DECENTRALIZED EXACT COUPLED OPTIMIZATION

Sulaiman A. Alghunaim[⋆] *Kun Yuan*[⋆] *Ali H. Sayed*[†]

[⋆]Department of Electrical and Computer Engineering, University of California, Los Angeles
[†]School of Engineering, Ecole Polytechnique Federale de Lausanne, Switzerland

ABSTRACT

This work develops an exact converging algorithm for the solution of a distributed optimization problem with partially-coupled parameters across agents in a multi-agent scenario. In this formulation, while the network performance is dependent on a collection of parameters, each individual agent may be influenced by only a subset of the parameters. Problems of this type arise in several applications, most notably in distributed control formulations and in power system monitoring. The resulting coupled exact diffusion strategy is shown to converge to the true optimizer at a linear rate for strongly-convex cost functions.

***Index Terms*—Distributed optimization, exact diffusion, coupled optimization, multi-agent networks.**

I. INTRODUCTION

In most multi-agent formulations of distributed optimization problems, each agent generally has an individual cost function, denoted by $J_k(w)$, and the goal is to minimize the sum of costs:

$$\underset{w\in\mathbb{R}^M}{\text{minimize}} \quad \sum_{k=1}^{N} J_k(w) \tag{1}$$

In this statement, all individual costs depend on one common parameter, $w \in \mathbb{R}^M$, which all agents need to estimate and agree upon [1]–[6]. However, there exist extensive scenarios such as in web-search ranking [7], distributed model predictive control [8], [9], distributed wireless acoustic sensor networks [10], distributed wireless localization [11], and distributed power system monitoring [12], where each local cost may be a function of *multiple variables* that make up the entries of w. This situation motivates us to study a broader problem, where each local cost contains multiple variables that get to be learned by the network cooperatively.

Thus, consider a parameter vector $w \in \mathbb{R}^M$ and assume it is partitioned into L sub-blocks as $w \triangleq \text{col}\{w^1, w^2, ..., w^L\}$, with $w^\ell \in \mathbb{R}^{M_\ell}$. Without loss of generality, we assume the variables $\{w^\ell\}$ are different in that they do not share entries. Let $\mathcal{I}_k$ denote the set of variable indices that affect the cost of agent k and define:

$$w_k \triangleq \text{col}\{w^\ell\}_{\ell\in\mathcal{I}_k} \in \mathbb{R}^{Q_k}, \quad Q_k \triangleq \sum_{\ell\in\mathcal{I}_k} M_\ell \tag{2}$$

We are then interested in solving the following optimization problem:

$$\underset{w\in\mathbb{R}^M}{\text{minimize}} \quad J^{\text{glob}}(w) \triangleq \sum_{k=1}^{N} J_k(w_k) \tag{3}$$

We denote the optimal solution of (3) by $w^\star$:

$$w^\star \triangleq \text{col}\{w^{1,\star}, w^{2,\star}, ..., w^{L,\star}\} = \underset{w^1,...,w^L}{\arg\min} \sum_{k=1}^{N} J_k(w_k) \tag{4}$$

One important fact to emphasize here is that different agents may be influenced by common sub-vectors of w. Therefore, coupling between agents occurs and hence cooperation becomes useful and often necessary. Figure 1 illustrates the formulation for a simple network.

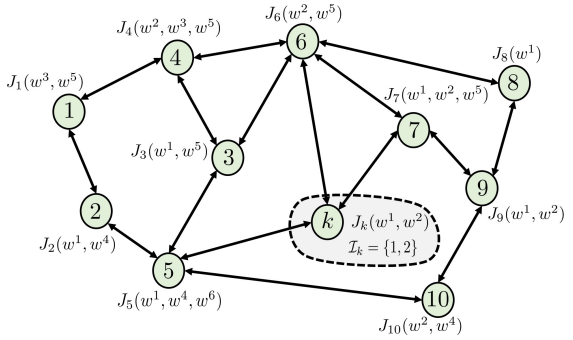

Fig. 1: A connected network of agents where the cost of each agent is a function of multiple parameters. Different agents generally depend on different sub-vectors of $w = [w^1, w^2, w^3, w^4, w^5, w^6]$. Cooperation is beneficial to promote correct inference across the network.

We remark that algorithms that solve (1) can be used to solve (3) by extending each local variable w_k into a longer global variable w, which would require unnecessary communications and memory allocation. This is because in (3), each local function contains only a subset of the global variable w. This approach can lead to performance degradation relative to the alternative solution proposed in

This work was supported in part by NSF grants ECCS-1407712 and CCF-1524250. Email: {salghunaim, kunyuan}@ucla.edu, and ali.sayed@epfl.ch

this work, which exploits more relaxed conditions — see the illustration and explanations in future Fig. 6. Therefore, solving (3) directly and effectively is important for large scale networks. On the other hand, algorithms that solve (3) are more general and can be used to solve (1). To see this, we let $L = 1$ and $\mathcal{I}_k = \{L\}, \forall\, k$, then problem (3) will depend only on one variable $w \triangleq w^L$. In this case, the cost function becomes $J^{\text{glob}}(w) \triangleq \sum_{k=1}^N J_k(w)$, which is of the same form as problem (1)

Distributed optimization problems of the type (3) have received less attention in the literature. Some related references are [12]–[16]. For example, in [12], an ADMM method is used to solve a distributed power system state estimation with constraints. In [13] an extended ADMM method is used to reduce communications but at the expense of stronger assumptions. In the model predictive control literature [9], [14], [15], most of the methods used are specific for the case where all agents that share a common variable w^ℓ form a star shaped subgraph. For example, in [14] another ADMM method is proposed, while [15] uses an inexact fast alternating minimization algorithm; this second method is equivalent to an inexact accelerated proximal-gradient method [16] applied to the dual problem. In all of these methods, a sub-minimization problem is solved at each iteration, which requires an inner iteration unless a closed form solution exists.

In this work, motivated by recent developments in [1], [3], [17], we propose a fully distributed first-order algorithm that does not involve inner minimization sub-problems and enjoys a linear convergence rate for strongly-convex cost functions. Our algorithm generalizes the exact diffusion strategy of [1], [3], [17] to the case of coupled parameters.

Notation: We use lowercase letters to denote vectors and scalars, and uppercase letters for matrices. We use $\text{diag}\{x_1, ..., x_N\}$ or $\text{diag}\{x_j\}_{j=1}^N$ to denote a (block) diagonal matrix consisting of diagonal entries (blocks) $x_1, ..., x_N$, and use $\text{col}\{x_1, ..., x_N\}$ or $\text{col}\{x_j\}_{j=1}^N$ to denote a column vector formed by stacking $x_1, ..., x_N$ on top of each other. For any set $\mathcal{X} = \{n_1, n_2, \cdots, n_x\}$, we let $U = [g_{ij}]_{i,j\in\mathcal{X}}$ denote a matrix that is formed as follows:

$$U = \begin{bmatrix} g_{n_1 n_1} & \cdots & g_{n_1 n_x} \\ \vdots & & \vdots \\ g_{n_x n_1} & \cdots & g_{n_x n_x} \end{bmatrix} \tag{5}$$

for some pre-defined scalars $\{g_{ij}\}_{i,j\in\mathcal{X}}$.

II. PROBLEM FORMULATION AND ALGORITHM DERIVATION

II-A. Problem Reformulation

In order to solve (3) in a distributed manner, we first need to reformulate (3) into an equivalent problem to account for one additional degree of freedom. Recall that the costs of two different agents, say, agents k and s, may depend on the same sub-vector, say, w^ℓ. Thus, we let w_k^ℓ denote the local copy of w^ℓ residing at agent k and let w_s^ℓ denote the local copy of the same w^ℓ residing at agent s. With this in mind, we redefine w_k from (2) in terms of the local copies, namely, we now write

$$w_k \triangleq \text{col}\{w_k^\ell\}_{\ell\in\mathcal{I}_k} \in \mathbb{R}^{Q_k} \tag{6}$$

We further let $\mathcal{C}_\ell$ denote the cluster (or sub-network) of nodes that is influenced by the variable w^ℓ i.e.,

$$\mathcal{C}_\ell \triangleq \{k \mid \ell \in \mathcal{I}_k\}. \tag{7}$$

It is necessary to require all local copies w_k^ℓ to coincide with each other, which is met by imposing the constraints:

$$w_k^\ell = w_s^\ell \ \ ,\forall\ k, s \in \mathcal{C}_\ell. \tag{8}$$

Using relations (6)–(8), we can rewrite problem (3) as:

$$\begin{aligned} \underset{w_1,\ldots,w_N}{\text{minimize}} \quad & \sum_{k=1}^N J_k(w_k) \\ \text{subject to} \quad & w_k^\ell = w_s^\ell \ \ \forall\, k, s \in \mathcal{C}_\ell,\ \forall\, \ell \in \{1, \cdots, L\}. \end{aligned} \tag{9}$$

We illustrate the above construction by means of an example.

Example: Consider the network with five agents shown in Figure 2a.

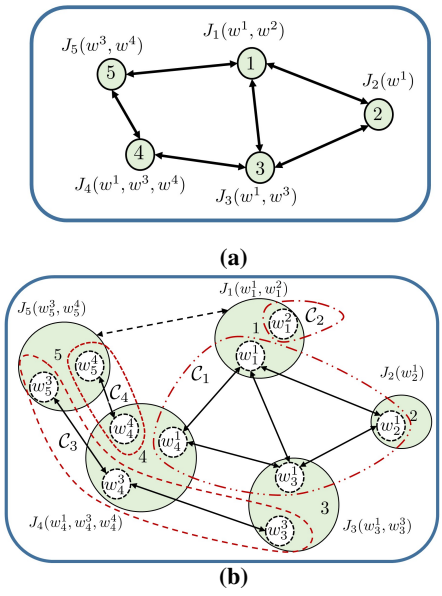

Fig. 2: (a) A 5-agent network to illustrate the setting of problem (9). (b) Cluster division of the network to highlight the common shared parameters across different agents. The connection between agent 1 and 5 is represented in dashed line to highlight the fact that they do not share any common parameters.

In this network, we have $w = \text{col}\{w^1, w^2, w^3, w^4\}$, $\mathcal{I}_1 = \{1,2\}, \mathcal{I}_2 = \{1\}$, $\mathcal{I}_3 = \{1,3\}$, $\mathcal{I}_4 = \{1,3,4\}$, and $\mathcal{I}_5 = \{3,4\}$. Consider further the optimization problem:

$$\min_{\{w^\ell\}} J_1(w^1, w^2) + J_2(w^1) + J_3(w^1, w^3) + J_4(w^1, w^3, w^4) + J_5(w^3, w^4) \tag{10}$$

To reformulate problem (10) into an equivalent problem that is amenable to distributed implementation, we introduce w_k^ℓ as the local copy of w^ℓ at agent k, and rewrite problem (10) as:

$$\begin{aligned} \text{minimize} \quad & J_1(w_1^1, w_1^2) + J_2(w_2^1) + J_3(w_3^1, w_3^3) + \\ & J_4(w_4^1, w_4^3, w_4^4) + J_5(w_5^3, w_5^4), \\ \text{subject to} \quad & w_1^1 = w_2^1 = w_3^1 = w_4^1 \\ & w_3^3 = w_4^3 = w_5^3 \\ & w_4^4 = w_5^4 \end{aligned} \tag{11}$$

We next introduce

$$w_1 \triangleq \text{col}\{w_1^1, w_1^2\}, \tag{12}$$

$$w_2 \triangleq \text{col}\{w_2^1\}, \tag{13}$$

$$w_3 \triangleq \text{col}\{w_3^1, w_3^3\}, \tag{14}$$

$$w_4 \triangleq \text{col}\{w_4^1, w_4^3, w_4^4\}, \tag{15}$$

$$w_5 \triangleq \text{col}\{w_5^3, w_5^4\} \tag{16}$$

and organize the network into $L = 4$ clusters with $\mathcal{C}_1 = \{1,2,3,4\}$, $\mathcal{C}_2 = \{1\}$, $\mathcal{C}_3 = \{3,4,5\}$, and $\mathcal{C}_4 = \{4,5\}$ as shown in Figure 2b. Each cluster $\mathcal{C}_\ell$ encircles the agents that depend on the corresponding parameter w^ℓ. Moreover, the links among the agents in each cluster $\mathcal{C}_\ell$ are defined by the links already existent in the network shown in Fig. 2a. Then, problem (11) becomes equivalent to

$$\begin{aligned} \underset{w_1, w_2, w_3, w_4}{\text{minimize}} \quad & \sum_{k=1}^{N} J_k(w_k), \\ \text{subject to} \quad & w_k^\ell = w_s^\ell, \ \forall\, k, s \in \mathcal{C}_\ell, \ \ell = 1,2,3,4. \end{aligned} \tag{17}$$

■

Remark 1. If we set $L = 1$ and $\mathcal{I}_k = \{1\}$, there will be only one cluster $\mathcal{C} = \{1, \cdots, N\}$ which is the network itself. Then, relation (6) will imply that $w_k = w_k^\ell$ (this is because $\mathcal{I}_k$ only contains one element). For this setting, problem (9) reduces to

$$\begin{aligned} \underset{w_1, \ldots, w_N}{\text{minimize}} \quad & \sum_{k=1}^{N} J_k(w_k) \\ \text{subject to} \quad & w_k = w_s \ \ \forall\, k, s \in \mathcal{C}, \end{aligned} \tag{18}$$

which is the problem formulation considered by exact diffusion in [3], [17]. ■

To solve (9), we associate with each cluster $\mathcal{C}_\ell$ a set of combination weights $\{a_{\ell,sk}\}_{s,k \in \mathcal{C}_\ell}$ such that:

$$\sum_{s \in \mathcal{C}_\ell} a_{\ell,sk} = 1, \quad \sum_{k \in \mathcal{C}_\ell} a_{\ell,sk} = 1 \tag{19}$$

$$a_{\ell,sk} \geq 0, \text{ and } a_{\ell,sk} = 0 \text{ if } s \notin \mathcal{N}_k \tag{20}$$

It should be noted that each agent k gets to choose its own combination weights. For example, let $n_{\ell,k} = \left|\mathcal{N}_k \cap \mathcal{C}_\ell\right|$ denote the number of agents that belong to $\mathcal{C}_\ell$ and are neighbors of agent k. Then, we can use the Metropolis rule to construct the combinations weights $\{a_{\ell,sk};\ s \in \mathcal{N}_k \cap \mathcal{C}_\ell,\ \ell \in \mathcal{I}_k\}$ that belong to agent k as follows:

$$a_{\ell,sk} = \begin{cases} \dfrac{1}{\max\{n_{\ell,k}, n_{\ell,s}\}}, & \text{if } s \in \mathcal{N}_k \cap \mathcal{C}_\ell,\ s \neq k \\ 1 - \displaystyle\sum_{r \in \mathcal{N}_k \cap \mathcal{C}_\ell \setminus \{k\}} a_{\ell,rk}, & s = k \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

Remark 2. Each agent is required to know the set $\mathcal{N}_k \cap \mathcal{C}_\ell$ for every $\ell \in \mathcal{I}_k$, i.e., to know the collection of neighboring agents that depend on the vector w^ℓ. This condition does not require agent k to know the agents in $\mathcal{C}_\ell$ beyond its neighborhood. In most networked problems of interest, this scenario is satisfied. For instance, in distributed wireless localization [11] and distributed model predictive control [9], [14] there are $L = N$ variables and it holds that $\mathcal{I}_k = \mathcal{C}_k = \mathcal{N}_k$ (see simulation section). Hence, the set $\mathcal{N}_k \cap \mathcal{C}_\ell$ for every $\ell \in \mathcal{I}_k$ can be easily known by agent k. ■

We now let N_ℓ denotes the cardinality of cluster $\mathcal{C}_\ell$ and define the $N_\ell \times N_\ell$ cluster combination matrices:

$$A_\ell \triangleq [a_{\ell,sk}]_{s,k \in \mathcal{C}_\ell}, \quad \forall\, \ell \in \{1, \cdots, L\} \tag{22}$$

We refer the reader to the notation section to see how construction (22) is formed. In-order to derive our distributed algorithm, we introduce the following assumption.

Assumption 1. *(**Each cluster is a connected subgraph**): The combinations submatrices $\{A_\ell\}$ are assumed to be primitive, i.e., we assume that there exists a large enough j_0 such that the elements of $A_\ell^{j_0}$ have strictly positive entries. This implies that for any two arbitrary agents in cluster $\mathcal{C}_\ell$, there exists at least one path with nonzero weights $\{a_{\ell,sk}\}_{s,k \in \mathcal{C}_\ell}$ linking one agent to the other. Moreover, at least one self weight $\{a_{\ell,kk}\}_{k \in \mathcal{C}_\ell}$ is nonzero. We further assume each A_ℓ to be symmetric and doubly stochastic.* ■

We note that the assumption that each cluster forms a connected network is not a stringent assumption. In many applications, this condition is automatically satisfied such as in maximum flow problems where it holds that $\mathcal{C}_\ell = \mathcal{N}_\ell$, which in turn implies that the $\mathcal{C}_\ell$ are connected [13], [14]. More generally, most networks of interest

are connected. Therefore, even if some cluster $\mathcal{C}_\ell$ is not connected, we can always construct a larger *connected* cluster $\mathcal{C}'_\ell$ such that $\mathcal{C}_\ell \subset \mathcal{C}'_\ell$. For example, consider the following network shown in Fig. 3.

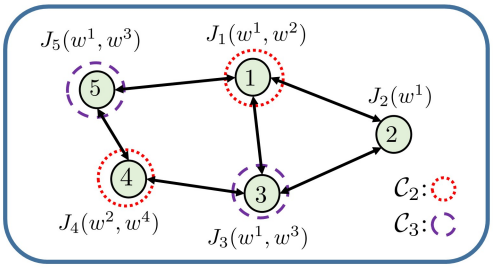

Fig. 3: A five-agent network with unconnected $\mathcal{C}_2$ and $\mathcal{C}_3$.

In this network, we have

$$\mathcal{C}_1 = \{1,2,3,5\},\ \mathcal{C}_2 = \{1,4\},\ \mathcal{C}_3 = \{3,5\},\ \mathcal{C}_4 = \{4\} \tag{23}$$

Note that $\mathcal{C}_4$ is a singleton. Therefore, w^4 will be optimized solely and separately by agent 4, and no communication is needed for that variable. Cluster $\mathcal{C}_1$ is connected, and agents $\{1,2,3,5\}$ cooperate in order to optimize w^1, with each agent sharing its estimate with neighbors. However, clusters $\mathcal{C}_2$ and $\mathcal{C}_3$ have disconnected graphs. This implies that agents 1 and 4 cannot communicate directly to optimize and reach consensus on w^2. Likewise, for agents $\{3,5\}$ regarding the variable w^3. To circumvent this issue, we can redefine the local costs $J_1(w^1, w^2)$ and $J_5(w^1, w^3)$ as

$$J'_1(w^1, w^2, w^3) \triangleq J_1(w^1, w^2) + 0 \cdot w^3 \tag{24}$$

$$J'_5(w^1, w^2, w^3) \triangleq J_5(w^1, w^3) + 0 \cdot w^2 \tag{25}$$

By doing so, the augmented costs $J'_1(w^1, w^2, w^3)$ and $J'_5(w^1, w^2, w^3)$ now involve w^3 and w^2, respectively, and the new clusters become

$$\mathcal{C}'_2 = \{1,4,5\}, \quad \mathcal{C}'_3 = \{1,3,5\} \tag{26}$$

which are connected and satisfy $\mathcal{C}_2 \subset \mathcal{C}'_2$ and $\mathcal{C}_3 \subset \mathcal{C}'_3$. Therefore, in this scenario, agents $\{1,4,5\}$ will now cooperate to optimize w^2 with agent 5 acting as a connection that allows information about w^2 to diffuse in the cluster. Likewise, for agents $\{1,3,5\}$, with agent 1 allowing information about w^3 to diffuse in the cluster. A second extreme approach would be to extend each local variable w_k to the global variable w, which reduces problem (3) to the formulation (1). We remark that the task of embedding smaller clusters into larger connected clusters can be achieved in a distributed fashion [13].

The following two auxiliary results are proven in [3].

Lemma 1. *For any $Q \times Q$ primitive, symmetric and doubly stochastic matrix A, it holds that $I_Q - A$ is symmetric and positive semi-definite. Moreover, if we introduce the eigen-decomposition $\frac{1}{2}(I_Q - A) = U\Sigma U^{\mathsf{T}}$, where U is orthogonal, and the symmetric square-root matrix:*

$$V \triangleq U\Sigma^{1/2}U^{\mathsf{T}} \in \mathbb{R}^{Q\times Q} \tag{27}$$

then, it holds that:

$$\text{null}(V) = \text{null}(I_Q - A) = \text{span}\{\mathbb{1}_Q\} \tag{28}$$

where $\mathbb{1}_Q$ denotes a column vector of size $Q \times 1$ with all its entries equal to one. ■

Corollary 1. *For the same setting of Lemma 1, let $\mathcal{A} = A \otimes I_M$, where $\otimes$ denotes the Kronecker product operation. Then, it holds that*

$$\text{null}(I - \mathcal{A}) = \text{span}\{\mathbb{1}_Q \otimes I_M\} \tag{29}$$

Moreover, for any block vector $x = \text{col}\{x^1, ..., x^Q\}$ in the nullspace of $I - \mathcal{A}$ with entries $x^k \in \mathbb{R}^M$ it holds that :

$$(I - \mathcal{A})x = 0 \iff x^1 = x^2 = ... = x^Q \tag{30}$$

If we further let $\mathcal{V} = V \otimes I_M$, then we have:

$$\mathcal{V}x = 0 \iff (I - \mathcal{A})x = 0 \iff x^1 = x^2 = ... = x^Q \tag{31}$$

■

Corollary 1 allows us to rewrite the constraints in (9) in an equivalent form that is amenable to distributed implementations. First, for each parameter vector w^ℓ, we collect its local copies into the extended vector

$$\mathcal{W}^\ell \triangleq \text{col}\{w_k^\ell\}_{k\in\mathcal{C}_\ell} \in \mathbb{R}^{N_\ell M_\ell} \tag{32}$$

With this notation, we can rewrite the cost function in problem (9) as

$$\mathcal{J}(\mathcal{W}^1, \mathcal{W}^2, \cdots, \mathcal{W}^\ell) \triangleq \sum_{k=1}^{N} J_k(w_k). \tag{33}$$

Now recalling that each cluster $\mathcal{C}_\ell$ is associated with a symmetric and doubly stochastic combination matrix A_ℓ defined in (22), we appeal to Lemma 1 to decompose

$$\frac{1}{2}(I_{N_\ell} - A_\ell) = U_\ell \Sigma_\ell U_\ell^{\mathsf{T}}. \tag{34}$$

If we let

$$V_\ell \triangleq U_\ell \Sigma_\ell^{1/2} U_\ell^{\mathsf{T}}, \tag{35}$$

$$\mathcal{V}_\ell \triangleq V_\ell \otimes I_{M_\ell}, \tag{36}$$

then using Corollary 1 and the definition of $\mathcal{W}^\ell$ in (32) we get

$$w_k^\ell = w_s^\ell,\ \forall\ k,s \in \mathcal{C}_\ell \iff \mathcal{V}_\ell \mathcal{W}^\ell = 0, \quad \forall\ \ell. \tag{37}$$

Using relations (33) and (37), we can rewrite problem (9) equivalently as

$$\begin{aligned} \underset{\mathcal{W}^1,\ldots,\mathcal{W}^L}{\text{minimize}} \quad & \mathcal{J}(\mathcal{W}^1, \cdots, \mathcal{W}^L) \\ \text{subject to} \quad & \mathcal{V}_\ell \mathcal{W}^\ell = 0,\ \forall\ \ell \end{aligned} \tag{38}$$

To rewrite problem (38) more compactly, we introduce

$$\mathcal{V} \triangleq \text{diag}\{\mathcal{V}_\ell\}_{\ell=1}^{L} \tag{39}$$

and

$$\mathcal{w} \triangleq \text{col}\{w^\ell\}_{\ell=1}^{L} \in \mathbb{R}^S \tag{40}$$

$$\mathcal{J}(\mathcal{w}) \triangleq \mathcal{J}(w^1, \cdots, w^L) \tag{41}$$

where $S \triangleq \sum_{\ell=1}^{L} N_\ell M_\ell$. Then, problem (38) becomes:

$$\underset{\mathcal{w}}{\text{minimize}} \quad \mathcal{J}(\mathcal{w}), \text{ s.t. } \mathcal{V}\mathcal{w} = 0 \tag{42}$$

For ease of reference, we summarize the notation in Table I.

$\mathcal{I}_k$	The set of variable indices that influence the cost of agent k.
w_k^ℓ	Local copy of w^ℓ at agent k.
w_k	Collection of parameters influencing agent k, $w_k \triangleq \text{col}\{w_k^\ell\}_{\ell \in \mathcal{I}_k}$
$\mathcal{C}_\ell$	Cluster of nodes that is influenced by the variable w^ℓ.
$\mathcal{w}^\ell$	Stacks all local copies of w^ℓ across $\mathcal{C}_\ell$, $\mathcal{w}^\ell = \text{col}\{w_k^\ell\}_{k \in \mathcal{C}_\ell}$
$\mathcal{w}$	Stacks $\mathcal{w}^\ell$ for all parameters, $\mathcal{w} = \text{col}\{\mathcal{w}^\ell\}_{\ell=1}^{L}$

Table I: A listing of the main symbols used in the problem formulation and their interpretation.

II-B. Algorithm Development

We can now arrive at the Coupled Exact Diffusion Algorithm (62a)–(62c) listed further ahead, by adjusting the arguments of [3]. We first note that

$$\mathcal{V}^2 = \text{diag}\{\mathcal{V}_\ell^2\}_{\ell=1}^{L} = \frac{1}{2}(I_S - \mathcal{A}) \tag{43}$$

where

$$\mathcal{A} \triangleq \text{diag}\{\mathcal{A}_\ell\}_{\ell=1}^{L}. \tag{44}$$

Next, we introduce the augmented Lagrangian:

$$\begin{aligned}\mathcal{L}_a(\mathcal{w}, \mathcal{y}) &= \mathcal{J}(\mathcal{w}) + \frac{1}{\mu}\mathcal{y}^{\mathsf{T}}\mathcal{V}\mathcal{w} + \frac{1}{2\mu}\|\mathcal{V}\mathcal{w}\|^2 \\ &= \mathcal{J}(\mathcal{w}) + \frac{1}{\mu}\mathcal{y}^{\mathsf{T}}\mathcal{V}\mathcal{w} + \frac{1}{4\mu}\mathcal{w}^{\mathsf{T}}(I - \mathcal{A})\mathcal{w}\end{aligned} \tag{45}$$

where $\mu > 0$ is a scaling parameter, and $\mathcal{y} = \text{col}\{\mathcal{y}^1, ..., \mathcal{y}^L\}$ is a dual variable with each block $\mathcal{y}^\ell = \text{col}\{y_k^\ell\}_{k \in \mathcal{C}_\ell} \in \mathbb{R}^{N_\ell M_\ell}$. Employing a standard primal-descent dual-ascent saddle point algorithm we get the following recursions using μ as a step-size parameter:

$$\mathcal{w}_i = \mathcal{w}_{i-1} - \mu\nabla_{\mathcal{w}}\mathcal{L}_a(\mathcal{w}_{i-1}, \mathcal{y}_{i-1}) \tag{46}$$

$$\mathcal{y}_i = \mathcal{y}_{i-1} + \mu\left(\frac{1}{\mu}\mathcal{V}\mathcal{w}_i\right) = \mathcal{y}_{i-1} + \mathcal{V}\mathcal{w}_i \tag{47}$$

The gradient vector appearing in (46) will involve three terms and, therefore, the update in (46) can be implemented in an incremental form. Specifically, referring to (45), let

$$\mathcal{D}(\mathcal{w}) = \frac{1}{4\mu}\mathcal{w}^{\mathsf{T}}(I - \mathcal{A})\mathcal{w}, \quad \mathcal{G}(\mathcal{w}, \mathcal{y}) = \frac{1}{\mu}\mathcal{y}^{\mathsf{T}}\mathcal{V}\mathcal{w} \tag{48}$$

so that:

$$\mathcal{L}_a(\mathcal{w}, \mathcal{y}_{i-1}) = \mathcal{J}(\mathcal{w}) + \mathcal{D}(\mathcal{w}) + \mathcal{G}(\mathcal{w}, \mathcal{y}_{i-1}) \tag{49}$$

All three terms on the right-hand side depend on $\mathcal{w}$. We can then implement the gradient descent operation in (46) in three successive steps and obtain the incremental form:

$$\theta_i = \mathcal{w}_{i-1} - \mu\nabla_{\mathcal{w}}\mathcal{J}(\mathcal{w}_{i-1}) \tag{50}$$

$$\phi_i = \theta_i - \mu\nabla_{\mathcal{w}}\mathcal{D}(\theta_i) = \frac{1}{2}(I_S + \mathcal{A})\,\theta_i = \bar{\mathcal{A}}\theta_i \tag{51}$$

$$\mathcal{w}_i = \phi_i - \mu\nabla_{\mathcal{w}}\mathcal{G}(\phi_i, \mathcal{y}_{i-1}) = \phi_i - \mathcal{V}\mathcal{y}_{i-1} \tag{52}$$

where in (51) we introduced :

$$\bar{\mathcal{A}} \triangleq \frac{1}{2}(I_S + \mathcal{A}) \tag{53}$$

Now if we substitute (50) and (51) into (52) we get:

$$\mathcal{w}_i = \bar{\mathcal{A}}\Big(\mathcal{w}_{i-1} - \mu\nabla_{\mathcal{w}}\mathcal{J}(\mathcal{w}_{i-1})\Big) - \mathcal{V}\mathcal{y}_{i-1} \tag{54}$$

Replacing (46) with (54), the resulting algorithm becomes:

$$\begin{cases} \mathcal{w}_i = \bar{\mathcal{A}}\Big(\mathcal{w}_{i-1} - \mu\nabla_{\mathcal{w}}\mathcal{J}(\mathcal{w}_{i-1})\Big) - \mathcal{V}\mathcal{y}_{i-1} \\ \mathcal{y}_i = \mathcal{y}_{i-1} + \mathcal{V}\mathcal{w}_i \end{cases} \tag{55}$$

We can rewrite (55) in a simpler form by eliminating the dual variable $\mathcal{y}$. First, we initialize $\mathcal{y}_{-1} = 0$ and $\mathcal{w}_{-1}$ to any value. Hence, for $i = 0$ we have:

$$\begin{cases} \mathcal{w}_0 = \bar{\mathcal{A}}\Big(\mathcal{w}_{-1} - \mu\nabla_{\mathcal{w}}\mathcal{J}(\mathcal{w}_{-1})\Big) \\ \mathcal{y}_0 = \mathcal{V}\mathcal{w}_0 \end{cases} \tag{56}$$

Moreover, by subtracting two successive iterations of (55) for $i = 1, 2, \ldots$ we get:

$$\begin{aligned}&\mathcal{w}_i - \mathcal{w}_{i-1} = -\mathcal{V}(\mathcal{y}_{i-1} - \mathcal{y}_{i-2}) + \\ &\bar{\mathcal{A}}\Big(\mathcal{w}_{i-1} - \mathcal{w}_{i-2} - \mu\big(\nabla_{\mathcal{w}}\mathcal{J}(\mathcal{w}_{i-1}) - \nabla_{\mathcal{w}}\mathcal{J}(\mathcal{w}_{i-2})\big)\Big)\end{aligned} \tag{57}$$

From the second step in (55) we have:

$$\mathcal{V}(\mathcal{y}_{i-1} - \mathcal{y}_{i-2}) = \mathcal{V}^2\mathcal{w}_{i-1} = \frac{1}{2}\big(I_S - \mathcal{A}\big)\mathcal{w}_{i-1} \tag{58}$$

Substituting (58) into (57), we arrive at:

$$\mathcal{w}_i = \bar{\mathcal{A}}\Big(2\mathcal{w}_{i-1} - \mathcal{w}_{i-2} - \mu\big(\nabla_{\mathcal{w}}\mathcal{J}(\mathcal{w}_{i-1}) - \nabla_{\mathcal{w}}\mathcal{J}(\mathcal{w}_{i-2})\big)\Big) \tag{59}$$

Algorithm (59) looks similar to the one in [3]. However, there are two subtle differences. First, the combination matrix $\bar{\mathcal{A}} = \frac{1}{2}(I_S + \text{diag}\{\mathcal{A}_\ell\}_{\ell=1}^{L})$ has a block diagonal structure and, second, the gradient $\nabla_{\mathcal{w}}\mathcal{J}(\mathcal{w})$ couples the variables $\{\mathcal{w}^\ell\}$ across the clusters. To see this, we note that the gradient vector is given by

$$\nabla_{\mathcal{w}}\mathcal{J}(\mathcal{w}) = \begin{bmatrix} \nabla_{w^1}\mathcal{J}(\mathcal{w}) \\ \vdots \\ \nabla_{w^L}\mathcal{J}(\mathcal{w}) \end{bmatrix} \tag{60}$$

with each $\nabla_{\mathcal{W}^\ell}\mathcal{J}(\mathcal{W})$ having the following form:

$$\nabla_{\mathcal{W}^\ell}\mathcal{J}(\mathcal{W}) = \mathrm{col}\{\nabla_{w_k^\ell} J_k(w_k)\}_{k\in\mathcal{C}_\ell} \tag{61}$$

It is clear that each block $\mathrm{col}\{\nabla_{w_k^\ell} J_k(w_k)\}_{k\in\mathcal{C}_\ell}$ depends on other clusters since the argument in $J_k(w_k)$ is w_k and agent k may belong to more than one cluster. For the special case that there exists only one cluster (i.e, $L=1$, $w_k = w_k^1$, and $\mathcal{A} = \mathcal{A}_1$), we recover the Algorithm in [3]. We can rewrite (59) in an equivalent distributed form, as listed in (62a)–(62c). In this listing, the variables $\psi_{k,i}$ and $\phi_{k,i}$ have the same structure as $w_{k,i}$, i.e., $\psi_{k,i} = \mathrm{col}\{\psi_{k,i}^\ell\}_{\ell\in\mathcal{I}_k}$ and $\phi_{k,i} = \mathrm{col}\{\phi_{k,i}^\ell\}_{\ell\in\mathcal{I}_k}$.

Algorithm (Coupled Exact Diffusion Strategy)

Setting: Let $\bar{A}_\ell = (I_{N_\ell} + A_\ell)/2$, and $w_{k,-1} = \psi_{k,-1}$ arbitrary. For every agent k, repeat for $i = 0, 1, 2, ...$

$$\psi_{k,i} = w_{k,i-1} - \mu\nabla_{w_k} J_k(w_{k,i-1}) \tag{62a}$$

$$\phi_{k,i} = \psi_{k,i} + w_{k,i-1} - \psi_{k,i-1} \tag{62b}$$

$$w_{k,i}^\ell = \sum_{s\in\mathcal{N}_k\cap\mathcal{C}_\ell} \bar{a}_{\ell,sk}\phi_{s,i}^\ell, \;\; \forall\, \ell\in\mathcal{I}_k \tag{62c}$$

Before we examine the convergence properties of the proposed algorithm, we introduce the following common assumption.

Assumption 2. *(**Individual Costs**): It is assumed that the individual cost functions $J_k(w_k)$ are each twice-differentiable, convex, and have Hessian matrices that are bounded from above:*

$$\nabla_{w_k}^2 J_k(w_k) \le \lambda_{k,\max} I_{M_k} \tag{63}$$

Moreover, for every cluster $\mathcal{C}_\ell$ there exists at least one agent k_ℓ such that:

$$\nabla_{w_{k_\ell}}^2 J_{k_\ell}(w_{k_\ell}) > \lambda_{\ell,\min} \tag{64}$$

for some strictly positive scalars $\{\lambda_{\ell,\min}\}$ and $\{\lambda_{k,\max}\}$. ■

Note that assumption (64) is similar to requiring at least one of the costs $J_k(.)$ to be strongly convex within each cluster – see [1], [2]. This guarantees that the aggregate cost is strongly convex, and therefore a unique minimizer exists.

Lemma 2. *(**An optimality condition**) If block vectors $(\mathcal{W}^\star, \mathcal{Y}^\star)$ exist that satisfy :*

$$\mu\bar{\mathcal{A}}\nabla_{\mathcal{W}}\mathcal{J}(\mathcal{W}^\star) + \mathcal{V}\mathcal{Y}^\star = 0 \tag{65}$$

$$\mathcal{V}\mathcal{W}^\star = 0 \tag{66}$$

then, it holds that each entry in each sub-block of the vector $\mathcal{W}^\star$ (i.e., block entries of $\mathcal{W}^{\ell,\star}$) satisfy:

$$w_k^{\ell,\star} = w^{\ell,\star}, \;\; k\in\mathcal{C}_\ell \tag{67}$$

where $w^{\ell,\star}$ is the ℓ-th block of $w^\star$ defined in (4)*, the unique solution of problem* (3)*.*

Proof: First let $\mathcal{W}^{\ell,\star} \triangleq \mathrm{col}\{w_k^{\ell,\star}\}_{\ell\in\mathcal{C}_\ell}$ and note that $\mathcal{V}\mathcal{W}^\star = \mathrm{col}\{\mathcal{V}_\ell\mathcal{W}^{\ell,\star}\}_{\ell=1}^L$. Therefore, from (37) we have:

$$\mathcal{V}\mathcal{W}^\star = 0 \iff w_k^{\ell,\star} = w_s^{\ell,\star}, \;\; \forall\, k,s\in\mathcal{C}_\ell \tag{68}$$

We now show that $w_k^{\ell,\star} = w^{\ell,\star}$. Let $\mathcal{Z} = \mathrm{diag}\{\mathbb{1}_{N_\ell}\otimes I_{M_\ell}\}_{\ell=1}^L$ and multiply (65) by $\mathcal{Z}^\mathsf{T}$ from the left:

$$\begin{aligned} 0 &= \mu\mathcal{Z}^\mathsf{T}\bar{\mathcal{A}}\nabla_{\mathcal{W}}\mathcal{J}(\mathcal{W}^\star) + \mathcal{Z}^\mathsf{T}\mathcal{V}\mathcal{Y}^\star \\ &\overset{(a)}{=} \mu\mathcal{Z}^\mathsf{T}\bar{\mathcal{A}}\nabla_{\mathcal{W}}\mathcal{J}(\mathcal{W}^\star) \\ &\overset{(b)}{=} \mu\begin{bmatrix} \sum_{k\in\mathcal{C}_1}\nabla_{w_k^1}J_k(w_k^\star) \\ \vdots \\ \sum_{k\in\mathcal{C}_L}\nabla_{w_k^L}J_k(w_k^\star)\end{bmatrix} \end{aligned} \tag{69}$$

where step (a) is because $\mathcal{Z}^\mathsf{T}\mathcal{V} = 0$, since $\mathcal{Z}$ is in the null space of $\mathcal{V}$ and step (b) is because of (60)–(61) and the fact that $\mathcal{Z}^\mathsf{T}\bar{\mathcal{A}} = \mathcal{Z}^\mathsf{T}$. Equations (68) and (69) are the optimality conditions for problem (9). Therefore, we conclude that for every k, the entries $\{w_k^{\ell,\star}\}$, which are identical, must coincide with the minimizer $w^{\ell,\star}$ which is the ℓ-th block of the minimzer $w^\star$ of problem (3). ■

We remark that $\mathcal{W}^\star$ is unique due to the fact that $w^\star$ is unique since $J^{\mathrm{glob}}(w)$ is assumed strongly convex. However, $\mathcal{Y}^\star$ is not necessarily unique due the fact that $\mathcal{V}$ can be rank-deficient. It can be shown that there exists a unique solution $\mathcal{Y}_o^\star$ lying in the span of $\mathcal{V}$.

Lemma 3. *(**Particular solution pair**) When $\mathcal{J}(\mathcal{W})$ is strongly convex and the combination matrices $\{A_\ell\}$ are primitive, symmetric, and doubly stochastic, 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 $\mathcal{V}$, and the optimality conditions (65)–(66) are satisfied.*

Proof: Omitted for brevity — see the arguments in [3], [17] though. ■

Theorem 1. *(**Linear convergence**): Suppose Assumptions 1 and 2 hold, then the coupled exact diffusion algorithm* (55) *converges exponentially fast to $(\mathcal{W}^\star, \mathcal{Y}_o^*)$ for step-sizes $\mu\le\mu_0$ for some small enough μ_0.*

Proof: Omitted for brevity — see the arguments in [17]. ■

III. EXAMPLE AND SIMULATION RESULTS

In this section we illustrate the operation of the algorithm by considering a situation in which the individual costs are quadratic. Each agent k seeks to estimate its own variable $w^k\in\mathbb{R}^{M_k}$ but is affected by the neighboring variables $\{w^\ell;\ell\in\mathcal{N}_k\}$ (i.e., $L=N$ and $\mathcal{I}_k=\mathcal{N}_k$), through a cost of the form:

$$\begin{aligned} J_k(w_k) &= w_k^\mathsf{T}R_k w_k + b_k^\mathsf{T}w_k + r_k \\ &= \sum_{s\in\mathcal{N}_k}\sum_{\ell\in\mathcal{N}_k}(w^s)^\mathsf{T}R_{k,s\ell}w^\ell + \sum_{\ell\in\mathcal{N}_k} b_{k,\ell}^\mathsf{T}w^\ell + r_k \end{aligned} \tag{70}$$

where $w_k \triangleq \text{col}\{w^\ell\}_{\ell\in\mathcal{N}_k}$, R_k is a $Q_k \times Q_k$ positive definite matrix, and $b_k \in \mathbb{R}^{Q_k}$. We partition R_k and b_k into block matrices $\{R_{k,s\ell} \in \mathbb{R}^{M_s\times M_\ell}\}$ and block vectors $\{b_{k,\ell} \in \mathbb{R}^{M_\ell}\}$ according to the block structure of w_k. Each agent k only knows its local quantities $\{R_k, b_k\}$. The aggregate cost is given by

$$J^{\text{glob}}(w) \triangleq \sum_{k=1}^{N} J_k(w_k) = w^{\mathsf{T}}\mathcal{R}w + \bar{b}^{\mathsf{T}}w \qquad (71)$$

where

$$\mathcal{R} \triangleq \begin{bmatrix} \sum_{k=1}^{N} R_{k,11} & \cdots & \sum_{k=1}^{N} R_{k,1L} \\ \vdots & & \vdots \\ \sum_{k=1}^{N} R_{k,L1} & \cdots & \sum_{k=1}^{N} R_{k,LL} \end{bmatrix}, \qquad (72)$$

$$\bar{b} \triangleq \begin{bmatrix} \sum_{k=1}^{N} b_{k,1} \\ \vdots \\ \sum_{k=1}^{N} b_{k,L} \end{bmatrix} \qquad (73)$$

with

$$R_{k,s\ell} = 0,\ b_{k,\ell} = 0,\ \text{if } \ell \notin \mathcal{N}_k \text{ or } s \notin \mathcal{N}_k \qquad (74)$$

Cost functions of the type (71) are common in the control literature, specifically in distributed linear quadratic regulator (LQR) problems [9], [14]. In our simulation, we consider a randomly generated network with $N = 20$ agents shown in Figure 4, where neighbors are decided by closeness in distance. We randomly generate R_k and b_k. The matrices $\{A_\ell\}$ are generated using the Metropolis rule (21). Figure 5 shows the relative error $(\|\mathcal{W}_i - \mathcal{W}^\star\|^2/\|\mathcal{W}_{-1} - \mathcal{W}^\star\|^2)$ with $M_\ell = 10$ for all variables and step size $\mu = 0.015$. We observe that the coupled exact diffusion algorithm (62a)–(62c) converges linearly to $\mathcal{W}^\star$. Figure 6 compares the proposed algorithm to the exact diffusion algorithm [3], [17]. In this figure we used $M_\ell = 5$ for all variables and step size $\mu = 0.01$ for both algorithms. We also used the Metropolis rule to create the combination matrices. We conclude that, in the case of problem formulation (3), it is not efficient to extend each local vector to the global one and then solve this extended problem [3], [17]. This can be reasoned as follows. First, extending each local vector implies that each Hessian matrix has a zero eigenvalue, which destroys the strong convexity of the individual costs used in this simulation. In comparison, the proposed coupled exact diffusion algorithm takes advantage of the strong convexity of the individual cost. Second, each agent using exact diffusion combines every entry with the same weights, which limits the flexibility of choosing more weights for more important entries. For example, agent 8 in Fig. 4 using exact diffusion, estimate the entire $w_{8,i} = \{w_{8,i}^\ell\}_{\ell=1}^{20}$ by sharing and combining all entries of $\phi_{8,i} = \{\phi_{8,i}^\ell\}_{\ell=1}^{20}$ according to the combination step $w_{8,i} = a_{8,8}\phi_{8,i} + a_{19,8}\phi_{19,i}$ even though it only contains information about the parameters w^8 and w^{19}. Thus the combination weights $a_{8,8}$ and $a_{19,8}$ are required to satisfy $a_{19,8} = a_{8,19}$, $a_{8,19} + a_{10,19} + a_{18,19} + a_{8,19} = 1$, and $a_{8,8} + a_{19,8} = 1$. From this we see that $a_{8,8}$ and $a_{19,8}$ can not be chosen independently (i.e, they depend on the weights $\{a_{10,19}, a_{18,19}, a_{19,19}\}$ which relate to agents 10 and 18). This is not the case in the proposed coupled diffusion algorithm since each cluster combination weights are chosen independently (e.g., agent 8 combination weights for $w_{8,i}^8$, $a_{8,8,8}$ and $a_{8,19,8}$ are required to satisfy $a_{8,19,8} = a_{8,8,19}$ and $a_{8,8,8} + a_{8,19,8} = 1$, and hence unlike exact diffusion, they do not depend on agents 10 and 18, which gives more freedom in choosing the weights). Moreover, in each iteration of Figure 6 each agent k using the exact diffusion algorithm needs to communicate an $5\times N = 100$ long vector, as opposed to the proposed algorithm where each agent k only communicates $5 \times |\mathcal{N}_k| < 100$.

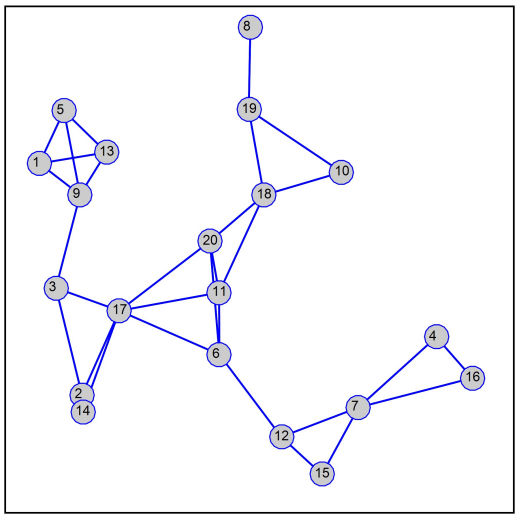

Fig. 4: Network topology used in the simulation results.

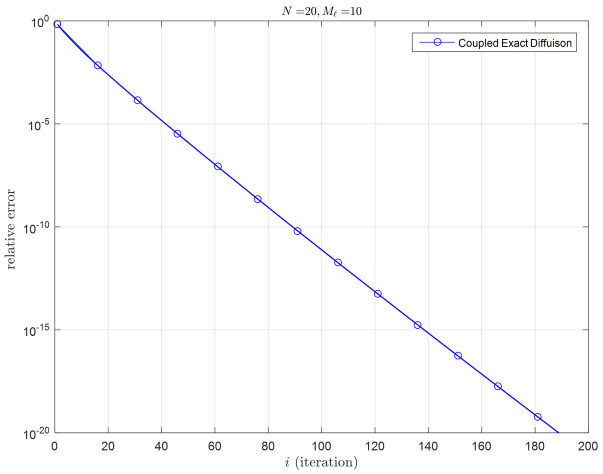

Fig. 5: Relative error.

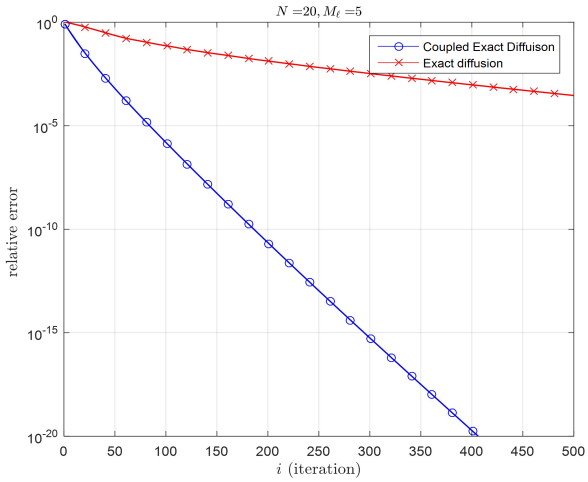

Fig. 6: Relative errors for the proposed coupled diffusion and the exact diffusion algorithm [3], [17].

IV. CONCLUSION

In this work, we solved a distributed optimization problem where each agent cost depends on multiple parameters, and agents are coupled in that they may share similar parameters. The proposed coupled exact diffusion strategy enjoys a linear convergence rate for strongly-convex cost functions and extends the exact diffusion approach of [3], [17] to the case of partially-coupled agent behavior.

V. REFERENCES

[1] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion strategy for optimization by networked agents," in *Proc. EUSIPCO*, Kos, Greece, Aug.–Sep. 2017.

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

[3] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion for distributed optimization and learning-Part I: Algorithm development," submitted for publication. *Available on arXiv:1702.05122*, Feb. 2017.

[4] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "DQM: Decentralized quadratically approximated alternating direction method of multipliers," *IEEE Trans. Signal Process.*, vol. 64, no. 19, pp. 5158–5173, 2016.

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

[6] 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.

[7] O. Chapelle, P. Shivaswmy, K. Q. Vadrevu, S. Weinberger, Y. Zhang, and B. Tseng, "Multi-task learning for boosting with applications to web search ranking," in *Proc. ACM SIGKDD*, Washington, DC, USA, Jul., 2010, pp. 1189–1198.

[8] R. Halvgaard, L. Vandenberghe, N. K. Poulsen, H. Madsen, and J. B. Jorgensen, "Distributed model predictive control for smart energy systems," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1675–1682, April 2016.

[9] P. D. Christofides, R. Scattolini, D. M. de la Pena, and J. Liu, "Distributed model predictive control: A tutorial review and future research directions," *Computers and Chemical Engineering*, vol. 51, pp. 21–41, April 2013.

[10] J. Plata-Chaves, A. Bertrand, and M. Moonen, "Incremental multiple error filtered-X LMS for node-specific active noise control over wireless acoustic sensor networks," *in* Proc. *IEEE Sensor Array and Multichannel Signal Processing Workshop*, pp. 1–5, July 2016, Rio de Janeiro, Brazil.

[11] F. Cattivelli and A. H. Sayed, "Distributed nonlinear Kalman filtering with applications to wireless localization," in *Proc. IEEE ICASSP*, Dallas, TX, Mar., 2010, pp. 3522–3525.

[12] V. Kekatos and G. B. Giannakis, "Distributed robust power system state estimation," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1617–1626, May 2013.

[13] J. Mota, J. Xavier, P. Aguiar, and M. Puschel, "Distributed optimization with local domains: Application in MPC and network flows," *IEEE Trans. Autom. Contr.*, vol. 60, no. 7, pp. 2004–2009, July 2015.

[14] T. H. Summers and J. Lygeros, "Distributed model predictive consensus via the alternating directoin method of multipliers," in Proc. *Allerton Confrence on Communication, Control, and Computing*, Monticello, IL, USA, 2012, pp. 79–84.

[15] Y. Pu, M. N. Zeilinger, and C. N. Jones, "Inexact fast alternating minimization algorithm for distributed model predictive control," in Proc. *IEEE Conference on Decision and Control (CDC)*, Los Angeles, CA, USA, 2014, pp. 5915–5921.

[16] M. Schmidt, N. L. Roux, and F. Bach, "Convergence rates of inexact proximal-gradient methods for convex optimization," *in* Proc. *Conference on Neural Information Processing* Systems *(NIPS)*, pp. 6819–6824, Granada, SPAIN, 2011.

[17] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion for distributed optimization and learning-Part II: Convergence analysis," submitted for publication. *Available on arXiv:1702.05142*, Feb. 2017.