Dual Coupled Diffusion for Distributed Optimization with Affine Constraints

S. A. Alghunaim*, K. Yuan*, and A. H. Sayed†, *Fellow*, IEEE

***Abstract*— In this work, a distributed multi-agent optimization problem is studied where different subsets of agents are coupled with each other through affine constraints. Moreover, each agent is only aware of its own contribution to the constraints and only knows which neighboring agents share constraints with it. An effective distributed first-order algorithm is developed, which requires sharing dual variables only and takes advantage of the constraint sparsity. The algorithm is shown to converge to the exact minimizer under sufficiently small constant step sizes. A simulation is given to illustrate the effect of the constraint structure and advantages of the proposed algorithm.**

I. Introduction

This work considers a distributed optimization problem in which agents are coupled through multiple affine equality constraints. Many previous works such as [1]–[6] considered optimization problems where a network of agents are coupled through separable constraints of the form:

$$\begin{aligned} \underset{w_1,w_2,\cdots,w_K}{\text{minimize}} \quad & \sum_{k=1}^{K} J_k(w_k) \\ \text{subject to} \quad & \sum_{k=1}^{K} B_k w_k = b, \end{aligned} \tag{1}$$

where $J_k(.)$: $\mathbb{R}^{Q_k} \to \mathbb{R}$, $w_k \in \mathbb{R}^{Q_k}$, $B_k \in \mathbb{R}^{S\times Q_k}$, and $b \in \mathbb{R}^S$. In this formulation, it is assumed that all agents are involved in the constraint, while each agent is assumed to have knowledge of B_k and b only. Problems of the form (1) find applications in many areas including network utility maximization (NUM) [7] and smart grids [8], [9]. They have also been considered widely in the literature and many useful distributed algorithms have been developed for their solution — see [1]–[6] and the references therein. Different from problem (1), in which *all* agents are involved in a single affine constraint, this paper considers a more general scenario, which allows the possibility that different subsets of agents may be involved in different affine constraints. Before, formalizing the problem we introduce the notation used in this work.

Notation. We write $\text{col}\{x_j\}_{j=1}^N$ to denote a column vector formed by stacking $x_1, ..., x_N$ on top of each other and blkdiag$\{X_j\}_{j=1}^N$ to denote a block diagonal matrix consisting of diagonal blocks $\{X_j\}$. We let blkrow$\{X_j\}_{j=1}^N = [X_1 \ \cdots \ X_N]$. For a set $\mathcal{X} = \{m_1, m_2, \cdots, m_x\}$, we let $U = [g_{mn}]_{m,n\in\mathcal{X}}$ denote a matrix with $(i,j)-$th entry equal to g_{m_i,m_j}.

This work was supported in part by NSF grants CCF-1524250 and ECCS-1407712.

*S. A. Alghunaim and K. Yuan are with the Electrical and Computer Engineering Department, University of California at Los Angeles (UCLA), CA 90095. Emails:{salghunaim,kunyuan}@ucla.edu.

†A. H. Sayed is with the Ecole Polytechnique Federale de Lausanne EPFL, School of Engineering, CH-1015 Lausanne, Switzerland e-mail: ali.sayed@epfl.ch.

A. Problem Formulation

Consider a network with K agents and assume it is divided into smaller E overlapping *sub-networks*. For each sub-network e, we let $\mathcal{C}_e$ denote all agents in this sub-network. Each sub-network e is associated with an affine constraint that involves all agents belonging to $\mathcal{C}_e$. Specifically, we formulate the following optimization problem:

$$\begin{aligned} \underset{w_1,w_2,\cdots,w_K}{\text{minimize}} \quad & \mathcal{J}(\mathcal{w}) \triangleq \sum_{k=1}^{K} J_k(w_k) \\ \text{subject to} \quad & \sum_{k\in\mathcal{C}_e} B_{e,k} w_k = b_e, \quad \forall\ e = 1, \cdots, E, \end{aligned} \tag{2}$$

where $\mathcal{w} = \text{col}\{w_k\}_{k=1}^K$, $B_{e,k} \in \mathbb{R}^{S_e\times Q_k}$ and $b_e \in \mathbb{R}^{S_e}$. An illustration of problem (2) is shown in Fig. 1. Note that problem (1) assumes that all agents are involved in the constraint, while problem (2) allows different subsets of agents to be involved in different constraints. For the case $E = 1$ and $\mathcal{C}_1 = \{1, \cdots, K\}$, we recover problem (1).

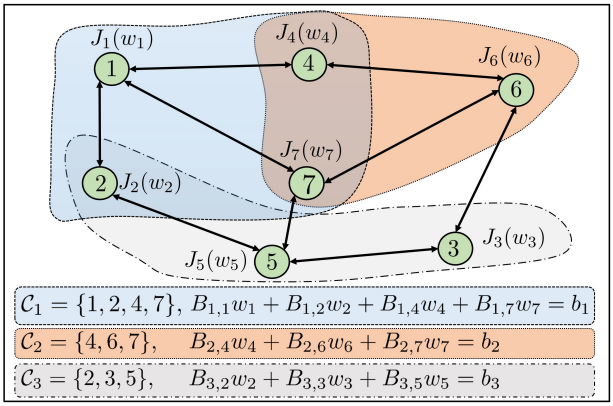


Fig. 1: A connected network of agents where different colors highlight different sub-networks.

Assumption 1. *(**Cost Function**): It is assumed that the cost function, $\mathcal{J}(\mathcal{w}) = \sum_{k=1}^K J_k(w_k)$, is a convex differentiable function with Lipschitz continuous gradient:*

$$\|\nabla\mathcal{J}(\mathcal{w}) - \nabla\mathcal{J}(z)\| \le \delta\|\mathcal{w} - z\| \tag{3}$$

Moreover, $\mathcal{J}(\mathcal{w})$ is strongly convex:

$$(\mathcal{w} - z)^{\mathsf{T}}\nabla\mathcal{J}(\mathcal{w}) \ge \mathcal{J}(\mathcal{w}) - \mathcal{J}(z) + \frac{\nu}{2}\|\mathcal{w} - z\|^2 \tag{4}$$

where $\{\delta, \nu\}$ are strictly positive scalars and $\delta > \nu$. ■

Assumption 2. *(**Connected Sub-networks**): The network graph is undirected and each sub-network $\mathcal{C}_e$ is connected.*

■

Assumption 2 can be satisfied for any connected network – see [10] for details. Moreover, in many situations, the set $\mathcal{C}_e$ involves the neighborhood (or a subset of the neighborhood) of each agent only. In that case, each neighborhood is connected and Assumption 2 is automatically satisfied. Examples of applications where these formulations and conditions are satisfied include network flow optimization [11], optimal power flow [12], [13], multitask problems [14], and distributed model predictive control [15]. We list one example next.

Example 1. (Distributed Model Predictive Control) We examine a distributed finite-horizon control problem, which is a special case of problem (2) [15]. Thus, consider $E = K$ subsystems. Given initial states $\{x_{e,0}\}$, each subsystem evolves over $t \geq 0$ according to the dynamics:

$$x_{e,t+1} = F_e x_{e,t} + G_{ee} u_{e,t} + \sum_{k \in \mathcal{N}_e'} \big(F_{ek} x_{k,t} + G_{ek} u_{k,t}\big) \quad (5)$$

where the matrices in (5) are of appropriate dimensions, $u_{e,t}$ is the input of subsystem e at time t, and $\mathcal{N}_e'$ denotes the neighborhood of agent e, excluding agent e. If we introduce the T block finite horizon vectors:

$$x_e \triangleq \mathrm{col}\{x_{e,t}\}_{t=1}^{T}, \quad u_e \triangleq \mathrm{col}\{u_{e,t}\}_{t=0}^{T-1} \quad (6)$$

then, by iterating (5), it can be verified that [16]:

$$x_e = G_{ee}' u_e + \sum_{k \in \mathcal{N}_e'} (F_{ek}' x_k + G_{ek}' u_k) + b_e \quad (7)$$

for matrices $\{G_{ek}', F_{ek}'\}$ constructed from $\{F_e, F_{ek}, G_{ek}\}$ and some vector b_e constructed from F_e, F_{ek} and the initial condition $x_{e,0}$. If we let $w_e \triangleq \mathrm{col}\{x_e, u_e\}$, and introduce:

$$B_{e,e} \triangleq \mathrm{blkrow}\{I, -G_{ee}'\}, \quad B_{e,k} \triangleq -\mathrm{blkrow}\{F_{ek}', G_{ek}'\},$$

then we can formulate the following problem [17]:

$$\begin{aligned} \underset{w_1, w_2, \cdots, w_K}{\text{minimize}} \quad & \sum_{k=1}^{K} \big(w_k^{\mathsf{T}} R_k w_k + r_k^{\mathsf{T}} w_k\big) \quad (8) \\ \text{subject to} \quad & \sum_{k \in \mathcal{N}_e} B_{e,k} w_k = b_e, \quad \forall\ e = 1, \cdots, K \end{aligned}$$

where $\mathcal{N}_e$ denotes the neighborhood of agent e, including agent e. In this example, the number of constraints is equal to the number of agents ($E = K$) and each neighborhood is involved in an affine constraint, i.e, $\mathcal{C}_e = \mathcal{N}_e$. ■

B. Related Work and Main Contributions

In this paragraph, we mention some related works that considered a problem similar to (2) and highlight the differences. As mentioned earlier, most previous works [1]–[6] focus on constraints of the type (1), but some instances of (2) have been considered. For example, in distributed control formulations [16], [17], the constraints are usually limited to the neighborhoods of each agent (i.e, $E = K$ and $\mathcal{C}_e = \mathcal{N}_e$) as in problem (8). In [16] and [17], distributed algorithms are developed to solve (8) including other constraints. Similarly, in optimal power flow [12], [13], $E = K$ and the set $\mathcal{C}_e$ also involves the neighborhood only, and, moreover, the matrices $\{B_{e,k}\}$ are identity matrices. In both control and power flow formulations, it assumed that the number of constraints is equal to the number of agents ($E = K$), and each constraint k involves only the direct neighbors of agent k; thus, it can be solely handled by agent k by receiving primal estimates from its neighbors. In contrast, in this work, the number of constraints and agents are not necessarily equal and the set $\mathcal{C}_e$ can include any arbitrary connected subset of agents. Moreover, agent k is only aware of the quantities $B_{e,k}$ and b_e if $k \in \mathcal{C}_e$. The algorithm developed in this work shares only dual variables and no primal information is shared.

In network utility maximization problems, a similar formulation appears but the matrices $\{B_{e,k}\}$ are identity matrices, albeit with a different distributed framework; it is assumed that the agents (called sources) in $\mathcal{C}_e$ are connected through one link that handles the constraint coupling these sources – see [7] and references therein. In [14], a multitask problem is considered for a quadratic stochastic optimization problem and the set $\mathcal{C}_e$ is limited to only a subset of the neighborhood of some agent with the assumption that all agents in that subset are directly connected, i.e., $\mathcal{C}_e \subseteq \mathcal{N}_k$ for all $k \in \mathcal{C}_e$. This assumption was then relaxed in [18] to handle constraints of the form (2), however, it is assumed that agent $k \in \mathcal{C}_e$ knows the global matrices $\{B_{e,s}\}$ for all $s \in \mathcal{C}_e$ and it can receive delayed estimates of w_s for all $s \in \mathcal{C}_e$ using a multi-hop relay protocol across the agents in $\mathcal{C}_e$. In this work, each agent k is only aware of the local matrices $\{B_{e,k}\}$ multiplying its variable w_k, and thus, is only aware of its own part for each constraint. Finally, in [19] a different consensus framework is considered, where each agent has a cost $J_k(w_k)$ and local constraints; the variables $\{w_k\}$ are coupled through sharing different subsets of entries across different subsets of agents.

Traditionally, problems with a coupling affine constraint of the form shown in (1) are tackled by using dual decomposition methods [1]–[6]. For each constraint that agent k is involved in, it has to maintain a dual variable associated with that constraint. For problem (1), this means that each agent will be involved in all constraints. By doing so, each agent will maintain a long dual vector to reflect *all* constraints, and *all* agents in the network will have to reach consensus on a longer dual vector than necessary. In contrast, if the sub-global coupled structure is considered, and the optimization problem is instead formulated in the form (2), then each agent will only need to maintain dual variables for the *related*, and not *all* constraints. Thus, only the agents involved in one particular constraint will need to agree on the associated dual variable. This sub-global coupled structure helps reduce communications and computations within the network. Therefore, for large networks with sparse constraints, in the sense that each agent is only involved in a few constraints, it is more effective to design an algorithm that directly solves (2) – see Section IV.

Given the above, the two main contributions of this work are: (a) designing a novel first-order algorithm that can be tailored to exploit the structure of the constraints with guaranteed convergence under sufficiently small constant step-sizes; and (b) highlighting the importance and effect of the sparsity in the constraints on the performance of the designed algorithm.

II. Algorithm Development

We start by introducing the Lagrangian function:

$$\mathcal{L}(\mathcal{W},y)=\sum_{k=1}^{K}J_k(w_k)+\sum_{e=1}^{E}(y^e)^{\mathsf{T}}\bigg(\sum_{k\in\mathcal{C}_e}B_{e,k}w_k-b_e\bigg) \tag{9}$$

where $y=\mathrm{col}\{y^e\}_{e=1}^{E}$ and $y^e\in\mathbb{R}^{S_e}$ denotes the dual variable associated with the e-th constraint. We will rewrite the Lagrangian (9) as a sum of local functions. To do that, we let $\mathcal{E}_k$ denote the set of equalities that agent k is involved in. For example, in Fig. 1 agent 2 is involved in equalities one and three, thus, $\mathcal{E}_2=\{1,3\}$. From the definition of $\mathcal{E}_k$, we have $\mathcal{C}_e=\{k\mid e\in\mathcal{E}_k\}$. Using this notation, the second term on the right hand side of (9) can be rewritten as follows:

$$\begin{aligned}&\sum_{e=1}^{E}(y^e)^{\mathsf{T}}\left(\sum_{k\in\mathcal{C}_e}B_{e,k}w_k-b_e\right)\\&\overset{(a)}{=}\sum_{e=1}^{E}\sum_{k\in\mathcal{C}_e}(y^e)^{\mathsf{T}}\left(B_{e,k}w_k-\frac{1}{N_e}b_e\right)\\&\overset{(b)}{=}\sum_{k=1}^{K}\sum_{e\in\mathcal{E}_k}(y^e)^{\mathsf{T}}\left(B_{e,k}w_k-\frac{1}{N_e}b_e\right)\end{aligned} \tag{10}$$

where N_e denotes the cardinality of the set $\mathcal{C}_e$ and in step (a) we included b_e inside the sum $\sum_{k\in\mathcal{C}_e}$. Step (b) holds since $k\in\mathcal{C}_e\iff e\in\mathcal{E}_k$. Therefore, if we introduce the vector y_k that collects the dual variables $\{y^e\}$ if agent k is involved in equality e, namely,

$$y_k\triangleq\mathrm{col}\{y^e\}_{e\in\mathcal{E}_k}, \tag{11}$$

then using (10) we can rewrite (9) as:

$$\mathcal{L}(\mathcal{W},y)=\sum_{k=1}^{K}L_k(w_k,y_k) \tag{12}$$

where

$$L_k(w_k,y_k)\triangleq J_k(w_k)+\sum_{e\in\mathcal{E}_k}(y^e)^{\mathsf{T}}\bigg(B_{e,k}w_k-\frac{1}{N_e}b_e\bigg) \tag{13}$$

From (12) we see that the Lagrangian is written as a sum of separable local terms $L_k(w_k,y_k)$ defined in (13). Moreover, different agents may share different subsets of $\{y^e\}$. We now are interested in the saddle point problem:

$$\max_{y}\min_{w}\mathcal{L}(\mathcal{W},y)=\max_{y}\left(\min_{\mathcal{W}}\sum_{k=1}^{K}L_k(w_k,y_k)\right) \tag{14}$$

Assumption 3. *A solution (i.e., saddle point) exists for* (14) *and strong duality holds.* ■

The optimal primal and dual solutions of (14) are denoted by $\mathcal{W}^\star=\mathrm{col}\{w_k^\star\}_{k=1}^{K}$ and $y^\star=\mathrm{col}\{y^{e,\star}\}_{e=1}^{E}$.

A. Coupled Exact Diffusion

In our previous work [10], we proposed a first-order distributed algorithm for solving an optimization problem where different agents share different subsets of variables. We briefly review it here since that algorithm will be critical for the solution of the problem under consideration.

Let $\{z^1,z^2,\cdots,z^E\}$ denote E variables where $z^e\in\mathbb{R}^{S_e}$. Consider an optimization problem of the form:

$$\underset{z^1,z^2,\cdots,z^E}{\text{minimize}}\quad\sum_{k=1}^{K}f_k(z_k),\quad z_k\triangleq\mathrm{col}\{z^e\}_{e\in\mathcal{E}_k} \tag{15}$$

where the variables $\{z_k\}$ are of similar structure to (11), i.e., different agents may share different subsets of $\{z^e\}$. Recall that $\mathcal{C}_e$ denotes the sub-network of nodes such that $e\in\mathcal{E}_k$. If we introduce the combination coefficients:

$$\sum_{s\in\mathcal{C}_e}a_{e,sk}=1,\quad\sum_{k\in\mathcal{C}_e}a_{e,ks}=1 \tag{16}$$

$$a_{e,sk}>0,\quad a_{e,sk}=0\text{ for }s\notin\mathcal{N}_k\cap\mathcal{C}_e \tag{17}$$

then problem (15) can be solved by using the following coupled diffusion algorithm derived in [10]. Set $z_{k,-1}^e=\psi_{k,-1}^e$ to arbitrary values. For each k repeat for $i\geq0$:

$$\psi_{k,i}^e=z_{k,i-1}^e-\mu_y\nabla_{z^e}f(z_{k,i-1}),\qquad\forall\ e\in\mathcal{E}_k \tag{18}$$

$$\phi_{k,i}^e=\psi_{k,i}^e+z_{k,i-1}^e-\psi_{k,i-1}^e,\qquad\forall\ e\in\mathcal{E}_k \tag{19}$$

$$z_{k,i}^e=\sum_{s\in\mathcal{N}_k\cap\mathcal{C}_e}\bar{a}_{e,sk}\phi_{s,i}^e,\qquad\forall\ e\in\mathcal{E}_k \tag{20}$$

where $z_{k,i}^e$ is the estimate of z^e at agent k, $\{\psi_{k,i}^e,\phi_{k,i}^e\}$ are intermediate vectors used to estimate $z_{k,i}^e$, and μ_y is a positive step-size parameter. The coefficients $\{\bar{a}_{e,sk}\}$ are defined as follows:

$$\bar{a}_{e,sk}\triangleq\begin{cases}0.5\ (1+a_{e,kk}), & \text{if } s=k\\ 0.5\ a_{e,sk}, & \text{otherwise}\end{cases} \tag{21}$$

B. Dual Coupled Diffusion

The previous algorithm can also be utilized to solve the saddle point problem (14). In this saddle point problem, the variable w_k is a "local variable" at agent k only since it is present only in $L_k(w_k,y_k)$. However, from (11), the functions $\{L_k(w_k,y_k)\}$ are coupled across the agents because different agents may share different subsets of $\{y^e\}$; therefore, y_k is a "global variable". Now, given $\mathcal{W}^\star$, problem (14) becomes of similar form to (15). To estimate $\mathcal{W}^\star$, we employ a primal-descent (with step-size $\mu_w>0$) and to estimate $y^\star$ we employ dual-ascent using the coupled diffusion algorithm (with step-size $\mu_y>0$) to arrive at algorithm (22) listed below. In this listing, $y_{k,i}^e$ denotes the estimate of y^e at time i for agent $k\in\mathcal{C}_e$. Steps (22a)–(22b) are local steps that do not require any communication between neighbors. Step (22d) is a combination step that requires agent k involved in the constraint e to combine its dual estimate with only the neighbors involved in the same constraint.

Algorithm (Dual Coupled Diffusion)

Setting: Let $w_{k,-1}$ and $y^e_{k,-1} = \psi^e_{k,-1}$ arbitrary.

For every agent k, at iteration $i \geq 0$ do:

$$w_{k,i} = w_{k,i-1} - \mu_w \nabla J_k(w_{k,i-1}) - \mu_w \sum_{e \in \mathcal{E}_k} B^{\mathsf{T}}_{e,k} y^e_{k,i-1} \tag{22a}$$

For all $e \in \mathcal{E}_k$:

$$\psi^e_{k,i} = y^e_{k,i-1} + \mu_y \left(B_{e,k} w_{k,i} - \frac{1}{N_e} b_e \right) \tag{22b}$$

$$\phi^e_{k,i} = \psi^e_{k,i} + y^e_{k,i-1} - \psi^e_{k,i-1} \tag{22c}$$

$$y^e_{k,i} = \sum_{s \in \mathcal{N}_k \cap \mathcal{C}_e} \bar{a}_{e,sk} \phi^e_{s,i} \tag{22d}$$

III. Main Results

We start by introducing the $N_e \times N_e$ matrix A_e that collects the coefficients $\{a_{e,sk}\}$:

$$A_e \triangleq [a_{e,sk}]_{s,k \in \mathcal{C}_e} \tag{23}$$

Assumption 4. *(**Combination matrices**): The combinations matrices $\{A_e\}$ are assumed to be primitive, symmetric, and doubly stochastic.* ■

Under Assumption 2, the previous assumption can be easily satisfied and many rules exist to construct such weights – see [20], [21]. In-order to analyze (22), we will rewrite it in a compact network form. To do that, we introduce the following sub-network quantities:

$$\mathcal{Y}^e_i \triangleq \mathrm{col}\{y^e_{k,i}\}_{k \in \mathcal{C}_e} \in \mathbb{R}^{N_e S_e}, \quad \bar{\mathcal{A}}_e \triangleq \bar{A}_e \otimes I_{S_e} \tag{24}$$

where $\bar{A}_e = \frac{1}{2}(I_{N_e} + A_e)$. We also introduce the network quantities:

$$\mathcal{W}_i \triangleq \mathrm{col}\{w_{k,i}\}_{k=1}^K \tag{25}$$

$$\nabla \mathcal{J}(\mathcal{W}_i) \triangleq \mathrm{col}\{\nabla J_k(w_{k,i})\}_{k=1}^K \tag{26}$$

$$\mathcal{Y}_i \triangleq \mathrm{col}\{\mathcal{Y}^e_i\}_{e=1}^E \tag{27}$$

$$\bar{\mathcal{A}} \triangleq \mathrm{blkdiag}\{\bar{\mathcal{A}}_e\}_{e=1}^E \tag{28}$$

$$b \triangleq \mathrm{col}\left\{ \frac{1}{N_e} (\mathbb{1}_{N_e} \otimes b_e) \right\}_{e=1}^E \tag{29}$$

To write (22) in network form, we need to rewrite the term $\sum_{e \in \mathcal{E}_k} B^{\mathsf{T}}_{e,k} y^e_{k,i-1}$ in terms of the network quantity $\mathcal{Y}_{i-1}$ defined in (27). This can be done as follows:

$$\sum_{e \in \mathcal{E}_k} B^{\mathsf{T}}_{e,k} y^e_{k,i-1} = \sum_{e \in \mathcal{E}_k} \mathcal{B}^{\mathsf{T}}_{ek} \mathcal{Y}^e_{i-1} = \sum_{e=1}^E \mathcal{B}^{\mathsf{T}}_{ek} \mathcal{Y}^e_{i-1} \tag{30}$$

where we introduced the $1 \times N_e$ block row matrix $\mathcal{B}^{\mathsf{T}}_{ek}$ of similar block structure as $\mathcal{Y}^e_{i-1}$ that picks $y^e_{k,i-1}$ if $e \in \mathcal{E}_k$ such that $\mathcal{B}^{\mathsf{T}}_{ek} \mathcal{Y}^e_{i-1} = B^{\mathsf{T}}_{e,k} y^e_{k,i-1}$ and $\mathcal{B}^{\mathsf{T}}_{ek} \mathcal{Y}^e_{i-1} = 0_{Q_k}$ if $e \notin \mathcal{E}_k$. This can be represented mathematically by:

$$\mathcal{B}^{\mathsf{T}}_{ek} = \mathrm{blkrow}\{B^{\mathsf{T}}_{e,kk'}\}_{k' \in \mathcal{C}_e} \tag{31}$$

$$B^{\mathsf{T}}_{e,kk'} \triangleq \begin{cases} B^{\mathsf{T}}_{e,k}, & \text{if } k \in \mathcal{C}_e \text{ , } k = k' \\ 0_{Q_k, S_e}, & \text{otherwise} \end{cases} \tag{32}$$

Therefore, if we let:

$$\mathcal{B}^{\mathsf{T}} \triangleq \begin{bmatrix} \mathcal{B}^{\mathsf{T}}_{11} & \cdots & \mathcal{B}^{\mathsf{T}}_{E1} \\ \vdots & & \vdots \\ \mathcal{B}^{\mathsf{T}}_{1K} & \cdots & \mathcal{B}^{\mathsf{T}}_{EK} \end{bmatrix} \tag{33}$$

then algorithm (22) can be written compactly as follows:

$$\mathcal{W}_i = \mathcal{W}_{i-1} - \mu_w \nabla \mathcal{J}(\mathcal{W}_{i-1}) - \mu_w \mathcal{B}^{\mathsf{T}} \mathcal{Y}_{i-1} \tag{34}$$

$$\mathcal{Y}_i = \bar{\mathcal{A}} \Big(2\mathcal{Y}_{i-1} - \mathcal{Y}_{i-2} + \mu_y \mathcal{B} (\mathcal{W}_i - \mathcal{W}_{i-1}) \Big) \tag{35}$$

for $i \geq 1$ with initialization:

$$\mathcal{Y}_0 = \mathcal{Y}_{-1} + \mu_y (\mathcal{B} \mathcal{W}_0 - b) \tag{36}$$

It can be shown that the second step (35) can be rewritten in an equivalent form. Let:

$$\mathcal{A} = \mathrm{blkdiag}\{A_e \otimes I_{S_e}\}_{e=1}^E \tag{37}$$

and introduce the eigenvalue decomposition [22]:

$$0.5(I_N - \mathcal{A}) = \begin{bmatrix} \mathcal{U}_1 & \mathcal{U}_2 \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathcal{U}_1^{\mathsf{T}} \\ \mathcal{U}_2^{\mathsf{T}} \end{bmatrix} = \mathcal{U}_1 \Sigma \mathcal{U}_1^{\mathsf{T}} \tag{38}$$

where $N = \sum_{e=1}^E N_e S_e$, $\mathcal{U}_1 \in \mathbb{R}^{N \times r}$, $\mathcal{U}_2 \in \mathbb{R}^{N \times (N-r)}$, and $\Sigma = \mathrm{diag}\{\lambda_j\}_{j=1}^r$ with $\lambda_1 \geq \cdots \geq \lambda_r > 0$ denoting the non-negative eigenvalues of the matrix $0.5(I - \mathcal{A})$ [20]. From Assumption 4 it is easy to check that those eigenvalues are strictly less the one. Using the decomposition (38), we can rewrite (35) equivalently as follows [23]:

$$\mathcal{W}_i = \mathcal{W}_{i-1} - \mu_w \nabla \mathcal{J}(\mathcal{W}_{i-1}) - \mu_w \mathcal{B}^{\mathsf{T}} \mathcal{Y}_{i-1} \tag{39}$$

$$x_i = x_{i-1} - \frac{1}{\mu_y} \mathcal{U}_1^{\mathsf{T}} \big(\mathcal{Y}_{i-1} + \mu_y (\mathcal{B} \mathcal{W}_i - b) + \mu_y \mathcal{U}_1 \Sigma x_{i-1} \big) \tag{40}$$

$$\mathcal{Y}_i = \mathcal{Y}_{i-1} + \mu_y \big(\mathcal{B} \mathcal{W}_i - b \big) + \mu_y \mathcal{U}_1 \Sigma x_i \tag{41}$$

for $i \geq 1$, where we introduced a new sequence x_i with $x_0 = 0$. Intuitively, step (41) can be regarded as a corrected gradient ascent step.

We now state the Lemmas that will be used in the analysis. The following auxiliary result is proven in [24].

Lemma 1. *For any $S \times S$ primitive, symmetric and doubly stochastic matrix A, it holds that $I_S - A$ is symmetric and positive semi-definite. If we let $\mathcal{A} = A \otimes I_M$, then, for any block vector $\mathcal{Z} = \mathrm{col}\{z^1, ..., z^S\}$ in the nullspace of $I - \mathcal{A}$ with entries $z^s \in \mathbb{R}^M$ it holds that:*

$$(I - \mathcal{A}) \mathcal{Z} = 0 \iff z^1 = z^2 = ... = z^S \tag{42}$$

■

Condition (42) in Lemma 1 will be used to ensure consensus among the dual variables across agents in the proof of the following Lemma regarding the optimality condition.

Lemma 2. *(**Optimality condition**) If there exists a point $(\mathcal{W}^\star, \mathcal{Y}^\star, x^\star)$ such that:*

$$\nabla \mathcal{J}(\mathcal{W}^\star) + \mathcal{B}^{\mathsf{T}} \mathcal{Y}^\star = 0 \tag{43}$$

$$\mathcal{U}_1^{\mathsf{T}} \mathcal{Y}^\star = 0 \tag{44}$$

$$(\mathcal{B} \mathcal{W}^\star - b) + \mathcal{U}_1 \Sigma x^\star = 0 \tag{45}$$

Then, it holds that:

$$y_k^{e,\star} = y^{e,\star}, \quad k \in \mathcal{C}_e \tag{46}$$

where $(\mathcal{W}^\star, y^{1,\star}, \cdots, y^{E,\star})$ *is a saddle point for the Lagrangian* (9)*.*

Proof: Since $\mathcal{U}_1^{\mathsf{T}}\mathcal{U}_1 = I$ and $\Sigma > 0$, condition (44) is equivalent to:

$$\mathcal{U}_1^{\mathsf{T}}\mathcal{Y}^\star = 0 \iff \mathcal{U}_1\Sigma\mathcal{U}_1^{\mathsf{T}}\mathcal{Y}^\star = 0 \iff \frac{1}{2}(I-\mathcal{A})\mathcal{Y}^\star = 0 \tag{47}$$

Therefore, from (42), and the block structure of $\mathcal{A}$ in (37), condition (44) gives:

$$y_k^{e,\star} = y_s^{e,\star}, \quad \forall\ k, s \in \mathcal{C}_e \tag{48}$$

Using the block structure of $\nabla\mathcal{J}(.)$ and $\mathcal{B}$ in (26) and (33), we can expand (43) into its components to get:

$$\nabla J_k(w_k^\star) + \sum_{e=1}^{E}\mathcal{B}_{ek}^{\mathsf{T}}\mathcal{Y}^{e,\star} = \nabla J_w(w_k^\star) + \sum_{e\in\mathcal{E}_k} B_{e,k}^{\mathsf{T}} y_k^{e,\star} = 0 \tag{49}$$

for all k. Now, let $\mathcal{Z} = \text{blkdiag}\{\mathbb{1}_{N_e}\otimes I_{S_e}\}_{e=1}^{E}$. Multiplying equation (45) on the left by $\mathcal{Z}^{\mathsf{T}}$ gives:

$$0 = \mathcal{Z}^{\mathsf{T}}(\mathcal{B}\mathcal{W}^\star - b) + \underbrace{\mathcal{Z}^{\mathsf{T}}\mathcal{U}_1}_{=0}\Sigma x^\star \overset{(a)}{=} \mathcal{Z}^{\mathsf{T}}(\mathcal{B}\mathcal{W}^\star - b) \tag{50}$$

where step (a) holds because

$$\mathcal{Z}^{\mathsf{T}}\mathcal{U}_1 = \mathcal{Z}^{\mathsf{T}}(\mathcal{U}_1\Sigma\mathcal{U}_1^{\mathsf{T}})\mathcal{U}_1\Sigma^{-1} = 0.5\mathcal{Z}^{\mathsf{T}}(I-\mathcal{A})\mathcal{U}_1\Sigma^{-1} = 0 \tag{51}$$

where we used the fact $\mathcal{U}_1^{\mathsf{T}}\mathcal{U}_1 = I$ and the last step holds because $\mathcal{Z}$ is in the nullspace of $I-\mathcal{A}$ (see (42)). Using the block structure of $\mathcal{B}$ and b, we can expand (50) into its components to get:

$$\begin{aligned}&\sum_{k=1}^{K}\left((\mathbb{1}_{N_e}^{\mathsf{T}}\otimes I_{S_e})\mathcal{B}_{ek}w_k^\star\right) - (\mathbb{1}_{N_e}^{\mathsf{T}}\otimes I_{S_e})\frac{1}{N_e}(\mathbb{1}_{N_e}\otimes b_e)\\ &= \sum_{k=1}^{K}\left((\mathbb{1}_{N_e}^{\mathsf{T}}\otimes I_{S_e})\mathcal{B}_{ek}w_k^\star\right) - b_e = 0\end{aligned} \tag{52}$$

for all e. Note that:

$$\begin{aligned}\mathcal{B}_{ek}^{\mathsf{T}}(\mathbb{1}_{N_e}\otimes I_{S_e}) &= \text{blkrow}\{B_{e,kk'}^{\mathsf{T}}\}_{k'\in\mathcal{C}_e}(\mathbb{1}_{N_e}\otimes I_{S_e})\\ &= \sum_{k'\in\mathcal{C}_e} B_{e,kk'}^{\mathsf{T}} = \begin{cases} B_{e,k}^{\mathsf{T}}, & \text{if } k\in\mathcal{C}_e\\ 0, & \text{otherwise}\end{cases}\end{aligned} \tag{53}$$

Substituting the above conclusion into (52) gives:

$$\sum_{k\in\mathcal{C}_e} B_{e,k}w_k^\star - b_e = 0 \tag{54}$$

Equations (49) and (54) along with (48) satisfy the optimality conditions of the Lagrangian (9). ■

We will show that recursions (39)–(41) converge to points that satisfy the optimality conditions given in Lemma 2. For this purpose, we introduce the error vectors:

$$\widetilde{\mathcal{W}}_i \triangleq \mathcal{W}^\star - \mathcal{W}_i, \quad \widetilde{x}_i \triangleq x^\star - x_i \quad \widetilde{\mathcal{Y}}_i \triangleq \mathcal{Y}^\star - \mathcal{Y}_i \tag{55}$$

and the positive definite matrix:

$$\mathcal{D} \triangleq \mu_y(\Sigma - \Sigma^2) > 0 \tag{56}$$

where Σ was introduced in (38). The next Lemma bounds the difference of the consecutive primal and dual errors. The proof can be found in [23].

Lemma 3. *(***Primal-dual bound***): Suppose Assumptions 1–4 hold, then:*

$$\begin{aligned}\|\widetilde{\mathcal{W}}_i\|^2 - \|\widetilde{\mathcal{W}}_{i-1}\|^2 \le{}& (-1-2\mu_w\nu+2\mu_w\delta)\|\mathcal{W}_i-\mathcal{W}_{i-1}\|^2\\ &- 2\mu_w(\mathcal{Y}_{i-1}-\mathcal{Y}^\star)^{\mathsf{T}}\mathcal{B}(\mathcal{W}_i-\mathcal{W}^\star)\\ &- 2\mu_w\nu\left(\|\widetilde{\mathcal{W}}_{i-1}\|^2 + \|\widetilde{\mathcal{W}}_i\|^2\right)\end{aligned} \tag{57}$$

and

$$\begin{aligned}&\|\widetilde{\mathcal{Y}}_i\|^2_{\mu_y^{-1}} + \|\widetilde{x}_i\|^2_{\mathcal{D}} - \|\widetilde{\mathcal{Y}}_{i-1}\|^2_{\mu_y^{-1}} - \|\widetilde{x}_{i-1}\|^2_{\mathcal{D}}\\ &= -\|x_i - x_{i-1}\|^2_{\mathcal{D}} - \|\Sigma(x^\star - x_i)\|^2_{\mu_y}\\ &\quad + 2(\mathcal{Y}_{i-1}-\mathcal{Y}^\star)^{\mathsf{T}}\mathcal{B}(\mathcal{W}_i-\mathcal{W}^\star) + \|\mathcal{B}(\mathcal{W}_i-\mathcal{W}^\star)\|^2_{\mu_y}\end{aligned} \tag{58}$$

where $(\mathcal{W}^\star, \mathcal{Y}^\star, x^\star)$ *satisfy the optimality conditions given in Lemma 2.* ■

The bounds in the previous Lemma are basically used for convergence. The following theorem is proven in [23].

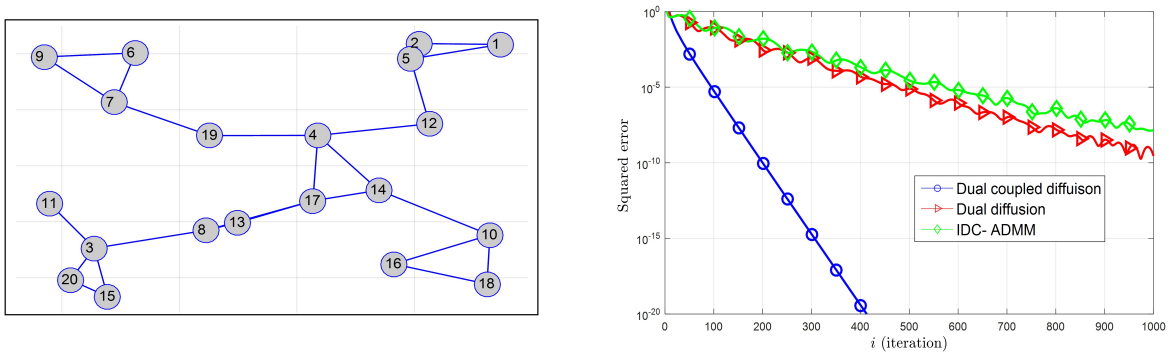
Theorem 1. *(***Convergence***): Suppose Assumptions 1–4 hold, then for positive step-sizes satisfying:*

$$\mu_w < \frac{1}{(2\delta-\nu)}, \quad \mu_y < \frac{\nu}{\lambda_{\max}(\mathcal{B}^{\mathsf{T}}\mathcal{B})} \tag{59}$$

recursion (39)–(41) *converges and it holds that* $\mathcal{W}_i$ *converges to the optimal solution of* (2)*.* ■

IV. Numerical Simulation

In this section, we illustrate the performance of our algorithm for problem (8) given in Example 1. In our simulation, we consider a randomly generated network with $K = 20$ agents shown in Figure 2a, where neighbors are decided by closeness in distance. Using the problem settings in (8), we randomly generate $R_k \in \mathbb{R}^{Q_k\times Q_k}$ and $b_k \in \mathbb{R}^{Q_k}$ ($Q_k = 10$) while making sure the matrix R_k is positive-definite and well conditioned (i.e., the difference between maximum and minimum singular values is not very large). Each vector $r_k \in \mathbb{R}^{Q_k}$ is randomly generated with each element uniformly chosen from $(-2,-1,0,1,2)$. Similarly, each matrix $B_{e,k} \in \mathbb{R}^{S_e\times Q_k}$ (with $S_e = 1$) is a randomly generated row vector with elements uniformly chosen from $(-2,-1,0,2-2)$. Each b_e is a scalar uniformly chosen from $(-1,0,1)$. The matrices $\{A_e\}$ are generated using the Metropolis rule [21]. We consider two approaches to solve problem (8). The first approach is to use the dual coupled diffusion (22) while considering the structure of the problem (8), i.e., run (22) with $E = K$, $\mathcal{C}_e = \mathcal{N}_e$. The second approach is to ignore the special structure of the problem and reformulate it into the form of problem (1); in this case we can also apply the dual coupled diffusion (22) with $E = 1, \mathcal{C}_1 = \{1,\cdots,K\}$, which we call dual diffusion. To compare with other works, we simulate the inexact distributed consensus ADMM (IDC-ADMM) from



(a) **(b)**

Fig. 2: (a) The network topology used in the simulations. **(b)** Squared error evolution over time between the different algorithms explained in the simulation section.

[1] designed for problem (1). The step-sizes are manually set to get the best possible result for each algorithm: with $(\mu_w = 0.1,\ \mu_y = 0.2)$ for the dual coupled diffusion, $(\mu_w = 0.1,\ \mu_y = 0.09)$ for the dual diffusion, and $(c = 1,\ \beta = 7)$ for IDC-ADMM [1]. Figure 2b shows the squared error $\|\mathcal{w}_i - \mathcal{w}^\star\|^2$ for each of the previous algorithms. It is observed that dual diffusion and the IDC-ADMM perform similarly. Moreover, the dual coupled diffusion outperforms both algorithms in this simulation example. One intuitive explanation is that the dual coupled diffusion takes advantage of the sparsity in the constraints and fewer consensus steps are required to reach agreement about the dual variable [23].

V. Conclusion

In this work, a distributed optimization problem is studied where coupling between different agents exists through different affine constraints. A distributed first-order algorithm is developed that converges to the minimizer for sufficiently small constant step-sizes. The sparsity of the constraints is exploited to arrive at a more effective distributed solution.

References

[1] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 482–497, Jan. 2015.

[2] T.-H. Chang, A. Nedić, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1524–1538, June 2014.

[3] T.-H. Chang, "A proximal dual consensus ADMM method for multi-agent constrained optimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3719–3734, July 2016.

[4] S. Lee and M. M. Zavlanos, "Distributed primal-dual methods for online constrained optimization," in Proc. *American Control Conference (ACC)*, Boston, MA, USA, July 2016, pp. 7171–7176.

[5] I. Notarnicola and G. Notarstefano, "Constraint coupled distributed optimization: Relaxation and duality approach," *available on arXiv: 1711.09221*, Nov. 2017.

[6] A. Falsone, K. Margellos, S. Garatti, and M. Prandini, "Dual decomposition for multi-agent distributed optimization with coupling constraints," *Automatica*, vol. 84, pp. 149–158, October 2017.

[7] D. P. Palomar and M. Chiang, "Alternative distributed algorithms for network utility maximization: Framework and applications," *IEEE Transactions on Automatic Control*, vol. 52, no. 12, pp. 2254–2269, Dec. 2007.

[8] J. Rivera, C. Goebel, and H.-A. Jacobsen, "Distributed convex optimization for electric vehicle aggregators," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1852–1863, Jan. 2017.

[9] 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.

[10] S. A. Alghunaim, K. Yuan, and A. H. Sayed, "Decentralized exact coupled optimization," in *Proc. Allerton Conference on Communication, Control, and Computing*, Allerton, IL, October 2017, pp. 338–345.

[11] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, NJ, 1993.

[12] S. Kar, G. Hug, J. Mohammadi, and J. M. Moura, "Distributed state estimation and energy management in smart grids: A consensus + innovations approach," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1022–1038, 2014.

[13] J. Guo, G. Hug, and O. Tonguz, "Enabling distributed optimization in large-scale power systems," *available on arXiv: 1605.09785*, May 2016.

[14] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Diffusion LMS for multitask problems with local linear equality constraints," *IEEE Trans. Signal Process*, vol. 65, no. 19, pp. 4979 – 4993, Jun. 2017.

[15] I. Necoara, V. Nedelcu, and I. Dumitrache, "Parallel and distributed optimization methods for estimation and control in networks," *Journal of Process Control*, vol. 21, no. 5, pp. 756–766, Jun 2011.

[16] P. Giselsson, M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer, "Accelerated gradient methods and dual decomposition in distributed model predictive control," *Automatica*, vol. 49, no. 3, pp. 829–833, Mar. 2013.

[17] R. Rostami, G. Costantini, and D. Gorges, "ADMM-based distributed model predictive control: Primal and dual approaches," in *IEEE Conference on Decision and Control (CDC)*, Melbourne, Australia, 2017, pp. 6598–6603.

[18] F. Hua, R. Nassif, C. Richard, and H. Wang, "Penalty-based multitask estimation with non-local linear equality constraints," in Proc. *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Curacao, Dec. 2017, pp. 433–437.

[19] S. A. Alghunaim and A. H. Sayed, "Distributed coupled learning over adaptive networks," in *Proc. IEEE ICASSP*, Calgary, Canada, April 2018, pp. 1–5.

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

[21] ——, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing*, vol. 3, pp. 323–453, Elsevier, 2014. Also available as arXiv:1205.4220, May 2012.

[22] A. J. Laub, *Matrix Analysis For Scientists And Engineers*. SIAM, PA, USA, 2004.

[23] S. A. Alghunaim, K. Yuan, and A. H. Sayed, "A proximal diffusion strategy for multi-agent optimization with sparse affine constraints," *submitted for publication* (will be available on arXiv), 2018.

[24] 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.