}_N$ is non-empty. The projections associated with the distributed solution of this problem are not straightforward to solve analytically. We let the inequality constraints drift with time, by allowing $w_{c,i}$ and $R_{c,k,i}$ to drift randomly with time. Non-zero mean Gaussian random perturbations are added to $w_{c,i}$ at each time-step i while a Gaussian random diagonal matrix is added to $R_{c,k,i}$ at every time iteration while guaranteeing that $R_{c,k,i}$ remains positive-definite and well-conditioned. We then track the progress of the algorithm as the estimates at the nodes move towards the true optimizer $w_i^\star$ of (58) at time i. The statistical distributions associated with $\boldsymbol{h}_{k,i}$ and $\boldsymbol{n}_k(i)$ remain fixed for the duration of the simulation—and, therefore, $J_k(w)$ is fixed in this simulation while the constraints are drifting. While this need not be the case in general, and the diffusion algorithm will handle the non-stationary cost function scenario as well, keeping the cost function fixed facilitates the illustration of the results.

The variance of the noise $\boldsymbol{n}_k(i)$ is chosen randomly for each node so that $\sigma_{v,k}^2 \sim U(0,1)$, where $U(0,1)$ denotes a uniform distribution on the range (0,1). The covariance matrices $\mathbb{E}\boldsymbol{h}_{k,i}\boldsymbol{h}_{k,i}^\top = R_{h,k}$ are generated as $R_{h,k} = Q_k \Lambda_k Q_k^\top$ where Q_k is a randomly generated orthogonal matrix and Λ_k is a diagonal matrix with random elements so that $(\Lambda_k)_{l,l} \sim U(0,1)$. The model vector $\overline{w} \in \mathbb{R}^2$ is chosen randomly for the simulation. The constraint set is also initialized randomly, morphs and moves as time progresses throughout the simulation. A stepsize of $\mu = 1.25 \times 10^{-3}$ is chosen with $\eta = 2.5$. The combination weights used throughout the simulation are based on the Metropolis rule (38). We used the function $\delta^{\text{IP}}(x) = \delta^{\text{SIP}}(x)$ listed in (6) with $\rho = 0.015$. Fig. 1(a) illustrates the evolution of the estimates across the nodes as time progresses (denoted by the green circles). We observe that in Fig. 1(b) that the nodes are attracted towards the feasible region from their initial position and quickly converge towards the true optimizer $w_i^\star$, which is initially stationary. As the constraint set begins to change after $i = 100$, we notice that each node's estimate of the optimizer changes and tracks $w_i^\star$ even as the feasible region shrinks and continues to move throughout the simulation. The green dotted line corresponds to the trajectory of each node's estimate while the cyan solid line represents the average trajectory of the nodes' estimates throughout the simulation. We see that the individual node estimates (green circles) overlap after $i = 100$ since all the nodes will have similar opinions regarding the optimizer. Having illustrated the tracking ability of the algorithm, we also illustrate the validity of (55) for a fixed optimizer $w^\star$. To accomplish this, we use the same simulation setup as above except that we fix the constraints at one point and we set $\eta = 0.001 \cdot \mu^{-0.9}$

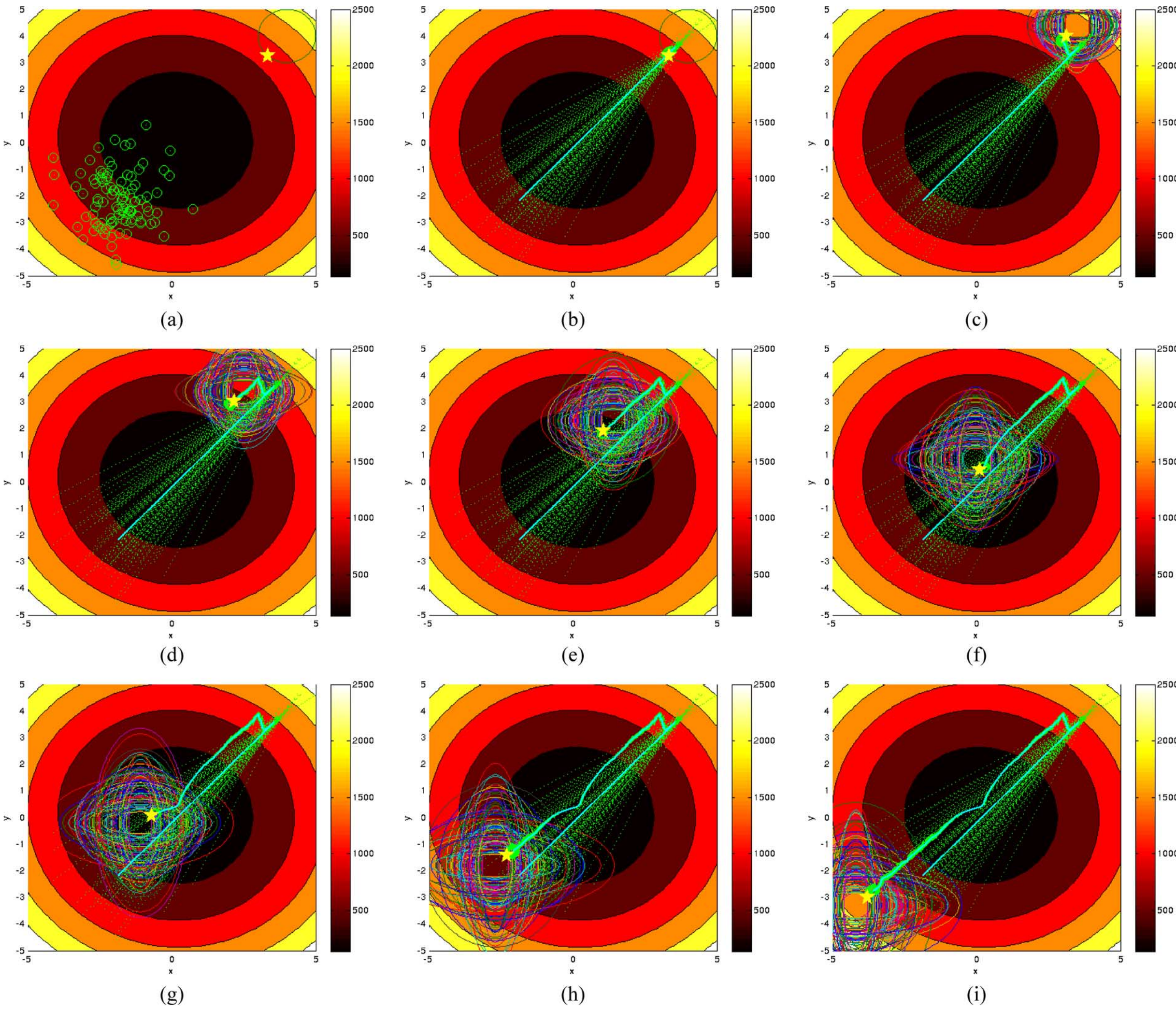


Fig. 1. The star indicates the location of the optimal minimizer, $w_i^\star$, which is allowed to drift in this simulation to illustrate the tracking ability of the algorithm. The ellipses in the graph denote the boundary of the feasible region for each node. The green circles illustrate the location of the estimates by the nodes, the dotted green lines indicate the trajectory of each node's estimate, the cyan line denotes the average network estimate trajectory; it is seen from the second plot from the left in the first row corresponding to $i = 100$ that this curve converges close to the minimizer location. As the constraint set begins to change starting at $i = 100$, we notice that the estimates are able to track the minimizer even as the feasible region shrinks and changes with time. (a) $i = 0$; (b) $i = 100$; (c) $i = 1100$; (d) $i = 3100$; (e) $i = 5300$; (f) $i = 8200$; (g) $i = 10300$; (h) $i = 13500$; (h) $i = 16500$.

as a function of the step-size μ (the factor 0.001 is present to guarantee convergence—see (51)). We then sweep the step-size μ from $\mu = 10^{-5}$ to $\mu = 10^{-3}$ and measure the average steady-state error $(1/N)\sum_{k=1}^{N}\mathbb{E}\|w^\star - \boldsymbol{w}_{k,i}\|^2$ and graph this relationship in Fig. 2. We observe that as the step-size is reduced, the expected error is also reduced, agreeing with (55). Note that this is the expected squared distance to the optimizer of the *original* problem (11), $w^\star$, and not just to the minimizer of the unconstrained problem (14), $w^o(\eta)$.

VII. Conclusion

In this work, we developed a distributed optimization strategy based on diffusion adaptation that allows a network of agents to solve a constrained convex problem in which the objective function is the aggregate sum of individual convex objective functions distributed across the nodes. The constraint set is the intersection of convex constraints at each node. The algorithm does not require the agents to know about other constraints besides their own and does not employ projection operators. We showed that through local interactions, the network is able to approach the desired global minimizer to arbitrarily good accuracy levels. The convergence analysis was performed in the stochastic setting in which the gradient vectors of the individual cost functions may not be available at each node and are approximated in the presence of gradient noise.

Appendix A

Proof of Theorem 2

In this section, we analyze how well the diffusion strategy (30a)–(30d) approaches the optimal solution $w^o(\eta)$ of the augmented cost (14). We examine this performance in terms of the mean squared error measure, $\mathbb{E}\|w^o(\eta) - \boldsymbol{w}_{k,i}\|^2$, in the presence of gradient noise, as modeled by Assumption 5. We extend the energy analysis framework from [30] to handle constrained optimization. Compared with the diffusion strategy studied in

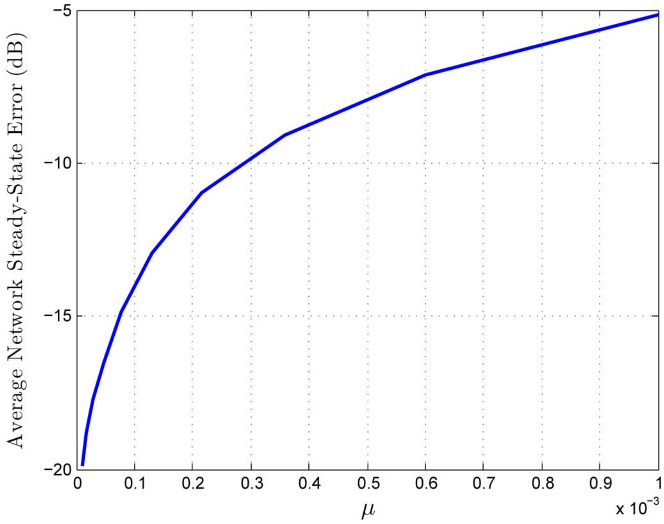


Fig. 2. Average steady-state error $(1/N)\sum_{k=1}^{N}\mathbb{E}\|w^\star - \boldsymbol{w}_{k,i}\|^2$ as a function of the step-size μ when $\eta = 0.001\mu^{-0.9}$.

[30], the models there did not incorporate penalty steps similar to (24b) and (25c). When these steps are incorporated, certain differences arise in the analysis that require attention (e.g., some symmetry properties present in the analysis of [30] are lost in the current context and need to be addressed). We first show that the diffusion strategy, in the absence of gradient noise, converges and has a fixed-point. Subsequently, we analyze the distance between this point and the vectors $w^o(\eta)$ and $\boldsymbol{w}_{k,i}$ in the mean-square-sense.

A. Existence of Fixed Point

At each iteration, we can view the diffusion strategy (30a)–(30d) as a mapping from the vectors $\{\boldsymbol{w}_{k,i-1}\}$ to the vectors $\{\boldsymbol{w}_{k,i}\}$ or, more generically, as a mapping from some block vector x to another block vector w. Thus, let $x = \text{col}\{x_1, x_2, \ldots, x_N\}$ denote a block vector with sub-vectors x_k of size $M \times 1$. Let also $P[x] \triangleq \text{col}\{\|x_1\|^2, \|x_2\|^2, \ldots, \|x_N\|^2\}$. Then, we observe that given any two input vectors $x^1, x^2 \in \mathbb{R}^{MN}$, the resulting updated vectors w^1 and w^2 are given by

$$w^1 = \left(A_2^\top \otimes I_M\right)\psi^1, \quad w^2 = \left(A_2^\top \otimes I_M\right)\psi^2 \tag{59a}$$

where the intermediate vectors ψ^1 and ψ^2 are constructed as follows in terms of other intermediate block vectors $\{\zeta^1, \zeta^2, \phi^1, \phi^2\}$:

$$\psi^1 = \begin{bmatrix} \zeta_1^1 - \mu\eta\nabla_w p_1\left(\zeta_1^1\right) \\ \vdots \\ \zeta_N^1 - \mu\eta\nabla_w p_N\left(\zeta_N^1\right)\end{bmatrix},$$

$$\psi^2 = \begin{bmatrix} \zeta_1^2 - \mu\eta\nabla_w p_1\left(\zeta_1^2\right) \\ \vdots \\ \zeta_N^2 - \mu\eta\nabla_w p_N\left(\zeta_N^2\right)\end{bmatrix}, \tag{59b}$$

$$\zeta^1 = \begin{bmatrix} \phi_1^1 - \mu\nabla_w J_1\left(\phi_1^1\right) \\ \vdots \\ \phi_N^1 - \mu\nabla_w J_N\left(\phi_N^1\right)\end{bmatrix},$$

$$\zeta^2 = \begin{bmatrix} \phi_1^2 - \mu\nabla_w J_1\left(\phi_1^2\right) \\ \vdots \\ \phi_N^2 - \mu\nabla_w J_N\left(\phi_N^2\right)\end{bmatrix}, \tag{59c}$$

$$\phi^1 = \left(A_1^\top \otimes I_M\right)x^1, \quad \phi^2 = \left(A_1^\top \otimes I_M\right)x^2. \tag{59d}$$

We now verify that the mapping $x \mapsto w$ is a contraction for sufficiently small step-sizes. Indeed, using the sub-multiplicative property of the block-maximum norm and the fact that A_1 and A_2 are left-stochastic [22], we conclude from (59a) and (59d):

$$\|w^1 - w^2\|_{b,\infty} \le \|\psi^1 - \psi^2\|_{b,\infty} \tag{60}$$

$$\|\phi^1 - \phi^2\|_{b,\infty} \le \|x^1 - x^2\|_{b,\infty} \tag{61}$$

Now, we can bound the quantity $\|\psi^1 - \psi^2\|_{b,\infty}$ by appealing to the mean-value theorem [28, p. 24] to write:

$$\nabla_w p_k\left(\zeta_k^1\right) - \nabla_w p_k\left(\zeta_k^2\right) = \left(\int_0^1 \nabla_w^2 p_k\left(\zeta_k^2 + t\left(\zeta_k^1 - \zeta_k^2\right)\right)dt\right)\left(\zeta_k^1 - \zeta_k^2\right) \tag{62}$$

from which we conclude that

$$\|\psi^1 - \psi^2\|_{b,\infty} \le \max_{1\le k\le N}\left\| I_M - \mu\eta\int_0^1 \nabla_w^2 p_k\left(\zeta_k^2 + t\left(\zeta_k^1 - \zeta_k^2\right)\right)dt\right\| \cdot \|\zeta_k^1 - \zeta_k^2\| \tag{63}$$

Now, due to Assumption 3, we have that

$$\left\| I_M - \mu\eta\int_0^1 \nabla_w^2 p_k\left(\zeta_k^2 + t\left(\zeta_k^1 - \zeta_k^2\right)\right)dt\right\| \le \max\left\{\left|1 - \mu\eta\lambda_{k,\max}^p\right|, 1\right\} \tag{64}$$

The bound on the right-hand side of (64) can be guaranteed to be at most one when

$$0 \le \mu\eta \le \min_{1\le k\le N}\left\{\frac{2}{\lambda_{k,\max}^p}\right\} \tag{65}$$

so that

$$\|\psi^1 - \psi^2\|_{b,\infty} \le \|\zeta^1 - \zeta^2\|_{b,\infty} \tag{66}$$

In a similar manner to (62), (63), we can verify that

$$\|\zeta^1 - \zeta^2\|_{b,\infty} \le \max_{1\le k\le N}\left\| I_M - \mu\int_0^1 \nabla_w^2 J_k\left(\phi_k^2 + t\left(\phi_k^1 - \phi_k^2\right)\right)dt\right\| \cdot \|\phi_k^1 - \phi_k^2\| \tag{67}$$

and due to Assumption 2,

$$\lambda_{k,\min} I_M \le \int_0^1 \nabla_w^2 J_k\left(\phi_k^2 + t\left(\phi_k^1 - \phi_k^2\right)\right)dt \le \lambda_{k,\max} I_M \tag{68}$$

It follows that

$$\|\zeta^1 - \zeta^2\|_{b,\infty} \leq \gamma \cdot \|\phi^1 - \phi^2\|_{b,\infty} \tag{69}$$

$$\gamma \triangleq \max_{1\leq k\leq N} \{\gamma_k\} \tag{70}$$

$$\gamma_k \triangleq \max\{|1-\mu\lambda_{k,\min}|, |1-\mu\lambda_{k,\max}|\} \tag{71}$$

and γ_k satisfies $0 \leq \gamma_k < 1$ when

$$0 < \mu < \min_{1\leq k\leq N}\left\{\frac{2}{\lambda_{k,\max}}\right\} \tag{72}$$

Combining the previous results together we arrive at

$$\|w^1 - w^2\|_{b,\infty} \leq \gamma\|x^1 - x^2\|_{b,\infty} \tag{73}$$

for $\gamma < 1$ when (65) and (72) are satisfied.

Remark 2: It is the above argument that relies on the requirement that all individual costs are strongly convex so that all the $\lambda_{k,\min}$ are strictly positive and each γ_k can be made strictly less than one. If we relax the strong convexity assumption and require only at least one of the individual costs to be strongly convex, then the above argument needs to be adjusted as done in [33], [34]. ■

We conclude that the diffusion mapping $x \mapsto w$ is a contraction mapping for sufficiently small step-sizes. By the Banach fixed point theorem [44, pp. 299–303], this mapping will have a unique fixed point, w_∞. Observe that this fixed point may not be $\mathbb{1}_N \otimes w^o(\eta)$. However, since we are interested in studying the rightmost term in (34) in the gradient noise case, or equivalently,

$$\limsup_{i\to\infty} \mathbb{E}\,\|\mathbb{1}_N \otimes w^o(\eta) - \boldsymbol{w}_i\|^2 \tag{74}$$

where $\boldsymbol{w}_i \triangleq \mathrm{col}\{\boldsymbol{w}_{1,i}, \ldots, \boldsymbol{w}_{N,i}\}$, we will decompose the above squared distance into two parts: (1) the expected squared distance from w_∞ to $\boldsymbol{w}_i$, and (2) the squared distance from $w^o(\eta)$ to w_∞ (the bias of the algorithm):

$$\begin{aligned}\mathbb{E}\,\|\mathbb{1}\otimes w^o(\eta) - \boldsymbol{w}_i\|^2 &= \mathbb{E}\,\|\mathbb{1}\otimes w^o(\eta) - w_\infty + w_\infty - \boldsymbol{w}_i\|^2\\ &\leq 2\mathbb{1}_N^\mathsf{T}\mathbb{E}P[\boldsymbol{w}_i - w_\infty] + 2\,\|\mathbb{1}\otimes w^o(\eta) - w_\infty\|^2 \end{aligned}\tag{75}$$

In order to proceed with the study, we first examine the quantity $\mathbb{E}P[\boldsymbol{w}_i - w_\infty]$ and assess the size of the right-most term.

B. Mean-Square-Distance to Fixed Point

We introduce the vectors ϕ_∞, ψ_∞, ζ_∞, and the fixed-point w_∞ and their respective block entries $\phi_{k,\infty}$, $\psi_{k,\infty}$, $\zeta_{k,\infty}$, and $w_{k,\infty}$ obtained by letting $x^1 = w_\infty$ in (59a)–(59d). The mean-square-error between the iterates $\boldsymbol{\phi}_{k,i-1}$ and $\boldsymbol{w}_{k,i}$ in the stochastic approximation setting, their respective limit points in the noiseless recursion, are bounded using Jensen's inequality [27, p. 77]

$$\mathbb{E}\|w_{k,\infty} - \boldsymbol{w}_{k,i}\|^2 \leq \sum_{\ell=1}^N a_{2,\ell k}\mathbb{E}\|\psi_{k,\infty} - \boldsymbol{\psi}_{\ell,i}\|^2 \tag{76}$$

$$\mathbb{E}\|\phi_{k,\infty} - \boldsymbol{\phi}_{k,i-1}\|^2 \leq \sum_{\ell=1}^N a_{1,\ell k}\mathbb{E}\|w_{k,\infty} - \boldsymbol{w}_{\ell,i-1}\|^2 \tag{77}$$

We also have that

$$\mathbb{E}\|\psi_{k,\infty} - \boldsymbol{\psi}_{k,i}\|^2 = \mathbb{E}\|\zeta_{k,\infty} - \boldsymbol{\zeta}_{k,i}\|^2_{\boldsymbol{\Omega}_{k,i}} \tag{78}$$

where

$$\boldsymbol{\Omega}_{k,i} \triangleq \left(I_M - \int_0^1 \nabla_w^2 p_k\left(\zeta_{k,\infty} - t(\zeta_{k,\infty} - \boldsymbol{\zeta}_{k,i})\right)dt\right)^2 \tag{79}$$

But due to (64), we have that

$$\mathbb{E}\|\psi_{k,\infty} - \boldsymbol{\psi}_{k,i}\|^2 \leq \mathbb{E}\|\zeta_{k,\infty} - \boldsymbol{\zeta}_{k,i}\|^2 \tag{80}$$

when (65) is satisfied. Moreover, the mean-square-error between $\zeta_{k,\infty}$ and $\boldsymbol{\zeta}_{k,i}$ can be bounded by

$$\begin{aligned}&\mathbb{E}\|\zeta_{k,\infty} - \boldsymbol{\zeta}_{k,i}\|^2\\ &\overset{(a)}{=} \mathbb{E}\|\phi_{k,\infty} - \boldsymbol{\phi}_{k,i-1}\|^2_{\boldsymbol{\Sigma}_{k,i-1}} + \mu^2\mathbb{E}\left\|\boldsymbol{v}_{k,i}(\boldsymbol{\phi}_{k,i-1})\right\|^2\\ &\overset{(b)}{\leq} \mathbb{E}\|\phi_{k,\infty} - \boldsymbol{\phi}_{k,i-1}\|^2_{\boldsymbol{\Sigma}_{k,i-1}} + \mu^2\left(\alpha\mathbb{E}\|\boldsymbol{\phi}_{k,i-1}\|^2 + \sigma_v^2\right)\\ &= \mathbb{E}\|\phi_{k,\infty} - \boldsymbol{\phi}_{k,i-1}\|^2_{\boldsymbol{\Sigma}_{k,i-1}}\\ &\quad+\mu^2\left(\alpha\mathbb{E}\left\|w^o(\eta) - \phi_{k,\infty} + \phi_{k,\infty} - \boldsymbol{\phi}_{k,i-1} - w^o(\eta)\right\|^2 + \sigma_v^2\right)\\ &\leq \mathbb{E}\|\phi_{k,\infty} - \boldsymbol{\phi}_{k,i-1}\|^2_{\boldsymbol{\Sigma}_{k,i-1}} + 2\mu^2\alpha\mathbb{E}\left\|w^o(\eta) - \phi_{k,\infty}\right\|^2\\ &\quad+ 2\mu^2\alpha\|\phi_{k,\infty} - \boldsymbol{\phi}_{k,i-1}\|^2 + \mu^2\left(2\alpha\left\|w^o(\eta)\right\|^2 + \sigma_v^2\right)\end{aligned} \tag{81}$$

where step (*a*) can be obtained via an argument similar to (62), step (*b*) is due to Assumption 5 and $\boldsymbol{\Sigma}_{k,i-1} \triangleq (I_M - \mu\boldsymbol{H}_{k,i-1})^2$, where $\boldsymbol{H}_{k,i-1}$ is defined as:

$$\boldsymbol{H}_{k,i-1} \triangleq \int_0^1 \nabla_w^2 J_k\left(\phi_{k,\infty} - t(\phi_{k,\infty} - \boldsymbol{\phi}_{k,i-1})\right)dt \tag{82}$$

Now, due to Assumption 2, we have that $0 \leq \boldsymbol{\Sigma}_{k,i-1} \leq \gamma_k^2 I_M$, where γ_k is defined in (71). Furthermore, from (30a) it is possible to bound $\|w^o(\eta) - \phi_{k,\infty}\|^2$ using Jensen's inequality:

$$\|w^o(\eta) - \phi_{k,\infty}\|^2 \leq \sum_{\ell=1}^N a_{1,\ell k}\left\|w^o(\eta) - w_{k,\infty}\right\|^2 \tag{83}$$

Substituting into (81), we get

$$\begin{aligned}&\mathbb{E}\|\zeta_{k,\infty} - \boldsymbol{\zeta}_{k,i}\|^2 \leq \left(\gamma_k^2 + 2\mu^2\alpha\right)\mathbb{E}\|\phi_{k,\infty} - \boldsymbol{\phi}_{k,i-1}\|^2\\ &+2\mu^2\alpha\sum_{\ell=1}^N a_{1,\ell k}\left\|w^o(\eta) - w_{k,\infty}\right\|^2 + \mu^2\left(2\alpha\left\|w^o(\eta)\right\|^2 + \sigma_v^2\right)\end{aligned} \tag{84}$$

Now, combining (76), (80), (84), and (77), we obtain the following recursion for $\mathbb{E}P[\boldsymbol{w}_i - w_\infty]$:

$$\mathbb{E}P[\boldsymbol{w}_i - w_\infty] \preceq A_2^\mathsf{T}\Gamma A_1^\mathsf{T}\mathbb{E}P[\boldsymbol{w}_{i-1} - w_\infty] + \mu^2 b \tag{85}$$

where $\Gamma \in \mathbb{R}^{N\times N}$ is a diagonal matrix with elements $\gamma_k^2 + 2\mu^2\alpha$ along the diagonal, and $b \in \mathbb{R}^N$ is defined as

$$b \triangleq 2\alpha A_2^\mathsf{T}A_1^\mathsf{T}P[\mathbb{1}_N\otimes w^o(\eta) - w_\infty] + \left(\sigma_v^2 + 2\alpha\left\|w^o(\eta)\right\|^2\right)\mathbb{1}_N \tag{86}$$

We prove in the next section that $b \leq c \cdot (\mu\eta)^2 + \text{constant}$ for constant $c > 0$—see (134). Iterating recursion (85) we obtain

$$\mathbb{E}P[\boldsymbol{w}_i - w_\infty] \preceq \left(A_2^\mathsf{T}\Gamma A_1^\mathsf{T}\right)^i \mathbb{E}P[\boldsymbol{w}_0 - w_\infty] + \mu^2 \sum_{j=0}^{i-1} \left(A_2^\mathsf{T}\Gamma A_1^\mathsf{T}\right)^j b \tag{87}$$

We remark that the matrix $A_2^\mathsf{T}\Gamma A_1^\mathsf{T}$ can be guaranteed to be stable for small step-sizes. To see this, we upper-bound the spectral radius by the matrix norm $\|B\|_\infty$, which is the maximum-absolute-row-sum:

$$\rho\left(A_2^\mathsf{T}\Gamma A_1^\mathsf{T}\right) \leq \left\|A_2^\mathsf{T}\Gamma A_1^\mathsf{T}\right\|_\infty \leq \|\Gamma\|_\infty = \max_{1\leq k\leq N}\left\{\gamma_k^2 + 2\mu^2\alpha\right\}$$

since A_1 and A_2 are left-stochastic matrices. We conclude then that the matrix $A_2^\mathsf{T}\Gamma A_1^\mathsf{T}$ is stable when

$$0 < \mu < \min_{1\leq k\leq N}\left\{\frac{2\lambda_{k,\min}}{\lambda_{k,\min}^2 + 2\alpha}, \frac{2\lambda_{k,\max}}{\lambda_{k,\max}^2 + 2\alpha}\right\} \tag{88}$$

In this case, we conclude from (87), the triangle inequality, and the submultiplicative property of induced norms, that

$$\begin{aligned}\left\|\limsup_{i\to\infty} \mathbb{E}P[\boldsymbol{w}_i - w_\infty]\right\|_\infty &\leq \mu^2\|b\|_\infty \left\|\sum_{j=0}^{\infty}\left(A_2^\mathsf{T}\Gamma A_1^\mathsf{T}\right)^j\right\|_\infty \\ &\leq \mu^2\|b\|_\infty \sum_{j=0}^{\infty}\|\Gamma\|_\infty^j \leq \mu^2\|b\|_\infty \sum_{j=0}^{\infty}(\gamma^2 + 2\mu^2\alpha)^j \\ &= \frac{\mu^2 \cdot \|b\|_\infty}{1 - \gamma^2 - 2\mu^2\alpha}\end{aligned} \tag{89}$$

where γ was defined in (70). Combining (70), (71), we have that γ^2 can be obtained as

$$\begin{aligned}\gamma^2 &= \max_{1\leq k\leq N}\left\{1 - 2\mu\lambda_{k,\min} + \mu^2\lambda_{k,\min}^2,\right. \\ &\qquad\qquad \left. 1 - 2\mu\lambda_{k,\max} + \mu^2\lambda_{k,\max}^2\right\} \\ &= 1 - \mu\min_{1\leq k\leq N}\left\{2\lambda_{k,\min} - \mu\lambda_{k,\min}^2, 2\lambda_{k,\max} - \mu\lambda_{k,\max}^2\right\}\end{aligned} \tag{90}$$

Substituting (90) into (89), we obtain

$$\left\|\limsup_{i\to\infty} \mathbb{E}P[\boldsymbol{w}_i - w_\infty]\right\|_\infty \leq \frac{\mu \cdot \|b\|_\infty}{\min_k\left\{2\lambda_{k,\min} - \mu\lambda_{k,\min}^2, 2\lambda_{k,\max} - \mu\lambda_{k,\max}^2\right\} - 2\mu\alpha} \tag{91}$$

Therefore, using the fact that $\|b\| \leq c \cdot (\mu\eta)^2 + \text{constant}$, as shown later in (134), we will be able to conclude that $\limsup_{i\to\infty} \mathbb{E}P[\boldsymbol{w}_i - w_\infty] = c_1 \cdot \mu$ for some constant $c_1 > 0$.

C. Bias Analysis at Small Step-Sizes

We now examine the dependence of $\|\mathbb{1}_N \otimes w^o(\eta) - w_\infty\|^2$ on μ; this term appears in expression (86) for b. First, we will derive an expression for $\tilde{w}_\infty \triangleq \mathbb{1}_N \otimes w^o(\eta) - w_\infty$. For the remainder of this appendix, we will write $w^o \triangleq w^o(\eta)$ in order to simplify the notation. Our arguments will still apply for any $\eta > 0$. Recall that w_∞ is the fixed point for the diffusion strategy in the absence of gradient noise. Therefore, let $i \to \infty$ in (30a)–(30d) in the absence of noise, and introduce the bias vectors $\tilde{w}_{k,\infty} = w^o - w_{k,\infty}$, $\tilde{\phi}_{k,\infty} = w^o - \phi_{k,\infty}$, $\tilde{\zeta}_{k,\infty} = w^o - \zeta_{k,\infty}$, and $\tilde{\psi}_{k,\infty} = w^o - \psi_{k,\infty}$. Subtracting $\phi_{k,\infty}, \zeta_{k,\infty}, \psi_{k,\infty}$, and $w_{k,\infty}$ from w^o yields,

$$\tilde{\phi}_{k,\infty} = \sum_{\ell=1}^{N} a_{1,\ell k}\tilde{w}_{\ell,\infty} \tag{92a}$$

$$\tilde{\zeta}_{k,\infty} = \tilde{\phi}_{k,\infty} + \mu\nabla_w J_k(\phi_{k,\infty}) \tag{92b}$$

$$\tilde{\psi}_{k,\infty} = \tilde{\zeta}_{k,\infty} + \mu\eta\nabla_w p_k(\zeta_{k,\infty}) \tag{92c}$$

$$\tilde{w}_{k,\infty} = \sum_{\ell=1}^{N} a_{2,\ell k}\tilde{\psi}_{\ell,\infty} \tag{92d}$$

Using the mean-value-theorem [28, p. 6], we can write

$$\nabla_w J_k(\phi_{k,\infty}) = \nabla_w J_k(w^o) - H_{k,\infty} \cdot \tilde{\phi}_{k,\infty} \tag{93}$$

where

$$H_{k,\infty} \triangleq \int_0^1 \nabla_w^2 J_k(w^o - t\tilde{\phi}_{k,\infty})dt \tag{94}$$

Therefore, (92b) becomes

$$\tilde{\zeta}_{k,\infty} = [I_M - \mu H_{k,\infty}] \cdot \tilde{\phi}_{k,\infty} + \mu\nabla_w J_k(w^o) \tag{95}$$

Similarly, we can obtain for (92c) that

$$\tilde{\psi}_{k,\infty} = [I_M - \mu\eta Z_{k,\infty}] \cdot \tilde{\zeta}_{k,\infty} + \mu\eta\nabla_w p_k(w^o) \tag{96}$$

where

$$Z_{k,\infty} \triangleq \int_0^1 \nabla_w^2 p_k\left(w^o - t\tilde{\zeta}_{k,\infty})\right)dt \tag{97}$$

To proceed, we introduce the extended quantities:

$$\mathcal{A}_1 \triangleq A_1 \otimes I_M \tag{98}$$

$$\mathcal{A}_2 \triangleq A_2 \otimes I_M \tag{99}$$

$$\mathcal{H}_\infty \triangleq \mathrm{diag}\{H_{1,\infty}, \ldots, H_{N,\infty}\} \tag{100}$$

$$\mathcal{Z}_\infty \triangleq \mathrm{diag}\{Z_{1,\infty}, \ldots, Z_{N,\infty}\} \tag{101}$$

$$g^o \triangleq \mathrm{col}\left\{\nabla_w J_1(w^o), \ldots, \nabla_w J_N(w^o)\right\} \tag{102}$$

$$f^o \triangleq \mathrm{col}\left\{\nabla_w p_1(w^o), \ldots, \nabla_w p_N(w^o)\right\} \tag{103}$$

as well as the network error vector $\tilde{w}_\infty = \mathrm{col}\{\tilde{w}_{1,\infty}, \ldots, \tilde{w}_{N,\infty}\}$. Using these block variables, recursions (92a)–(92d) lead to the following expression for $\tilde{w}_\infty$:

$$\begin{aligned}\tilde{w}_\infty = &\left[I_{MN} - \mathcal{A}_2^\mathsf{T}(I_{MN} - \mu\eta\mathcal{Z}_\infty)(I_{MN} - \mu\mathcal{H}_\infty)\mathcal{A}_1^\mathsf{T}\right]^{-1} \\ &\times\left[\mu\mathcal{A}_2^\mathsf{T}\left(I_{MN} - \mu\eta\mathcal{Z}_\infty\right)g^o + \mu\eta\mathcal{A}_2^\mathsf{T}f^o\right]\end{aligned} \tag{104}$$

when the inverse exists. The matrix is invertible when $\mathcal{A}_2^\mathsf{T}(I_{MN} - \mu\eta\mathcal{Z}_\infty)(I_{MN} - \mu\mathcal{H}_\infty)\mathcal{A}_1^\mathsf{T}$ is stable. Since the spectral radius of a matrix is upper-bounded by any of its induced norms, we have that

$$\rho\left(\mathcal{A}_2^\top(I_{MN}-\mu\eta\mathcal{Z}_\infty)(I_{MN}-\mu\mathcal{H}_\infty)\mathcal{A}_1^\top\right) \le \|I_{MN}-\mu\eta\mathcal{Z}_\infty\|_{b,\infty}\cdot\|I_{MN}-\mu\mathcal{H}_\infty\|_{b,\infty} \quad (105)$$

where $\|\cdot\|_{b,\infty}$ denotes the block-maximum norm [22], [35]. Now, it is sufficient to show that $\|I_{MN}-\mu\mathcal{H}_\infty\|_{b,\infty}<1$ and $\|I_{MN}-\mu\eta\mathcal{Z}_\infty\|_{b,\infty}\le 1$. For the former, observe that

$$\|I_{MN}-\mu\mathcal{H}_\infty\|_{b,\infty}=\max_{1\le k\le N}\{\|I_{MN}-\mu H_{k,\infty}\|_2\} \quad (106)$$

and due to Assumption 2,

$$(1-\mu\lambda_{k,\max})I_M\le I_M-\mu H_{k,\infty}\le(1-\mu\lambda_{k,\min})I_M \quad (107)$$

We conclude that $\|I_{MN}-\mu H_{k,\infty}\|_2\le\gamma_k$ where γ_k is defined in (71) and that $\|I_{MN}-\mu\mathcal{H}_\infty\|_{b,\infty}=\max_{1\le k\le N}\gamma_k$. Similarly, using Assumption 3, it can be verified that

$$\|I_{MN}-\mu\mathcal{Z}_\infty\|_{b,\infty}=\max_{1\le k\le N}\max\left\{1,|1-\mu\eta\lambda_{k,\max}^p|\right\} \quad (108)$$

Finally, observe that $\gamma_k<1$ is satisfied for all $1\le k\le N$ when μ is chosen according (72). Also, $\max\{1,|1-\mu\eta\lambda_{k,\max}^p|\}\le 1$ is satisfied for $1\le k\le N$ when $\mu\eta$ is chosen according to (65).

Comparing (104) with expression (85) in [30], it is clear now how the current setup is different and leads to additional challenges in the analysis. Observe that expression (104) contains the additional terms $\mathcal{Z}_\infty$ and f^o, which are due to the penalty functions. If these terms are set to zero, then (104) simplifies to expression (85) in [30]. Moreover, we rewrite (104) as:

$$\tilde{w}_\infty=\left[I_{MN}-\mathcal{A}_2^\top\mathcal{A}_1^\top+\mu\mathcal{A}_2^\top\mathcal{K}_\infty\mathcal{A}_1^\top\right]^{-1} \times\left[\mu\mathcal{A}_2^\top\left(g^o+\eta f^o-\mu\eta\mathcal{Z}_\infty g^o\right)\right] \quad (109)$$

where we introduced the matrix:

$$\mathcal{K}_\infty\triangleq\eta\mathcal{Z}_\infty+\mathcal{H}_\infty-\mu\eta\mathcal{Z}_\infty\mathcal{H}_\infty \quad (110)$$

Observe that if $\mathcal{Z}_\infty=0$, then $\mathcal{K}_\infty$ would be a symmetric matrix, which is the case studied in [30] in the context of unconstrained optimization. Here, the penalty functions introduce the additional factor $\mathcal{Z}_\infty$, in addition to f^o in (109).

Our goal now is to show that

$$\lim_{\mu\to 0}\frac{\|\mathbb{1}\otimes w^o-w_\infty\|}{\mu}=C \quad (111)$$

for some constant C that may be dependent on η (the approximation parameter), but not μ (the algorithm parameter). To begin with, we introduce the Jordan canonical decomposition of the matrix $A_2^\top A_1^\top=T^{-\top}DT^\top$ so that

$$\mathcal{A}_2^\top\mathcal{A}_1^\top=A_2^\top A_1^\top\otimes I_M=(T^{-\top}\otimes I_M)(D\otimes I_M)(T^\top\otimes I_M) \quad (112)$$

Then, we may re-write (109) as

$$\tilde{w}_\infty=(T^{-\top}\otimes I_M)[I_{MN}-D\otimes I_M+\mu E]^{-1} \times(T^\top\otimes I_M)\left[\mu\mathcal{A}_2^\top\left(g^o+\eta f^o-\mu\eta\mathcal{Z}_\infty g^o\right)\right] \quad (113)$$

$$E\triangleq(T^\top\otimes I_M)\mathcal{A}_2^\top\mathcal{K}_\infty\mathcal{A}_1^\top(T^{-\top}\otimes I_M) \quad (114)$$

By Assumption 4 we know that $A_2^\top A_1^\top$ is a doubly stochastic and primitive matrix. It follows from the Perron-Frobenius theorem [45, pp. 730–731] that $A_2^\top A_1^\top$ has a single eigenvalue at one with all other eigenvalues strictly inside the unit circle. Therefore, we may partition D and T as follows:

$$D=\mathrm{diag}\{1,D_0\},\quad T^\top=\mathrm{col}\{\mathbb{1}^\top,T_R\},\quad T^{-\top}=[\mathbb{1},T_L] \quad (115)$$

where D_0 has a block Jordan structure satisfying $\rho(D_0)<1$. Substituting (115) into (114), we can partition E into blocks:

$$E_{11}\triangleq(\mathbb{1}^\top\otimes I_M)\mathcal{A}_2^\top\mathcal{K}_\infty\mathcal{A}_1^\top(\mathbb{1}\otimes I_M) \quad (116)$$

$$E_{12}\triangleq(\mathbb{1}^\top\otimes I_M)\mathcal{A}_2^\top\mathcal{K}_\infty\mathcal{A}_1^\top(T_L\otimes I_M) \quad (117)$$

$$E_{21}\triangleq(T_R\otimes I_M)\mathcal{A}_2^\top\mathcal{K}_\infty\mathcal{A}_1^\top(\mathbb{1}\otimes I_M) \quad (118)$$

$$E_{22}\triangleq(T_R\otimes I_M)\mathcal{A}_2^\top\mathcal{K}_\infty\mathcal{A}_1^\top(T_L\otimes I_M) \quad (119)$$

where E_{ij} indicates the (i,j)-th block. Substituting into (113):

$$\tilde{w}_\infty=(T^{-\top}\otimes I_M)\begin{bmatrix}\mu E_{11} & \mu E_{12}\\ \mu E_{21} & I-D_0\otimes I_M+\mu E_{22}\end{bmatrix}^{-1} \times\begin{bmatrix}(\mu\mathbb{1}^\top\otimes I_M)\mathcal{A}_2^\top(g^o+\eta f^o)-\mu^2\eta(\mathbb{1}^\top\otimes I_M)\mathcal{A}_2^\top\mathcal{Z}_\infty g^o\\ \mu\cdot(T_R\otimes I_M)\mathcal{A}_2^\top(g^o+\eta f^o-\mu\eta\mathcal{Z}_\infty g^o)\end{bmatrix} \quad (120)$$

Furthermore, recalling that w^o is the solution to the minimization problem (14), we see that it is the root of

$$(\mathbb{1}^\top\otimes I_M)(g^o+\eta f^o)=0 \quad (121)$$

Using the fact that the matrix A_2 is doubly stochastic, expression (120) simplifies to

$$\tilde{w}_\infty=\mu\cdot(T^{-\top}\otimes I_M)\begin{bmatrix}\mu E_{11} & \mu E_{12}\\ \mu E_{21} & I-D_0\otimes I_M+\mu E_{22}\end{bmatrix}^{-1} \times\begin{bmatrix}-\mu\eta(\mathbb{1}^\top\otimes I_M)\mathcal{A}_2^\top\mathcal{Z}_\infty g^o\\ (T_R\otimes I_M)\mathcal{A}_2^\top\left(g^o+\eta f^o-\mu\eta\mathcal{Z}_\infty g^o\right)\end{bmatrix} \quad (122)$$

Let us denote

$$G\triangleq\begin{bmatrix}G_{11} & G_{12}\\ G_{21} & G_{22}\end{bmatrix}=\begin{bmatrix}\mu E_{11} & \mu E_{12}\\ \mu E_{21} & I-D_0\otimes I_M+\mu E_{22}\end{bmatrix}^{-1} \quad (123)$$

where, using the block inversion formula [46, p. 48]:

$$G_{11}=\mu^{-1}E_{11}^{-1}+E_{11}^{-1}E_{12}G_{22}E_{21}E_{11}^{-1} \quad (124)$$

$$G_{12}=-E_{11}^{-1}E_{12}G_{22} \quad (125)$$

$$G_{21}=-G_{22}E_{21}E_{11}^{-1} \quad (126)$$

$$G_{22}=\left(I-D_0\otimes I_M+\mu(E_{22}-E_{21}E_{11}^{-1}E_{12})\right)^{-1} \quad (127)$$

Observe that G is invertible since G^{-1} is similar to $I_{MN}-\mathcal{A}_2^\top\mathcal{A}_1^\top+\mu\mathcal{A}_2^\top\mathcal{K}_\infty\mathcal{A}_1^\top$ from (109), which we have already shown to be invertible when (65) and (72) are satisfied. Then, from (122), $\tilde{w}_\infty$ is given by:

$$\tilde{w}_\infty=\mu(T^{-\top}\otimes I_M)G\begin{bmatrix}-\mu\eta\cdot p_1\\ p_2\end{bmatrix} \quad (128)$$

$$p_1\triangleq(\mathbb{1}^\top\otimes I_M)\mathcal{A}_2^\top\mathcal{Z}_\infty g^o \quad (129)$$

$$p_2\triangleq(T_R\otimes I_M)\mathcal{A}_2^\top(g^o+\eta f^o-\eta\mu\mathcal{Z}_\infty g^o) \quad (130)$$

Substituting (124)–(127) into (128), we obtain

$$\tilde{w}_\infty = \mu(T^{-\mathsf{T}} \otimes I_M) \times \begin{bmatrix} -\eta E_{11}^{-1}p_1 - \mu\eta E_{11}^{-1}E_{12}G_{22}E_{21}E_{11}^{-1}p_1 - E_{11}^{-1}E_{12}G_{22}p_2 \\ \mu\eta G_{22}E_{21}E_{11}^{-1}p_1 + G_{22}p_2 \end{bmatrix} \tag{131}$$

Therefore,

$$\lim_{\mu\to 0} \frac{\|\tilde{w}_\infty\|}{\mu} = \lim_{\mu\to 0} \Big\|(T^{-\mathsf{T}} \otimes I_M) \times \begin{bmatrix} -\eta E_{11}^{-1}p_1 - \mu\eta E_{11}^{-1}E_{12}G_{22}E_{21}E_{11}^{-1}p_1 - E_{11}^{-1}E_{12}G_{22}p_2 \\ \mu\eta G_{22}E_{21}E_{11}^{-1}p_1 + G_{22}p_2 \end{bmatrix}\Big\|$$

But the right-hand-side is constant since the only matrices with dependence on μ are G_{22} and p_2, which satisfy:

$$G_{22,\infty} \triangleq \lim_{\mu\to 0} G_{22} = (I_{MN} - D_0 \otimes I_M)^{-1} \tag{132}$$

$$p_{2,\infty} \triangleq \lim_{\mu\to 0} p_2 = (T_R \otimes I_M)\mathcal{A}_2^{\mathsf{T}}(g^o + \eta f^o) \tag{133}$$

so we have

$$\lim_{\mu\to 0} \frac{\|\tilde{w}_\infty\|}{\mu} = \left\|(T^{-\mathsf{T}} \otimes I_M)\begin{bmatrix} I & -E_{11}^{-1}E_{12} \\ 0 & I \end{bmatrix}\begin{bmatrix} -\eta E_{11}^{-1}p_1 \\ G_{22,\infty}p_{2,\infty} \end{bmatrix}\right\| = d_1 \cdot \eta$$

for some constant $d_1 > 0$. We conclude that

$$\|\tilde{w}_\infty\|^2 \le d_2 \cdot (\mu\eta)^2 \tag{134}$$

for $d_2 > 0$. Therefore, the bias $P[\mathbb{1}_N \otimes w^o(\eta) - w_\infty]$ diminishes with μ^2. Since the bias appears in (91) through the vector b defined in (86), we conclude that $b \to (\sigma_v^2 + 2\alpha\|w^o(\eta)\|^2)\mathbb{1}_N$, a constant, at a rate of $(\mu\eta)^2$ and therefore (91) is on the order of μ. The second term of (75) is, as we just established, on the order of $(\mu\eta)^2$. We conclude, therefore that

$$\limsup_{i\to\infty} \mathbb{E}\,\|\mathbb{1}_N \otimes w^o(\eta) - \boldsymbol{w}_i\| \le c_1 \cdot \mu + c_2 \cdot (\mu\eta)^2 \tag{135}$$

for some constants $c_1, c_2 > 0$, which is (52).

ACKNOWLEDGMENT

The authors are thankful to J. Chen from the UCLA Adaptive Systems Laboratory for helpful feedback on the manuscript.

REFERENCES

[1] Z. J. Towfic and A. H. Sayed, "Adaptive stochastic convex optimization over networks," in *Proc. Allerton Conf.*, Allerton, IL, USA, Oct. 2013, pp. 1–6.

[2] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Distributed detection and estimation in wireless sensor networks," in *Academic Press Library in Signal Processing*, R. Chellapa and S. Theodoridis, Eds. New York, NY, USA: Academic Press/Elsevier, 2014, vol. 2, pp. 329–408.

[3] A. Bertrand and M. Moonen, "Seeing the bigger picture: How nodes can learn their place within a complex *ad hoc* network topology," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 71–82, May 2013.

[4] S. Lee and A. Nedic, "Distributed random projection algorithm for convex optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 221–229, Apr. 2013.

[5] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803–812, Sep. 1986.

[6] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA, USA: Athena Scientific, 1997.

[7] S. Kar, J. M. F. Moura, and K. Ramanan, "Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication," *IEEE Trans. Inf. Theory*, vol. 58, no. 6, pp. 3575–3605, Jun. 2012.

[8] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 97–123, Jan. 2011.

[9] N. Takahashi and I. Yamada, "Parallel algorithms for variational inequalities over the Cartesian product of the intersections of the fixed point sets of nonexpansive mappings," *J. Approx. Theory*, vol. 153, no. 2, pp. 139–160, Aug. 2008.

[10] P. M. Djuric and Y. Wang, "Distributed bayesian learning in multiagent systems: Improving our understanding of its capabilities and limitations," *IEEE Signal Process. Mag.*, vol. 29, no. 2, pp. 65–76, Mar. 2012.

[11] S. Cui, J.-J. Xiao, A. J. Goldsmith, Z.-Q. Luo, and H. V. Poor, "Estimation diversity and energy efficiency in distributed sensing," *IEEE Trans. Signal Process.*, vol. 55, no. 9, pp. 4683–4695, Sep. 2007.

[12] Y. Hu and A. Ribeiro, "Adaptive distributed algorithms for optimal random access channels," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2703–2715, Aug. 2011.

[13] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4692–4707, Oct. 2011.

[14] D. H. Dini and D. P. Mandic, "Cooperative adaptive estimation of distributed noncircular complex signals," in *Proc. Asilomar Conf.*, Pacific Grove, CA, USA, Nov. 2012, pp. 1518–1522.

[15] L. Li and J. A. Chambers, "A new incremental affine projection-based adaptive algorithm for distributed networks," *Signal Process.*, vol. 88, no. 10, pp. 2599–2603, Oct. 2008.

[16] O. N. Gharehshiran, V. Krishnamurthy, and G. Yin, "Distributed energy-aware diffusion least mean squares: Game-theoretic learning," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 5, pp. 821–836, Oct. 2013.

[17] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[18] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[19] A. Nedic and A. Ozdaglar, "Cooperative distributed multi-agent optimization," in *Convex optimization in Signal Processing and Communications*, D. P. Palomar and Y. C. Eldar, Eds. New York, NY, USA: Cambridge Univ. Press, 2010.

[20] F. Yan, S. Sundaram, S. V. N. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 11, pp. 2483–2493, Nov. 2013.

[21] S. S. Ram, A. Nedic, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optim. Theory Appl.*, vol. 147, no. 3, pp. 516–545, Jul. 2010.

[22] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing*, R. Chellapa and S. Theodoridis, Eds. New York, NY, USA: Academic Press/Elsevier, 2014, vol. 3, pp. 323–454.

[23] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.

[24] A. H. Sayed, "Adaptive networks," *Proc. IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[25] A. H. Sayed, *Adaptive Filters*. Hoboken, NJ, USA: Wiley, 2008.

[26] K. Srivastava and A. Nedic, "Distributed asynchronous constrained stochastic optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 772–790, Aug. 2011.

[27] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.

[28] B. T. Polyak, *Introduction to Optimization*. New York, NY, USA: Optimization Software, 1987.

[29] Z. J. Towfic, J. Chen, and A. H. Sayed, "On distributed online classification in the midst of concept drifts," *Neurocomput.*, vol. 112, pp. 138–152, Jul. 2013.

[30] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 205–220, Apr. 2013.

[31] S. Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[32] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*. New York, NY, USA: Wiley, 1993.

[33] J. Chen and A. H. Sayed, "On the limiting behavior of distributed optimization strategies," in *Proc. Allerton Conf. Commun., Control, Comput.*, Allerton, IL, USA, Oct. 2012, pp. 1535–1542.

[34] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks—Part I: Transient analysis," Dec. 2013 [Online]. Available: http://arxiv.org/abs/1312.7581, submitted for publication

[35] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.

[36] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM J. Optim.*, vol. 7, no. 4, pp. 913–926, Nov. 1997.

[37] A. Nedic and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM J. Optim.*, vol. 12, no. 1, pp. 109–138, Jul. 2001.

[38] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798–808, Apr. 2005.

[39] J.-W. Lee, S.-E. Kim, W.-J. Song, and A. H. Sayed, "Spatio-temporal diffusion strategies for estimation and detection over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 4017–4034, Aug. 2012.

[40] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York, NY, USA: Wiley, 1987.

[41] S. P. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM Rev.*, vol. 46, no. 4, pp. 667–689, Dec. 2004.

[42] X. Zhao and A. H. Sayed, "Performance limits of distributed estimation over LMS adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 1–16, Oct. 2012.

[43] E. M. Stein and R. Shakarchi, *Real Analysis: Measure Theory, Integration, and Hilbert Spaces*. Princeton, NJ, USA: Princeton Univ. Press, 2005.

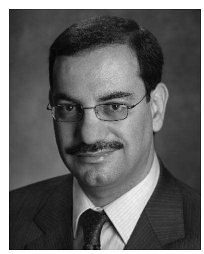
[44] E. Kreyszig, *Introductory Functional Analysis With Applications*. New York, NY, USA: Wiley, 1989.

[45] A. Papoulis and S. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. New York, NY, USA: McGraw-Hill, 2002.

[46] A. J. Laub, *Matrix Analysis for Scientists and Engineers*. Philadelphia, PA: SIAM, 2005.

Zaid J. Towfic (S'06) received his B.S. degrees in electrical engineering, computer science, and mathematics from the University of Iowa, Iowa City, IA, in 2007. He received his M.S. degree from the University of California, Los Angeles, CA, in 2009.

From 2007 to 2008, he worked for Rockwell Collins' Advanced Technology Center, Cedar Rapids, IA, working on MIMO, embedded sensing, and digital modulation classification. He is currently a Ph.D. candidate at the electrical engineering department of the University of California, Los Angeles, CA. His research interests include signal processing, self-healing circuitry, adaptive systems, stochastic optimization, and machine learning.



Ali H. Sayed (S'90–M'92–SM'99–F'01) is Professor and former Chairman of electrical engineering at the University of California, Los Angeles, where he directs the UCLA Adaptive Systems Laboratory. His research activities involve several areas including adaptation and learning, network science, information processing theories, distributed processing, statistical data analysis, and biologically-inspired designs. He is coauthor of the textbook *Linear Estimation* (Englewood Cliffs, NJ: Prentice-Hall, 2000), of the research monograph *Indefinite Quadratic Estimation and Control* (Philadelphia, PA: SIAM, 1999), and coeditor of *Fast Algorithms for Matrices with Structure* (Philadelphia, PA: SIAM, 1999). He is also the author of the textbooks *Fundamentals of Adaptive Filtering* (Hoboken, NJ: Wiley, 2003), and *Adaptive Filters* (Hoboken, NJ: Wiley, 2008). He has contributed several encyclopedia and handbook articles.

Prof. Sayed is also a Fellow of the American Association for the Advancement of Science (AAAS). His work has been recognized with several awards including the 2014 Athanasios Papoulis Award from the European Association for Signal Processing, the 2013 Meritorious Service Award and the 2012 Technical Achievement Award from the IEEE Signal Processing Society, the 2005 Terman Award from the American Society for Engineering Education, the 2003 Kuwait Prize, and the 1996 Donald G. Fink Prize from IEEE. He served as a Distinguished Lecturer of the IEEE Signal Processing Society in 2005. His articles received best paper awards from the IEEE Signal Processing Society in 2002, 2005, and 2012. He has been active in serving the Signal Processing community in various roles. Among other activities, he served as Editor-in-Chief of the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2003–2005), Editor-in-Chief of the *EURASIP Journal on Advances in Signal Processing* (2006–2007), General Chairman of ICASSP, Las Vegas, 2008, and Vice-President of Publications of the IEEE Signal Processing Society (2009–2011). He also served as member of the Board of Governors (2007–2011), Awards Board (2005), Publications Board (2003–2005), Conference Board (2007–2011), and Technical Directions Board (2008–2009) of the same Society.