Exact Diffusion Strategy for Optimization by Networked Agents

Kun Yuan, Bicheng Ying, Xiaochuan Zhao, and Ali H. Sayed
Department of Electrical Engineering, University of California, Los Angeles

***Abstract*—This work develops a distributed optimization algorithm with guaranteed exact convergence for a broad class of left-stochastic combination policies. The resulting exact diffusion strategy is shown to have a wider stability range and superior convergence performance than the EXTRA consensus strategy. The exact diffusion solution is also applicable to non-symmetric left-stochastic combination matrices, while most earlier developments on exact consensus implementations are limited to doubly-stochastic matrices or right-stochastic matrices; these latter policies impose stringent constraints on the network topology. Stability and convergence results are noted, along with numerical simulations to illustrate the conclusions.**

***Index Terms*—distributed optimization, diffusion, consensus, exact convergence, stochastic matrix, balanced policy.**

I. Introduction and Motivation

This work deals with *deterministic* optimization problems where a collection of N networked agents operate cooperatively to solve an aggregate optimization problem of the form:

$$w^o = \arg\min_{w\in\mathbb{R}^M} \quad \mathcal{J}^o(w) = \sum_{k=1}^{N} J_k(w). \tag{1}$$

In this formulation, each risk function $J_k(w)$ is convex and differentiable, while the aggregate cost $\mathcal{J}^o(w)$ is strongly-convex. All agents seek to determine the unique global minimizer, w^o, under the constraint that agents can only communicate with their neighbors. Problems of this type find applications in a wide range of areas.

There are several classes of distributed algorithms that can be used to solve problem (1). In the primal domain, implementations that are based on gradient-descent methods are effective and easy to implement. There are at least two prominent variants under this class: the consensus strategy [1], [2] and the diffusion strategy [3]–[5]. Primal algorithms are easy to implement, and enjoy fast convergence rate under constant step-size learning. These algorithms exhibit minimal bias and converge towards a neighborhood of the optimal solution, w^o, of square-error size $O(\mu^2)$. Another important family of distributed algorithms are those based on the distributed alternating direction method of multipliers (ADMM) [6] and its variants [7]. It is shown in [8] that distributed ADMM with constant penalty coefficients can converge exponentially fast to the *exact* global solution w^o. However, distributed ADMM solutions are computationally expensive since they necessitate the solution of optimal sub-problems at each iteration.

In the work [9], a modified implementation of consensus iterations, referred to as EXTRA, was shown to remove the bias and converge to the *exact* minimizer w^o rather than to an $O(\mu^2)-$neighborhood around w^o. Motivated by [9], other variations with similar properties were proposed in [10], [11]. These variations, compared to EXTRA, have two information combinations per recursion, which can be a burden when communication resources are limited. Moreover, while EXTRA [9] and ADMM-based algorithms [6], [7] require symmetric and doubly-stochastic combination matrices, the variations DIGing [10], ExtraPush [12], and Aug-DGM [11] require right-stochastic combination matrices. All these types of combination matrices impose stringent constraints on the network topology and communication protocols because each agent will need to be aware of its neighbors in advance.

The current work is motivated by the following considerations. The result in [9] shows that the EXTRA technique resolves the bias problem in consensus implementations. However, it is known that diffusion strategies outperform consensus strategies [3], [4]. Would it then be possible to correct the bias in the diffusion implementation and attain an algorithm that is superior to EXTRA in terms of a wider stability range and better performance/convergence? We answer this question in the affirmative in this article and provide three main contributions: (a) first, we develop a diffusion strategy that attains *exact* convergence for deterministic optimization problems; (b) we show that this strategy has a wider stability range and enhanced performance than EXTRA; and (c) we show that the proposed strategy is applicable to the larger, and also more practical class of left-stochastic matrices.

II. Diffusion and combination policies

We start our exposition by considering a more general optimization problem than (1). Specifically, we introduce a weighted aggregate cost of the form:

$$w^\star = \arg\min_{w\in\mathbb{R}^M} \quad \mathcal{J}^\star(w) = \sum_{k=1}^{N} q_k J_k(w), \tag{2}$$

for some positive coefficients $\{q_k\}$. Problem (1) is a special case when the q_k are uniform, i.e., $q_1 = q_2 = \ldots = q_N$, in

This work was supported in part by NSF grants CCF-1524250 and ECCS-1407712. Emails:{kunyuan,ybc,xiaochuanzhao, sayed}@ucla.edu

ISBN 978-0-9928626-7-1

which case $w^\star = w^o$. Note also that the aggregate cost $\mathcal{J}^\star(w)$ is strongly-convex when $\mathcal{J}^o(w)$ is strongly-convex. To solve problem (2) over a *strongly-connected* network of agents, we consider the standard (Adapt-then-Combine) diffusion strategy [3]–[5], which takes the following form:

$$\psi_{k,i} = w_{k,i-1} - \mu_k \nabla J_k(w_{k,i-1}), \tag{3}$$

$$w_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \psi_{\ell,i}, \tag{4}$$

where the $\{\mu_k\}_{k=1}^N$ are positive step-sizes, and the $\{a_{\ell k}\}_{\ell=1,k=1}^N$ are nonnegative combination weights satisfying

$$\sum_{\ell \in \mathcal{N}_k} a_{\ell k} = 1. \tag{5}$$

Here, the symbol $\mathcal{N}_k$ denotes the set of neighbors of agent k, and $\nabla J_k(\cdot)$ denotes the gradient vector of J_k relative to w. It follows from (5) that $A = [a_{\ell k}] \in \mathbb{R}^{N \times N}$ is a left-stochastic matrix. It also follows from the strong-connectedness of the graph that the matrix A is primitive. This implies, in view of the Perron-Frobenius theorem [4], that there exists an eigenvector p with positive entires satisfying

$$Ap = p, \quad \mathbb{1}_N^{\mathsf{T}} p = 1, \quad p \succ 0. \tag{6}$$

We refer to p as the Perron eigenvector of A.

Next, we introduce the vector $q = \mathrm{col}\{q_1, \ldots, q_N\} \in \mathbb{R}^N$, where q_k is the weight associated with $J_k(w)$ in (2). We also let $\beta > 0$ denote the constant that ensures the following equality:

$$q = \beta \, \mathrm{diag}\{\mu_1, \mu_2, \cdots, \mu_N\} p. \tag{7}$$

Condition (7) is not restrictive and a constant β can be chosen to ensure (7) as follows. Note first that β should satisfy $\beta = q_k/(p_k \mu_k)$ for all k. To make this expression for β independent of k, we assume the step-sizes are parameterized (or selected) as $\mu_k = \mu_o q_k / p_k$ for some small $\mu_o > 0$. Then, $\beta = 1/\mu_o$, which is independent of k, and relation (7) is satisfied.

It was shown by Theorem 3 in [13] that under (7), the iterates $w_{k,i}$ generated through the diffusion recursion (3)-(4) will approach $w^\star$ in the following sense:

$$\limsup_{i \to \infty} \|w^\star - w_{k,i}\|^2 = O(\mu_{\max}^2), \ \forall \ k = 1, \cdots, N, \tag{8}$$

where $\mu_{\max} = \max\{\mu_1, \cdots, \mu_N\}$. Result (8) is reassuring: it ensures that the squared-error is small whenever $\mu_{\max}$ is small; moreover, the result holds for *any* left-stochastic matrix.

Moving forward, we will focus on an important subclass of left-stochastic matrices, namely, those that satisfy a mild *local balance* condition (we shall refer to these matrices as *balanced* left-stochastic policies). The point is that we will be able to derive distributed optimization strategies with exact convergence guarantees for this sub-class of matrices (which is already significantly more relaxed than earlier results from the literature that are limited to the more stringent right- or doubly-stochastic class of matrices). We will also comment on how critical the balanced condition is. Thus, let $P = \mathrm{diag}\{p\}$ correspond to the diagonal matrix constructed from p. The matrix A is said to be balanced if it holds that

$$a_{\ell k} \, p_k = a_{k \ell} \, p_\ell, \quad k, \ell = 1, \cdots, N \tag{9}$$

or, equivalently, in matrix form:

$$PA^{\mathsf{T}} = AP. \tag{10}$$

TABLE I

Properties of balanced primitive left-stochastic matrices A

1. A is diagonalizable with *real* eigenvalues in $(-1, 1]$;
2. A has a single eigenvalue at 1;
3. $AP - P + I_N$ is symmetric, primitive, doubly-stochastic;
4. $P - AP$ is positive semi-definite;
5. $\mathrm{null}(P - AP) = \mathrm{span}(\mathbb{1}_N)$;
6. $\mathrm{null}(\mathcal{P} - \mathcal{A}\mathcal{P}) = \mathrm{span}\{\mathbb{1}_N \otimes I_M\}$.

Relations of the form (9) are common in the context of Markov chains. They are used there to model an equilibrium scenario for the probability flux into the Markov states, where the $\{a_{\ell k}\}$ represent the transition probabilities from states ℓ to k and the $\{p_\ell\}$ denote the steady-state distribution for the Markov chain.

We can provide an interpretation for (9) in the context of multi-agent networks by considering two generic agents, k and ℓ, from an arbitrary network, as shown in Fig. 1. The coefficient $a_{\ell k}$ is used by agent k to scale information arriving from agent ℓ. Therefore, this coefficient reflects the amount of confidence that agent k has in the information arriving from agent ℓ. Likewise, for $a_{k\ell}$. Since the combination policy is not necessarily symmetric, it will hold in general that $a_{\ell k} \neq a_{k\ell}$. However, agent k can re-scale the incoming weight $a_{\ell k}$ by p_k, and likewise for agent ℓ, so that the local balance condition (9) requires each pair of rescaled weights to match each other. We can interpret $a_{\ell k}$ to represent the (fractional) amount of information flowing from ℓ to k and p_k to represent the price paid by agent k for that information. Expression (9) is then requiring the information-cost benefit to be equitable across all linked agents.

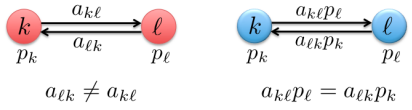

Fig. 1. Illustration of the local balance condition (9).

It can be shown that the local balancing condition (9) is satisfied by many important left-stochastic policies such as the Hastings rule, the averaging rule, the relative-degree rule, and various doubly-stochastic rules such as the Laplacian rule, maximum-degree rule, and the Metropolis rule [4]. It can be further shown that balanced left-stochastic matrices have several useful properties, which are listed in Table I without proof for lack of space. In the table, $\mathcal{P} = P \otimes I_M$ and $\mathcal{A} = A \otimes I_M$.

One may wonder whether exact convergence can be guaranteed for left-stochastic matrices that are not necessarily balanced. It turns out that one can provide examples of combination matrices that are left-stochastic (but not necessarily balanced) for which exact convergence occurs and others for which exact convergence does not not occur. In other words, exact convergence is not always guaranteed beyond the balanced class.

III. Development of Exact Diffusion

In this section, we reformulate the unconstrained optimization problem (2) into the equivalent constrained problem (17),

ISBN 978-0-9928626-7-1 © EURASIP 2017

which will be solved using a penalized formulation. This derivation will help clarify the origin of the $O(\mu_{\max}^2)$ bias from (8) in the standard diffusion implementation.

To begin with, note that the unconstrained problem (2) is equivalent to the following constrained problem:

$$\min_{\{w_k\}} \quad \sum_{k=1}^{N} q_k J_k(w_k), \quad \text{s.t.} \quad w_1 = \cdots = w_N. \tag{11}$$

Let $\mathcal{w} \triangleq \mathrm{col}\{w_1, \cdots, w_N\} \in \mathbb{R}^{NM}$ and

$$\mathcal{J}^\star(\mathcal{w}) \triangleq \sum_{k=1}^{N} q_k J_k(w_k), \quad \mathcal{J}^o(\mathcal{w}) \triangleq \sum_{k=1}^{N} J_k(w_k). \tag{12}$$

Using Property 6 from Table I, problem (11) is equivalent to

$$\min_{\mathcal{w} \in \mathbb{R}^{NM}} \quad \mathcal{J}^\star(\mathcal{w}), \quad \text{s.t.} \quad \frac{1}{2}(\mathcal{P} - \mathcal{A}\mathcal{P})\,\mathcal{w} = 0. \tag{13}$$

From Property 3 in Table I, we can decompose

$$\tfrac{1}{2}(P - AP) = U \Sigma U^{\mathsf{T}}, \tag{14}$$

where $\Sigma \in \mathbb{R}^{N \times N}$ is a non-negative diagonal matrix and $U \in \mathbb{R}^{N \times N}$ is an orthogonal matrix. If we introduce the symmetric square-root matrix

$$V \triangleq U \Sigma^{1/2} U^{\mathsf{T}} \in \mathbb{R}^{N \times N}, \quad \mathcal{V} \triangleq V \otimes I_M, \tag{15}$$

then it holds that

$$P - AP = 2V^2, \quad \mathcal{P} - \mathcal{A}\mathcal{P} = 2\mathcal{V}^2. \tag{16}$$

Using (16), problem (13) can be verified to be equivalent to

$$\min_{\mathcal{w} \in \mathbb{R}^{NM}} \quad \mathcal{J}^\star(\mathcal{w}), \quad \text{s.t.} \quad \mathcal{V}\mathcal{w} = 0. \tag{17}$$

In this way, we have transformed the original problem (2) to the equivalent constrained problem (17).

We now apply the primal-dual saddle point method to solve problem (17) directly. For this purpose, we first introduce the augmented Lagrangian function:

$$\begin{aligned} \mathcal{L}_a(\mathcal{w}, \mathcal{y}) &= \mathcal{J}^\star(\mathcal{w}) + \frac{1}{\alpha}\mathcal{y}^{\mathsf{T}}\mathcal{V}\mathcal{w} + \frac{1}{2\alpha}\|\mathcal{V}\mathcal{w}\|^2 \\ &\overset{(16)}{=} \mathcal{J}^\star(\mathcal{w}) + \frac{1}{\alpha}\mathcal{y}^{\mathsf{T}}\mathcal{V}\mathcal{w} + \frac{1}{4\alpha}\mathcal{w}^{\mathsf{T}}(\mathcal{P} - \mathcal{P}\mathcal{A}^{\mathsf{T}})\mathcal{w}, \end{aligned} \tag{18}$$

where $\mathcal{y} = \mathrm{col}\{y_1, \cdots, y_N\} \in \mathbb{R}^{NM}$ is the dual variable. The standard primal-dual saddle point algorithm [14] will then be:

$$\begin{cases} \mathcal{w}_i = \mathcal{w}_{i-1} - \alpha \nabla_{\mathcal{w}} \mathcal{L}_a(\mathcal{w}_{i-1}, \mathcal{y}_{i-1}), \\ \mathcal{y}_i = \mathcal{y}_{i-1} + \alpha \left(\dfrac{1}{\alpha}\mathcal{V}\mathcal{w}_i\right) = \mathcal{y}_{i-1} + \mathcal{V}\mathcal{w}_i. \end{cases} \tag{19}$$

The first recursion in (19) is the primal descent step while the second is the dual ascent step. Now, instead of performing the descent step directly as shown in the first recursion in (19), we perform it in an incremental manner. Thus, let

$$\mathcal{D}(\mathcal{w}) \triangleq \frac{1}{4\alpha}\mathcal{w}^{\mathsf{T}}(\mathcal{P} - \mathcal{P}\mathcal{A}^{\mathsf{T}})\mathcal{w}, \; \mathcal{C}(\mathcal{w}, \mathcal{y}) \triangleq \frac{1}{\alpha}\mathcal{y}^{\mathsf{T}}\mathcal{V}\mathcal{w}, \tag{20}$$

so that

$$\mathcal{L}_a(\mathcal{w}, \mathcal{y}_{i-1}) = \mathcal{J}^\star(\mathcal{w}) + \mathcal{D}(\mathcal{w}) + \mathcal{C}(\mathcal{w}, \mathcal{y}_{i-1}). \tag{21}$$

The diagonally incremental recursion that corresponds to the first step in (19) is then:

$$\begin{cases} \theta_i = \mathcal{w}_{i-1} - \alpha \mathcal{P}^{-1} \nabla \mathcal{J}^\star(\mathcal{w}_{i-1}), \\ \phi_i = \theta_i - \alpha \mathcal{P}^{-1} \nabla \mathcal{D}(\theta_i) = \dfrac{I_{MN} + \mathcal{A}^{\mathsf{T}}}{2}\,\theta_i = \overline{\mathcal{A}}^{\mathsf{T}}\theta_i, \\ \mathcal{w}_i = \phi_i - \alpha \mathcal{P}^{-1} \nabla_{\mathcal{w}} \mathcal{C}(\phi_i, \mathcal{y}_{i-1}) = \phi_i - \mathcal{P}^{-1}\mathcal{V}\mathcal{y}_{i-1}, \end{cases} \tag{22}$$

where in the second recursion of (22) we introduced

$$\overline{\mathcal{A}} \triangleq (I_{MN} + \mathcal{A})/2. \tag{23}$$

Algorithm 1 Exact diffusion strategy for agent k

Setting: Let $\overline{A} = (I_N + A)/2$, $w_{k,-1}$ be arbitrary and $\psi_{k,-1} = w_{k,-1}$

Repeat for $i = 0, 1, 2, \cdots$

$$\psi_{k,i} = w_{k,i-1} - \mu_k \nabla J_k(w_{k,i-1}), \quad \text{(adaptation)} \tag{24}$$

$$\phi_{k,i} = \psi_{k,i} + w_{k,i-1} - \psi_{k,i-1}, \quad \text{(correction)} \tag{25}$$

$$w_{k,i} = \sum_{\ell \in \mathcal{N}_k} \overline{a}_{\ell k}\phi_{\ell,i}. \quad \text{(combination)} \tag{26}$$

We further let $\alpha = 1/\beta$, where β is the constant in relation (7). It can be verified that

$$\alpha \mathcal{P}^{-1} \nabla \mathcal{J}^\star(\mathcal{w}_{i-1}) = \mathcal{M} \nabla \mathcal{J}^o(\mathcal{w}_{i-1}), \tag{27}$$

where $\mathcal{M} = \mathrm{diag}\{\mu_1, \cdots, \mu_N\} \otimes I_M$. With (27), if we substitute the first two recursions into the third one in (22), we get

$$\mathcal{w}_i = \overline{\mathcal{A}}^{\mathsf{T}}\Big(\mathcal{w}_{i-1} - \mathcal{M}\nabla \mathcal{J}^o(\mathcal{w}_{i-1})\Big) - \mathcal{P}^{-1}\mathcal{V}\mathcal{y}_{i-1}. \tag{28}$$

Replacing (19) with (28), the primal-dual saddle point recursion (19) becomes

$$\boxed{\begin{cases} \mathcal{w}_i = \overline{\mathcal{A}}^{\mathsf{T}}\Big(\mathcal{w}_{i-1} - \mathcal{M}\nabla \mathcal{J}^o(\mathcal{w}_{i-1})\Big) - \mathcal{P}^{-1}\mathcal{V}\mathcal{y}_{i-1} \\ \mathcal{y}_i = \mathcal{y}_{i-1} + \mathcal{V}\mathcal{w}_i \end{cases}} \tag{29}$$

Recursion (29) is the primal-dual form of the exact diffusion recursion we are seeking. For the initialization step, we set $y_{-1} = 0$ and $\mathcal{w}_{-1}$ to be any value, and hence for $i = 0$:

$$\begin{cases} \mathcal{w}_0 = \overline{\mathcal{A}}^{\mathsf{T}}\Big(\mathcal{w}_{-1} - \mathcal{M}\nabla \mathcal{J}^o(\mathcal{w}_{-1})\Big), \\ \mathcal{y}_0 = \mathcal{V}\mathcal{w}_0. \end{cases} \tag{30}$$

We can rewrite (29) in a simpler form by eliminating the dual variable $\mathcal{y}$ from the first recursion:

$$\boxed{\mathcal{w}_i = \overline{\mathcal{A}}^{\mathsf{T}}\Big(2\mathcal{w}_{i-1} - \mathcal{w}_{i-2} - \mathcal{M}\big(\nabla \mathcal{J}^o(\mathcal{w}_{i-1}) - \nabla \mathcal{J}^o(\mathcal{w}_{i-2})\big)\Big)} \tag{31}$$

Recursion (31) is the primal version of the exact diffusion. For comparison purposes, the EXTRA consensus recursion takes the following form but only when A is symmetric, doubly-stochastic and all agents employ the same step-size μ, whereas the exact diffusion recursion (31) is applicable more broadly to possibly non-symmetric balanced left-stochastic matrices and allows for heterogeneous step-sizes):

$$\mathcal{w}_i^e = \overline{\mathcal{A}}\Big(2\mathcal{w}_{i-1}^e - \mathcal{w}_{i-2}^e\Big) - \mu\big(\nabla \mathcal{J}^o(\mathcal{w}_{i-1}^e) - \nabla \mathcal{J}^o(\mathcal{w}_{i-2}^e)\big), \tag{32}$$

where we use the notation $\mathcal{w}_i^e$ to refer to the primal iterates in the EXTRA implementation.

We can rewrite the exact diffusion recursion (31) in a distributed form that resembles (3)–(4) more closely, as listed in Algorithm 1, where we denote the entries of $\overline{A}$ by $\overline{a}_{\ell k}$. It is observed that the resulting strategy resembles (3)–(4) to great extent, with the addition of a "correction" step between the adaptation and combination step. Note that the exact diffusion recursions (24)–(26) are synchronous, and they solve the differentiable problem (2). However, by following the idea in [15], it is easy to extend it to the asynchronous version and adjust them to solve the non-differentiable problem.

IV. Convergence Analysis

It can be shown that the iterates $w_{k,i}$ that result from the exact diffusion implementation (24)–(26) converge exactly to $w^\star$ at an exponential rate. The following two auxiliary lemmas,

ISBN 978-0-9928626-7-1 © EURASIP 2017

stated without proof, prepare for the statement of the general convergence result.

Lemma 1 (OPTIMALITY CONDITION)**.** *If condition* (7) *holds and block vectors* $(\mathcal{W}^\star, \mathcal{Y}^\star)$ *exist that satisfy:*

$$\overline{\mathcal{A}}^{\mathsf{T}}\mathcal{M}\nabla\mathcal{J}^o(\mathcal{W}^\star) + \mathcal{P}^{-1}\mathcal{V}\mathcal{Y}^\star = 0, \tag{33}$$

$$\mathcal{V}\mathcal{W}^\star = 0. \tag{34}$$

then it holds that the block entries of $\mathcal{W}^\star$ *satisfy:*

$$w_1^\star = w_2^\star = \cdots = w_N^\star = w^\star \tag{35}$$

where $w^\star$ *is the unique solution to problem* (2). ■

Observe that since $\mathcal{J}^\star(w)$ is assumed strongly-convex, then the solution to problem (2), $w^\star$, is unique, and hence $\mathcal{W}^\star$ is also unique. However, since $\mathcal{V}$ is rank-deficient, there can be multiple solutions $\mathcal{Y}^\star$ satisfying (35). Using an argument similar to [8], [9], we can show that among all possible $\mathcal{Y}^\star$, there is a unique solution $\mathcal{Y}_o^\star$ lying in the column span of $\mathcal{V}$.

Lemma 2 (PARTICULAR SOLUTION PAIR)**.** *When condition* (7) *holds and* $\mathcal{J}^o(w)$ *defined by* (1) *is strongly-convex, there exists a unique pair of variables* $(\mathcal{W}^\star, \mathcal{Y}_o^\star)$*, in which* $\mathcal{Y}_o^\star$ *lies in the range space of* $\mathcal{V}$*, that satisfies conditions* (33)-(34). ■

Using the above two lemmas, we can show that $(\mathcal{W}_i, \mathcal{Y}_i)$ generated through the exact diffusion recursion (29) will converge exponentially fast to $(\mathcal{W}^\star, \mathcal{Y}_o^\star)$. We first introduce a common assumption.

Assumption 1 (CONDITIONS ON COST FUNCTIONS)**.** *Each* $J_k(w)$ *is twice differentiable, and its Hessian matrix satisfies* $\nabla^2 J_k(w) \leq \delta I_M$*. Moreover, there exists at least one agent* k_o *such that* $J_{k_o}(w)$ *is* ν*-strongly convex, i.e.* $\nabla^2 J_{k_o}(w) > \nu I_M$*.* ■

We first define

$$\mathcal{W}^\star \triangleq \mathbb{1}_N \otimes w^\star,\ \widetilde{\mathcal{W}}_i \triangleq \mathcal{W}^\star - \mathcal{W}_i,\ \widetilde{\mathcal{Y}}_i \triangleq \mathcal{Y}_o^\star - \mathcal{Y}_i. \tag{36}$$

Theorem 1 (LINEAR CONVERGENCE)**.** *Suppose each cost function* $J_k(w)$ *satisfies Assumption 1, the left-stochastic matrix* A *satisfies the local balance condition* (9)*, and also condition* (7) *holds. Then, there exists a finite upper bound* $\bar{\mu} > 0$ *such that exact diffusion recursion* (29)*, or equivalently* (24)–(26)*, converges exponentially fast to* $(\mathcal{W}^\star, \mathcal{Y}_o^\star)$ *for step-sizes satisfying*

$$\mu_{\max} \leq \bar{\mu}, \tag{37}$$

Moreover, the convergence rate for the error variables is exponential and given by

$$\left\| \begin{bmatrix} \widetilde{\mathcal{W}}_i \\ \widetilde{\mathcal{Y}}_i \end{bmatrix} \right\|^2 \leq C\rho^i, \tag{38}$$

for some $C > 0$ *and* $\rho = 1 - O(\mu_{\max})$. ■

Expressions for $\bar{\mu}$ and ρ are derived in [16]. It is worth commenting on how this result compares to what is known for exact solutions that are based instead on consensus strategies. For instance, it was shown in [9], [16] that EXTRA consensus is stable for a range similar to (37), namely, $\mu < \bar{\mu}^e$ for some $\bar{\mu}^e$. It was however shown in [16] that $\bar{\mu} > \bar{\mu}^e$, which suggests that exact diffusion has a provably wider stability range (as illustrated by the simulation examples later in this article).

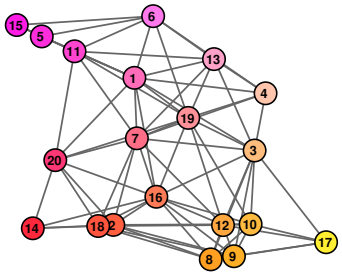

Fig. 2. Network topology used in the simulations.

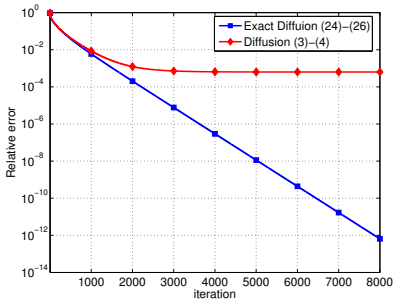

Fig. 3. Convergence comparison between standard diffusion and exact diffusion for distributed logistic regression (42).

Example. We illustrate this behavior by considering the following quadratic example resulting from a mean-square-error network [4]:

$$\min_{w\in\mathbb{R}^M} \frac{1}{2}\sum_{k=1}^{N}\left(w^{\mathsf{T}}R_{u,k}w - 2r_{du,k}^{\mathsf{T}}w\right), \tag{39}$$

where $R_{u,k}$ is positive definite. We can employ either the exact diffusion recursion (31) or the EXTRA recursion (32) to solve (39). To illustrate the stability issue, it is sufficient to consider a network with 2 agents and with diagonal Hessian matrices:

$$R_{u,1} = R_{u,2} = \sigma^2 I_M. \tag{40}$$

We assume the agents use the combination weights $\{a, 1-a\}$ with $a \in (0,1)$, so that

$$A = \begin{bmatrix} a & 1-a \\ 1-a & a \end{bmatrix} \in \mathbb{R}^{2\times 2}, \tag{41}$$

It can be verified for this example that exact diffusion is mean-square-error stable for any positive step-size satisfying $\mu\sigma^2 < 2$, while EXTRA consensus is unstable for step-sizes satisfying $\mu\sigma^2 \geq 1+a$. But since $1+a < 2$, we conclude that exact diffusion has a larger range of stability than EXTRA. In particular, if agents place small weights on their own data, i.e., when $a \approx 0$, the stability range for exact diffusion will be almost twice as large as that of EXTRA consensus. ■

V. NUMERICAL EXPERIMENTS

In this section we illustrate the performance of the exact diffusion algorithm (24)–(26). In all figures, the y-axis indicates the relative error, i.e., $\|\mathcal{W}_i - \mathcal{W}^o\|^2/\|\mathcal{W}_0 - \mathcal{W}^o\|^2$, where $\mathcal{W}_i = \mathrm{col}\{w_{1,i}, \cdots, w_{N,i}\} \in \mathbb{R}^{NM}$ and $\mathcal{W}^o = \mathrm{col}\{w^o, \cdots, w^o\} \in \mathbb{R}^{NM}$. All simulations employ the connected network topology with $N = 20$ nodes shown in Fig. 2.

We consider a pattern classification scenario. Each agent k holds local data samples $\{h_{k,j}, \gamma_{k,j}\}_{j=1}^{L}$, where $h_{k,j} \in \mathbb{R}^M$

ISBN 978-0-9928626-7-1 © EURASIP 2017

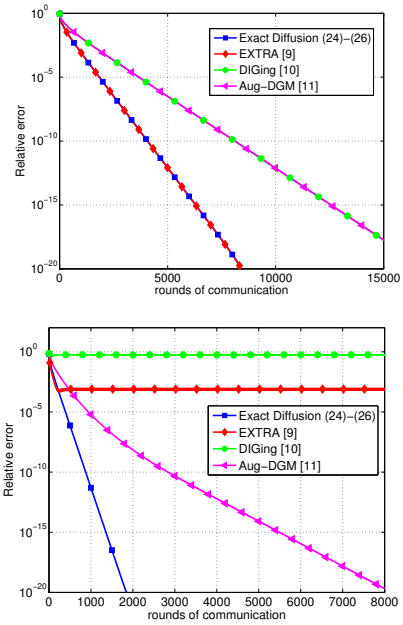

Fig. 4. Convergence comparison between exact diffusion (31), EXTRA [9], DIGing [10], and Aug-DGM [11] for distributed logistic regression problem (42). The step-size is chosen as $\mu = 0.01$ in the top plot, and, $\mu = 0.04$ in the bottom plot.

is a feature vector and $\gamma_{k,j} \in \{-1, +1\}$ is the corresponding label. Moreover, the value L is the number of local samples at each agent. All agents will cooperatively solve the regularized logistic regression problem:

$$\min_{w\in\mathbb{R}^M} \sum_{k=1}^{N} \Big[\frac{1}{L}\sum_{\ell=1}^{L} \ln\big(1+\exp(-\gamma_{k,\ell} h_{k,\ell}^{\mathsf{T}} w)\big) + \frac{\rho}{2}\|w\|^2\Big]. \quad (42)$$

In the experiments, we set $N = 20$, $M = 30$, and $L = 50$. For local data samples $\{h_{k,j}, \gamma_{k,j}\}_{j=1}^{L}$ at agent k, each $h_{k,j}$ is generated from the standard normal distribution $\mathcal{N}(0; \Lambda)$, where Λ is a diagonal matrix with each diagonal entry generated from the uniform distribution $\mathcal{U}(0, 1)$. To generate $\gamma_{k,j}$, we first generate an auxiliary random vector $w_0 \in \mathbb{R}^M$ with each entry following $\mathcal{N}(0, 1)$. Next, we generate $\gamma_{k,j}$ from a uniform distribution $\mathcal{U}(0, 1)$. If $\gamma_{k,j} \leq 1/[1 + \exp(-(h_{k,j})^{\mathsf{T}} w_0)]$ then $\gamma_{k,j}$ is set as $+1$; otherwise $\gamma_{k,j}$ is set as -1. We set $\rho = 0.1$.

We first compare the convergence behavior of standard diffusion (3)-(4) and exact diffusion (24)-(26). The left-stochastic matrix A is generated through the averaging rule, and each agent k employs $\mu_k = \mu_o/n_k$. The convergence of both algorithms is shown in Fig. 3. The step-size $\mu_o = 0.05$. It is observed that exact diffusion corrects the bias in standard diffusion.

In the second experiment, we compare exact diffusion with EXTRA consensus [9], DIGing [10], and Aug-DGM [11]. These algorithms require symmetric doubly-stochastic matrices or right-stochastic matrices. Therefore, we now consider a symmetric doubly stochastic matrix (the Metropolis rule). Moreover, there are two information combinations per iteration in DIGing and Aug-DGM algorithms, and each information combination corresponds to one round of communication. In comparison, there is only one information combination (or round of communication) in EXTRA consensus and exact diffusion. For fairness we will compare the algorithms based on the rounds of communications, rather than iterations. All agents employ the same step-size μ. When $\mu = 0.01$ is chosen in the top plot in Fig. 4, it is observed that all four algorithms converge exponentially to the solution w^o, and exact diffusion and EXTRA are almost twice as fast as DIGing and Aug-DGM. When a larger step-size $\mu = 0.04$ is chosen in the bottom plot in Fig. 4, it is observed that both exact diffusion and Aug-DGM are still able to converge linearly to w^o, while EXTRA and DIGing fail to do so. Moreover, exact diffusion is considerably faster than Aug-DGM.

REFERENCES

[1] 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.

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

[3] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, April 2014.

[4] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[5] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks—Part I: Transient analysis," *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3487–3517, 2015.

[6] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Püschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2718–2723, 2013.

[7] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "DLM: Decentralized linearized alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 4051–4064, 2015.

[8] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[9] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[10] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *arXiv:1607.03218*, Jul. 2016.

[11] A. Nedić, A. Olshevsky, W. Shi, and C. A. Uribe, "Geometrically convergent distributed optimization with uncoordinated step-sizes," *arXiv:1609.05877*, Sep. 2016.

[12] J. Zeng and W. Yin, "ExtraPush for convex smooth decentralized optimization over directed networks," *arXiv:1511.02942*, Nov. 2015.

[13] J. Chen and A. H. Sayed, "Distributed pareto optimization via diffusion strategies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 205–220, 2013.

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

[15] T. Wu, K. Yuan, Q. Ling, W. Yin, and A. H. Sayed, "Decentralized consensus optimization with asynchrony and delays," to appear in *IEEE Transactions on Signal and Information Processing over Networks*. See also arXiv:1612.00150, Dec. 2016.

[16] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact dffusion for distributed optimization and learning – Part II: Convergence analysis," *arXiv:1702.05142*, Feb. 2017.

ISBN 978-0-9928626-7-1 © EURASIP 2017