Distributed Algorithms for Solving Locally Coupled Optimization Problems on Agent Networks

Jianghai Hu, Yingying Xiao and Ji Liu

Abstract— **In this paper, we study the optimization problems for a group of agents whose individual objective functions and constraints may depend on the variables of neighboring agents. Several algorithms are proposed based on operator splitting techniques that can iteratively converge to an optimal primal (or dual) solution of the optimization problems. Then, via random coordinate updates, asynchronous implementations of the algorithms are developed with low computation and communication complexity and guaranteed almost sure convergence to an optimal solution. Numerical results are presented to illustrate the proposed algorithms.**

I. INTRODUCTION

Distributed optimization problems have been studied extensively in the past decade due to their applications in a wide range of fields such as multi-agent coordination [1], networked systems [2], distributed model predictive control [3], machine learning [4], to name a few. A class of distributed optimization problems that receives the most attention is the consensus optimization problem, where a group of m agents with local variables $x_1, \ldots, x_m$ tries to minimize the (separable) objective function $f_1(x_1) + \cdots + f_m(x_m)$ while simultaneously achieving consensus $x_1 = \cdots = x_m$ through local information exchanges. Representative algorithms developed for their solutions include, e.g., subgradient-based methods [5], [6], dual-decomposition methods [7], proximal gradient descent methods [8], and general first order algorithms [9], [10], [11], to name a few.

On the other hand, practical distributed optimization problems arising in agent networks, e.g., the formation control problem [12] and the network localization problem [13], often have coupled objective functions and constraints for individual agents. In theory this general problem can still be formulated as a consensus optimization problem with each agent maintaining copies of all other agents' variables. However, doing so may result in excessive memory and communication overhead, especially when the number of agents is large. In addition, many of the methods developed for consensus optimization problems, e.g., subgradient-based methods, require carefully tuned variable step sizes to ensure convergence, which is practical in many applications.

In this paper, we study the distributed optimization problems on agent networks with coupled local convex objective functions and constraints. The couplings are represented by a directed dependency graph. Our goal is to develop distributed solution algorithms with the following properties: constant step size, general objectives and constraints, general dependency graph, guaranteed convergence to optimality, and capability of dealing with asynchrony due to network uncertainty and inhomogeneity. Towards this goal, we will utilize the operator splitting techniques [14], [15] to propose several general-purpose distributed synchronous solution algorithms, as well as their randomized versions via the random coordinate descent method [16], [17], [18]. Our focus will be on finding algorithms with less communication/computation overhead. Specifically, at any round of our proposed algorithms, an agent communicates only with those neighbors whose variables affect its objective and constraints; only a small amount of most pertinent information is being exchanged between them; and expensive operations (e.g., solving local optimization problems) are carried out only once with the rest being simple linear vector operations.

Asynchronous algorithms have been proposed for solving the special class of consensus optimization problem [16], [19], [20], with the algorithm in [16] requiring the activation of at least two agents in each round and the algorithms in [21], [18] requiring the knowledge of activation probabilities. The AD-ADMM algorithm in [22] is applicable only to the star topology (albeit with possible network delays). Our proposed asynchronous algorithms have no such limitations.

This paper is organized as follows. The problem under study is formulated in Section II. In Section III, some useful facts on averaged operators are reviewed. Three synchronous distributed solution algorithms are proposed in Section IV, and their asynchronous implementations are presented in Section V. Simulation results of the proposed algorithms on a network localization problem are given in Section VI. Finally, Section VII summarizes the paper.

II. PROBLEM FORMULATION

Consider a group of m agents indexed by $i \in [m] := \{1, \ldots, m\}$. Each agent i has a local variable $x_i \in \mathbb{R}^{n_i}$ (which could be null) as well as a local objective function[1] f_i (which could also be null) that depends on not only x_i but also its neighboring agents' variables. This dependency is modeled by a directed graph called the dependency graph $([m], \mathcal{E})$ with the vertex set $[m]$ and the edge set $\mathcal{E} \subset [m] \times [m]$ so that an edge $(j, i) \in \mathcal{E}$ indicates that the objective function f_i of agent i depends on the variable x_j

J. Hu and Y. Xiao are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA {jianghai,xiao106}@purdue.edu

J. Liu is with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA ji.liu@stonybrook.edu

[1] We assume that all the local constraints of agent i have been incorporated into f_i using convex indicator functions.

of agent j. Denote by $\mathcal{N}_i^+ := \{j \in [m] \,|\, (j,i) \in \mathcal{E}\}$ and $\mathcal{N}_i^- := \{j \in [m] \,|\, (i,j) \in \mathcal{E}\}$ the sets of in-neighbors and out-neighbors, respectively, and by $\mathcal{N}_i = \mathcal{N}_i^+ \cup \mathcal{N}_i^-$ the set of neighbors, all of agent i. Then, the objective function of agent i is $f_i(\hat{x}_i)$, where $\hat{x}_i := \left(x_i, (x_j)_{j\in\mathcal{N}_i^+}\right)$ has the dimension $N_i = n_i + \sum_{j\in\mathcal{N}_i^+} n_j$.

Assumption 1: Each $f_i : \mathbb{R}^{N_i} \to \mathbb{R} \cup \{+\infty\}$ is an extended-real-valued closed convex proper (CCP) function, i.e., f_i is lower semi-continuous, convex, and $f_i \not\equiv +\infty$.

Denote by $x := (x_i)_{i\in[m]}$ the concatenated vector of all x_i's. Then $x \in \mathbb{R}^n$ where $n = \sum_{i\in[m]} n_i$. Our objective is to solve the following problem:

$$\text{minimize} \quad f(x) := \sum_{i\in[m]} f_i(\hat{x}_i). \tag{1}$$

Assumption 2: The set of minimizers of f, $C := \{x \,|\, f(x) < \infty,\ f(x) \le f(x'),\ \forall x'\}$, is nonempty.

We list several examples of Problem (1) below. The first one will later be used to illustrate the proposed algorithms.

Example 1: Consider two agents with variables $x_1, x_2 \in \mathbb{R}$ and local objective functions $f_1(x_1, x_2) = (x_1^2 + x_2^2)/2$ and $f_2(x_2) = -x_2$. The dependency graph has only one edge $(2,1)$. Note that $f = f_1 + f_2$ has a unique minimizer $x_1^* = 0$ and $x_2^* = 1$, even though no minimizer exists for f_2.

Example 2: (L_1-regularized least square problem) To find sparse approximate solutions to the linear equation $Ax = b$, one can solve the optimization problem $\min\left(\|Ax-b\|^2 + \lambda\|x\|_1\right)$ for given $\lambda > 0$. Here, $\|\cdot\|$ and $\|\cdot\|_1$ denote the L_2 and L_1 norms, respectively. By decomposing x into $(x_i)_{i\in[m]}$ and A into block matrices $(A_{ij})_{i,j\in[m]}$, this is equivalent to Problem (1) with $f_i = \|\sum_{j\in\mathcal{N}_i^+} A_{ij}x_j - b_i\|^2 + \lambda\|x_i\|_1$. Note that $\mathcal{N}_i^+ = \{j \,|\, A_{ij} \neq 0\}$ and $\mathcal{N}_i^- = \{j \,|\, A_{ji} \neq 0\}$.

Example 3: (Convex feasibility problem) Let $f_i(\hat{x}_i) = \imath_{\mathcal{F}_i}(\hat{x}_i)$ be the convex indicator function of some convex set $\mathcal{F}_i \subset \mathbb{R}^{N_i}$, i.e., $\imath_{\mathcal{F}_i}(\hat{x}_i) = 0$ if $\hat{x}_i \in \mathcal{F}_i$ and $\imath_{\mathcal{F}_i}(\hat{x}_i) = +\infty$ if otherwise. Then Problem (1) is equivalent to finding a point x in the intersection of the sets $\{x \,|\, \hat{x}_i \in \mathcal{F}_i\}$ for $i \in [m]$.

Example 4: (Consensus optimization) Suppose for each i, $x_i \in \mathbb{R}^n$, $f_i(\hat{x}_i) = g_i(x_i) + \imath_{\{x_i = x_j,\, \forall j\in\mathcal{N}_i^+\}}$, and the dependency graph is weakly connected. Then Problem (1) is equivalent to the problem of minimizing $g_1(x) + \cdots + g_m(x)$.

Example 5: (Coordinated optimization) Suppose each agent $i \in \{2, \ldots, m\}$ has a local variable x_i and a local objective function $f_i(x_i)$, while agent 1 is a coordinator with no local variables and a non-separable objective function $f_1(x_2, \ldots, x_m)$. As examples, we can have $f_1 = \|\sum_{i=2}^m x_i\|^2$, $f_1 = \max_{i=2,\ldots,m} \|x_i\|_\infty$ where $\|\cdot\|_\infty$ denotes the L_∞-norm, or $f_1 = \imath_{\{A_2x_2 + \cdots + A_mx_m = b\}}$. In this case, the dependency graph has the edges $(2,1), (3,1), \ldots, (m,1)$.

Our goal is to design distributed iterative algorithms $x^{k+1} = T(x^k)$ for solving Problem (1) so that, at any iteration, each agent updates its variable by using only the variables of its neighbors and itself, and that the iteration result x^k converges to a solution $x^* \in C$ for all initial x^0.

Towards this goal, we first reformulate Problem (1). For each neighboring agent pair $(j,i) \in \mathcal{E}$, suppose agent i maintains an extra variable $x_{ij} \in \mathbb{R}^{n_j}$ representing its desired value for the variable x_j of its in-neighboring agent j (it is possible that $x_{ij} \neq x_j$). Denote by $\mathbf{x}_i = \left(x_i, (x_{ij})_{j\in\mathcal{N}_i^+}\right) \in \mathbb{R}^{N_i}$ the augmented variable of agent i, and let $\mathbf{x} := (\mathbf{x}_i)_{i\in[m]} \in \mathbb{R}^N$ where $N = \sum_{i\in[m]} N_i$. Then, Problem (1) is equivalent to the following optimization problem:

$$\text{minimize} \quad F(\mathbf{x}) := \sum_{i\in[m]} f_i(\mathbf{x}_i) \quad \text{subject to} \quad \mathbf{x} \in \mathcal{A}. \tag{2}$$

Here, $\mathcal{A}$ is the consensus subspace defined by

$$\mathcal{A} = \{\mathbf{x} \,|\, x_{ij} = x_j,\ \forall (j,i) \in \mathcal{E}\} \subset \mathbb{R}^N. \tag{3}$$

Problem (2) is further equivalent to

$$\text{minimize} \quad F(\mathbf{x}) + \imath_{\mathcal{A}}(\mathbf{x}). \tag{4}$$

III. AVERAGED OPERATORS

In this section, some useful facts about averaged operators will be reviewed (see [14] for more general results). For mappings (or operators) $S, T : \mathbb{R}^n \to \mathbb{R}^n$, S is called nonexpansive if $\|Sx - Sy\| \le \|x - y\|$ for all $x, y \in \mathbb{R}^n$; and T is called (α-)averaged if $T = (1-\alpha)I + \alpha S$ for some nonexpansive mapping S and $\alpha \in (0,1)$. An α-averaged operator T satisfies

$$\|Tx - Ty\|^2 \le \|x - y\|^2 - \frac{1-\alpha}{\alpha}\|(Tx - x) - (Ty - y)\|^2,$$

for all $x, y \in \mathbb{R}^n$. In particular, if $\alpha = \frac{1}{2}$, T is firmly nonexpansive: $\|Tx - Ty\|^2 \le (x-y)^\top(Tx - Ty)$, $\forall x, y$. Denote by $\mathrm{fix}(T) := \{x \,|\, Tx = x\}$ the set of fixed points of T (which could be empty). For an averaged operator T, the iteration $x^{k+1} = T(x^k)$ always converges to some $x^* \in \mathrm{fix}(T)$, provided that $\mathrm{fix}(T)$ is nonempty.

The following result, a special case of [17, Thm. 3], will be useful for asynchronous design later on.

Theorem 1: Let $T : \mathbb{R}^n \to \mathbb{R}^n$ be an α-averaged operator with $\mathrm{fix}(T) \neq \emptyset$. Partition x into $(x_1, \ldots, x_m)$ and Tx into $(T_1x, \ldots, T_mx)$ where $x_i, T_ix \in \mathbb{R}^{n_i}$ for $i \in [m]$. Consider the following iteration. At each step $k = 0, 1, \ldots$, first an index $i^k \in [m]$ is chosen randomly and independently with the probabilities $\mathbb{P}\left(i^k = j\right) = p_j \ge \varepsilon$, $j \in [m]$, for some positive ε; then x^k is updated to x^{k+1} where $x_{i^k}^{k+1} = T_{i^k}x^k$ and $x_\ell^{k+1} = x_\ell^k$ for $\ell \neq i^k$. Then, x^k converges almost surely to some $x^* \in \mathrm{fix}(T)$ as $k \to \infty$.

IV. SYNCHRONOUS ALGORITHMS

In this section, several distributed algorithms for solving Problem (2) are proposed. Recall that $\mathbf{x} = (\mathbf{x}_i)_{i\in[m]} \in \mathbb{R}^N$, where $\mathbf{x}_i = \left(x_i, (x_{ij})_{j\in\mathcal{N}_i^+}\right)$ is the local augmented variable kept by agent i. Denote by $\bar{\mathbf{x}} := \Pi_{\mathcal{A}}(\mathbf{x})$ the orthogonal projection of $\mathbf{x}$ onto the consensus subspace $\mathcal{A}$ defined in (3). Then, $\bar{\mathbf{x}} = (\bar{\mathbf{x}}_i)_{i\in[m]}$ is given by

$$\bar{x}_{ij} = \bar{x}_j = \tfrac{1}{|\mathcal{N}_j^-|+1}\left(x_j + \textstyle\sum_{\ell\in\mathcal{N}_j^-} x_{\ell j}\right), \ \ \forall (j,i) \in \mathcal{E}. \tag{5}$$

The projection of $\mathbf{x}$ onto the orthogonal complementary subspace $\mathcal{A}^\perp$ is given by $\Pi_{\mathcal{A}^\perp}(\mathbf{x}) = \mathbf{x} - \bar{\mathbf{x}}$.

A. Douglas-Rachford Algorithm

We first review some basic facts about monotone operators [14]. A set-valued operator $T : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ is called *monotone* if $(x-y)^\top(u-v) \geq 0$ for all $x, y \in \mathbb{R}^n$ and all $u \in Tx$, $v \in Ty$. For a monotone operator T, its resolvent is given by the single-valued mapping $J_T := (I+\rho T)^{-1}$ for some given $\rho > 0$. It can be shown that the reflected resolvent $2J_T - I$ is nonexpansive; hence J_T is $(1/2)$-averaged. If T is further *maximal monotone*, i.e., its graph $\{(x,u) \,|\, u \in Tx\}$ is not properly contained in the graph of any other monotone operator, then the domain of J_T is $\mathbb{R}^n$.

The subdifferential operators ∂g of a closed convex proper functions $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is maximal monotone, with its resolvent being the proximal operator of g:

$$\mathrm{prox}_{\rho g}(x) = \arg\min_z \left(g(z) + \|z-x\|^2/2\rho\right), \quad \forall x \in \mathbb{R}^n.$$

The proximal operator $\mathrm{prox}_{\rho g}$ is $(1/2)$-averaged and its fixed points, if exist, are exactly the minimizer of g. Further, if $g = \imath_C$ for a closed convex set C, then $\mathrm{prox}_{\rho g} = \Pi_C$ is the projection onto C. If $g(x) = \sum_{i\in[m]} g_i(x_i)$ is block separable, so is $\mathrm{prox}_{\rho g}(x) = \left(\mathrm{prox}_{\rho g_i}(x_i)\right)_{i\in[m]}$ (see [23]).

Let S and T be two maximal monotone operators on $\mathbb{R}^n$. A common problem is to find points in the zero set of $S+T$, namely, $\mathrm{zer}(S+T) := \{x \,|\, 0 \in Sx + Tx\}$. In particular, if $S = \partial f$ and $T = \partial g$ for CCP functions f and g, then $\mathrm{zer}(S+T)$ (if nonempty) consists of the minimizers of $f+g$. Suppose $\mathrm{zer}(S+T) \neq \emptyset$. Then one solution $x \in \mathrm{zer}(S+T)$ can be obtained as $x = J_T(z)$ where z is a fixed point of the nonexpansive map $(2J_S - I)(2J_T - I)$. Such a fixed point z can be found by iterating using the α-averaged map $(1-\alpha)I + \alpha(2J_S - I)(2J_T - I)$ for $\alpha \in (0,1)$, which results in the (generalized) Douglas-Rachford (D-R) algorithm [24]:

$$\begin{aligned} x^{k+1} &= J_T\left(z^k\right), \\ z^{k+1} &= z^k + 2\alpha\left[J_S(2x^{k+1} - z^k) - x^{k+1}\right], \end{aligned} \tag{6}$$

for $k = 0, 1, \ldots$. Starting from any z^0, the generated sequence x^k will converge to a solution $x^* \in \mathrm{zer}(S+T)$ and the convergence rate can be characterized [25], [9].

We now apply the D-R algorithm to Problem (4). Choose $S = \partial F$ and $T = \partial \imath_{\mathcal{A}}$. Then, $J_S = \mathrm{prox}_{\rho F}$ and $J_T = \Pi_{\mathcal{A}}$. The Douglas-Rachford algorithm for $\alpha \in (0,1)$ becomes

$$\mathbf{x}_i^{k+1} = \bar{\mathbf{z}}_i^k, \quad i \in [m]; \tag{7a}$$

$$\mathbf{z}_i^{k+1} = \mathbf{z}_i^k + 2\alpha\left(\mathrm{prox}_{\rho f_i}(2\mathbf{x}_i^{k+1} - \mathbf{z}_i^k) - \mathbf{x}_i^{k+1}\right), \ i \in [m]. \tag{7b}$$

Note that step (7a) is carried out in two fully synchronous stages: first each agent i gathers variables z_{ji}^k from all of its out-neighbors j and computes the average of z_i^k and the gathered variables as x_i^{k+1}; then, after the first stage is completed for all agents, each agent i gathers the variables x_j^{k+1} from all of its in-neighbors j and updates x_{ij}^{k+1} to x_j^{k+1}. Step (7b) requires no inter-agent communication. The whole algorithm is summarized in Algorithm 1.

As the iteration from $\mathbf{z}^k$ to $\mathbf{z}^{k+1}$ is given by an α-averaged map in (6), the sequence $\mathbf{z}^k$ obtained by Algorithm 1 converges to some $\mathbf{z}^*$ for which $\mathbf{x}^* := \bar{\mathbf{z}}^*$ yields a solution to Problem (4). Note that in general $\mathbf{x}^* \neq \mathbf{z}^*$, i.e., it is possible that $z_{ij}^* \neq z_j^*$ for some $(j,i) \in \mathcal{E}$.

Algorithm 1 Synchronous Douglas-Rachford Algorithm

1: Initialize $\mathbf{z}^0$, and let $k \leftarrow 0$
2: **repeat**
3: **for** $i = 1, \ldots, m$ **do**
4: $\mathbf{x}_i^{k+1} \leftarrow \bar{\mathbf{z}}_i^k$
5: **for** $i = 1, \ldots, m$ **do**
6: $\mathbf{z}_i^{k+1} \leftarrow \mathbf{z}_i^k + 2\alpha\left(\mathrm{prox}_{\rho f_i}(2\mathbf{x}_i^{k+1} - \mathbf{z}_i^k) - \mathbf{x}_i^{k+1}\right)$
7: $k \leftarrow k+1$
8: **until** $|\mathbf{z}^k - \mathbf{z}^{k-1}|$ is sufficiently small
9: **return** $\mathbf{x}^k$

Remark 1: Another version of the Douglas-Rachford algorithm is obtained by letting $S = \partial \imath_{\mathcal{A}}$ and $T = \partial F$:

$$\mathbf{x}_i^{k+1} = \mathrm{prox}_{\rho f_i}(\mathbf{z}_i^k), \quad i \in [m]; \tag{8a}$$

$$\mathbf{z}_i^{k+1} = \mathbf{z}_i^k + 2\alpha\big(2\bar{\mathbf{x}}_i^{k+1} - \bar{\mathbf{z}}_i^k - \mathbf{x}_i^{k+1}\big), \quad i \in [m]. \tag{8b}$$

Example 6: In Example 1, let $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2) \in \mathbb{R}^3$ where $\mathbf{z}_1 = (z_1, z_{12}) \in \mathbb{R}^2$ and $\mathbf{z}_2 = z_2 \in \mathbb{R}$. Then $\mathrm{prox}_{\rho f_1}(\mathbf{z}_1) = \mathbf{z}_1/(1+\rho)$ and $\mathrm{prox}_{\rho f_2}(\mathbf{z}_2) = \mathbf{z}_2 + \rho$. Algorithm 1 becomes

$$\begin{aligned} z_1^{k+1} &= (1 - 2\alpha\rho/(1+\rho))z_1^k, \\ z_{12}^{k+1} &= (1-\alpha)z_{12}^k + \alpha(1-\rho)/(1+\rho)z_2^k, \\ z_2^{k+1} &= (1-\alpha)z_2^k + \alpha z_{12}^k + 2\alpha\rho. \end{aligned}$$

It is easily verified that $\mathbf{z}^k$ converges to $\mathbf{z}^* = (0, 1-\rho, 1+\rho)$. From this, $\mathbf{x}^* = \bar{\mathbf{z}}^* = (0,1,1)$ is a solution to Problem (4) and thus $x^* = (0,1)$ is a solution to Problem (1).

B. Douglas-Rachford Algorithm for the Dual Problem

Let $S_i \in \mathbb{R}^{N_i \times n}$ be the selection matrix consisting of rows of the n-by-n identify matrix such that $\hat{x}_i = S_i x$, $i \in [m]$. In other words, S_i selects from $x = (x_i)_{i\in[m]}$ those variables that f_i depends on and arranges them as $\hat{x}_i = \left(x_i, (x_j)_{j\in\mathcal{N}_i^+}\right)$. Let $S = \begin{bmatrix} S_1^\top & \cdots & S_m^\top \end{bmatrix}^\top$. Then $\mathbf{z} = Sx$ satisfies $\mathbf{z} = (\mathbf{z}_i)_{i\in[m]}$ where $z_{ij} = z_j = x_j$ for all $(j,i) \in \mathcal{E}$, i.e., $\mathbf{z} \in \mathcal{A}$. Thus, the range space of S is the consensus subspace $\mathcal{A}$ and the null space of $S^\top$ is $\mathcal{A}^\perp$.

Problem (2) can be reformulated as follows:

$$\text{minimize} \ \sum_{i\in[m]} f_i(\mathbf{z}_i) \ \text{ subject to } \ \mathbf{z} = Sx. \tag{9}$$

Introduce the dual variable $\mathbf{p} \in \mathbb{R}^N$ and define the Lagrangian $L(x, \mathbf{z}, \mathbf{p}) = \sum_{i\in[m]}(f_i(\mathbf{z}_i) - \mathbf{p}_i^\top(\mathbf{z}_i - S_i x))$. Then the dual problem of (9) is

$$\text{minimize } F^*(\mathbf{p}) := \sum_{i\in[m]} f_i^*(\mathbf{p}_i) \ \text{ subject to } \ \mathbf{p} \in \mathcal{A}^\perp, \tag{10}$$

where $f_i^*(\mathbf{p}_i) := \sup_{\mathbf{z}_i}\left(\mathbf{p}_i^\top \mathbf{z}_i - f_i(\mathbf{z}_i)\right)$ is the convex conjugate of f_i. We assume that the dual problem has an optimal solution $\mathbf{p}^*$ with the same optimal value as that of the problem (2). This is the case, e.g., if L has a saddle point.

By applying the Douglas-Rachford algorithm in (6) to problem (10) with $S = \partial F^*$, $T = \partial \imath_{\mathcal{A}^\perp}$, $\alpha \in (0,1)$, and with ρ^{-1} in place of ρ, we have, for $k = 0, 1, \ldots$,

$$\mathbf{p}^{k+1} = \Pi_{\mathcal{A}^\perp}(\mathbf{w}^k) = \mathbf{w}^k - \bar{\mathbf{w}}^k,$$
$$\mathbf{w}_i^{k+1} = \mathbf{w}_i^k + 2\alpha \left(\mathrm{prox}_{f_i^*/\rho}(2\mathbf{p}_i^{k+1} - \mathbf{w}_i^k) - \mathbf{p}_i^{k+1} \right).$$

By the Moreau's decomposition $\mathrm{prox}_{f_i^*/\rho} + \rho^{-1} \mathrm{prox}_{\rho f_i} \circ \rho I = I$, the above iteration can be rewritten as

$$\mathbf{w}_i^{k+1} = \mathbf{w}_i^k - 2\alpha \bar{\mathbf{w}}_i^k - 2\alpha\rho^{-1} \mathrm{prox}_{\rho f_i}(\rho \mathbf{w}_i^k - 2\rho \bar{\mathbf{w}}_i^k),$$

which is carried out by Algorithm 2 below. Starting from any $\mathbf{w}^0$, the sequence $\mathbf{w}^k$ converges to some $\mathbf{w}^*$ for which $\mathbf{p}^* = \mathbf{w}^* - \bar{\mathbf{w}}^*$ is an optimal solution to the dual problem (10).

Algorithm 2 Synchronous Dual D-R Algorithm

Initialize $\mathbf{w}^0$, and let $k \leftarrow 0$
repeat
  for $i = 1, \ldots, m$ **do**
    $\mathbf{u}_i^{k+1} \leftarrow \bar{\mathbf{w}}_i^k$
  for $i = 1, \ldots, m$ **do**
    $\mathbf{v}_i^{k+1} \leftarrow \mathrm{prox}_{\rho f_i}(\rho \mathbf{w}_i^k - 2\rho \mathbf{u}_i^{k+1})$
    $\mathbf{w}_i^{k+1} \leftarrow \mathbf{w}_i^k - 2\alpha \mathbf{u}_i^{k+1} - 2\alpha\rho^{-1} \mathbf{v}_i^{k+1}$
  $k \leftarrow k + 1$
until $|\mathbf{w}^k - \mathbf{w}^{k-1}|$ is sufficiently small
return $\mathbf{w}^k - \bar{\mathbf{w}}^k$

Remark 2: It is a standard result that the Douglas-Rachford algorithm on the dual problem is equivalent to the ADMM algorithm [4]. For Problem (9) we define the augmented Lagrangian $L_\rho(x, \mathbf{z}, \mathbf{y}) = \sum_{i\in[m]} f_i(\mathbf{z}_i) + \mathbf{y}_i^\top (S_i x - \mathbf{z}_i) + \frac{1}{2\rho} \|S_i x - \mathbf{z}_i\|^2$. The ADMM algorithm first minimizes w.r.t. x and $\mathbf{z}$ in a sequential fashion, and then uses the primal residue to update the dual variable $\mathbf{y}$ (see [4]):

$$\mathbf{x}^{k+1} = \bar{\mathbf{z}}^k - \rho \bar{\mathbf{y}}^k, \tag{11a}$$
$$\mathbf{z}_i^{k+1} = \mathrm{prox}_{\rho f_i}(\mathbf{x}_i^{k+1} + \rho \mathbf{y}_i^k) = \mathrm{prox}_{\rho f_i}\left(\bar{\mathbf{z}}_i^k + \rho(\mathbf{y}_i^k - \bar{\mathbf{y}}_i^k)\right),$$
$$\mathbf{y}^{k+1} = \mathbf{y}^k - \bar{\mathbf{y}}^k - (\mathbf{z}^{k+1} - \bar{\mathbf{z}}^k)/\rho. \tag{11b}$$

V. ASYNCHRONOUS ALGORITHMS

The algorithms proposed in Section IV can be turned into asynchronous ones by utilizing Theorem 1. Some algorithms will have "better" asynchronous implementations compared to others. Here, the criteria for comparing different asynchronous implementations are in terms of the computation and communication overheads incurred in each iteration.

A. Asynchronous Douglas-Rachford Algorithm

By applying Theorem 1 to the Douglas-Rachford algorithm (8), we obtain the following asynchronous algorithm:

For $k = 0, 1, \ldots$, pick $i \in [m]$ with i.i.d. probability $p_i > 0$
$\mathbf{x}_j^{k+1} = \mathrm{prox}_{\rho f_j}\left(\mathbf{z}_j^k\right), \ \ \forall j \in \mathcal{N}_i \cup (\cup_{\ell \in \mathcal{N}_i^+} \mathcal{N}_\ell^-)$;
$\mathbf{z}_i^{k+1} = \mathbf{z}_i^k + 2\alpha \left(2\bar{\mathbf{x}}_i^{k+1} - \bar{\mathbf{z}}_i^k - \mathbf{x}_i^{k+1}\right)$ for the chosen i.

Even though at each round only one agent i is activiated to carry out the update, its updated variable $\mathbf{z}_i^{k+1}$ relies on $\bar{\mathbf{x}}_i^{k+1}$ whose evaluation requires each agent in an extended (2-hop) neighborhood $\mathcal{N}_i \cup (\cup_{\ell \in \mathcal{N}_i^+} \mathcal{N}_\ell^-)$ to evaluate its proximal operator and send out its local information. This may result in large computation and communication overheads. For instance, for the coordinated optimization problem in Example 5, regardless of which agent is chosen to do the update, all the agents need to evaluate their proximal operators and communicate the results to their neighbors.

A better option is to apply Theorem 1 to Algorithm 1:

For $k = 0, 1, \ldots$, pick $i \in [m]$ with i.i.d. probability $p_i > 0$
$$\mathbf{x}_i^{k+1} = \bar{\mathbf{z}}_i^k; \tag{13a}$$
$$\mathbf{z}_i^{k+1} = \mathbf{z}_i^k + 2\alpha \left(\mathrm{prox}_{\rho f_i}(2\mathbf{x}_i^{k+1} - \mathbf{z}_i^k) - \mathbf{x}_i^{k+1}\right). \tag{13b}$$

In each round, only the activated agent i needs to evaluate its proximal operator once. Starting from any $\mathbf{z}^0$, the sequence $\mathbf{z}^k$ generated by Algorithm (13) converges almost surely to some $\mathbf{z}^*$ for which $\bar{\mathbf{z}}^*$ is a solution to Problem (4).

Example 7: For Example 1, Algorithm (13) becomes:

$$\begin{cases} z_1^{k+1} = (1 - 2\alpha\rho/(1+\rho)) z_1^k, \\ z_{12}^{k+1} = (1-\alpha) z_{12}^k + \alpha(1-\rho)/(1+\rho) z_2^k, \\ z_2^{k+1} = z_2^k, \end{cases}$$

if agent 1 is activated, and

$$z_1^{k+1} = z_1^k, \ z_{12}^{k+1} = z_{12}^k, \ z_2^{k+1} = (1-\alpha) z_2^k + \alpha z_{12}^k + 2\alpha\rho$$

if agent 2 is activated. With probability one, $\mathbf{z}^k \to \mathbf{z}^* = (0, 1-\rho, 1+\rho)$ and $\mathbf{x}^k \to \bar{\mathbf{z}}^* = (0, 1, 1)$.

In Algorithm (13), to compute $\bar{\mathbf{z}}_i^k$ in step (13a), agent i needs to collect $z_{\ell i}^k$ from its out-neighbors ℓ and $\bar{z}_j^k$ from its in-neighbors j. The latter requires agents j to gather data from their respective out-neighbors. To avoid this, we can let each agent $i \in [m]$ maintain an extra variable $\bar{z}_i \in \mathbb{R}^{n_i}$ that always has the latest averaged value of z_i and $z_{\ell i}$ for $\ell \in \mathcal{N}_i^-$ ($\bar{z}_i$ is null if agent i does not have a local variable). Then Algorithm (13) is equivalent to Algorithm 3 below.

Algorithm 3 Asynchronous Douglas-Rachford Algorithm

Choose any $\mathbf{z}^0$, and let $k \leftarrow 0$
for $i = 1, \ldots, m$ **do**
  $\bar{z}_i^0 \leftarrow (z_i^0 + \sum_{\ell \in \mathcal{N}_i^-} z_{\ell i}^0)/(|\mathcal{N}_i^-| + 1)$
repeat
  Pick $i \in [m]$ with i.i.d. probability $p_i > 0$
  $x_i^{k+1} \leftarrow \bar{z}_i^k$
  for $j \in \mathcal{N}_i^+$ **do**
    $x_{ij}^{k+1} \leftarrow \bar{z}_j^k$
  $\mathbf{z}_i^{k+1} \leftarrow \mathbf{z}_i^k + 2\alpha \left(\mathrm{prox}_{\rho f_i}(2\mathbf{x}_i^{k+1} - \mathbf{z}_i^k) - \mathbf{x}_i^{k+1}\right)$
  $\bar{z}_i^{k+1} \leftarrow \bar{z}_i^k + (z_i^{k+1} - z_i^k)/(|\mathcal{N}_i^-| + 1)$
  for $j \in \mathcal{N}_i^+$ **do**
    $\bar{z}_j^{k+1} \leftarrow \bar{z}_j^k + (z_{ij}^{k+1} - z_{ij}^k)/(|\mathcal{N}_j^-| + 1)$
  $k \leftarrow k + 1$
until k is sufficiently large
return $\mathbf{x}^k$

Algorithm 3 has low communication and computation complexity. In each round, the activated agent i only needs to communicate with its in-neighbors by collecting information in step 8 and sending information in step 12; its in-neighbors perform simple updates (step 12); while all other agents (even if they are out-neighbors of agent i) can remain idle. For the coordinated optimization problem in Example 5, if the activated agent $i \in \{2, \ldots, m\}$, then only agent i needs to carry out steps 9 and 10, while all other agents including agent 1 remain idle. On average, in each round of Algorithm 3, there are $\sum_i 2p_i|\mathcal{N}_i^+| \leq \max_i 2|\mathcal{N}_i^+|$ one-way transmissions, $\sum_i 2p_i(N_i - n_i) \leq \max_i 2(N_i - n_i)$ scalar variables transmitted, and one proximal operator evaluation.

The following result is immediate from Theorem 1.

Corollary 1: Suppose $\alpha \in (0, 1)$, $\rho > 0$, and $p_i > 0$ for $i \in [m]$. Starting from any $\mathbf{z}^0$, with probability one the sequence $\mathbf{x}^k$ generated by Algorithm 3 converges to a solution to Problem (2).

B. Asynchronous Dual Douglas-Rachford Algorithm

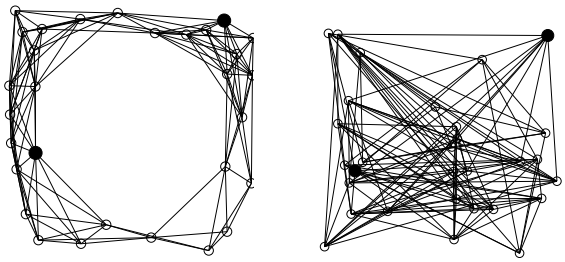
Let each agent i maintain the variables $\mathbf{w}_i, \mathbf{u}_i, \mathbf{v}_i \in \mathbb{R}^{N_i}$ and $\bar{w}_i \in \mathbb{R}^{n_i}$. The asynchronous version of Algorithm 2 obtained using Theorem 1 is given by Algorithm 4 below. In each round, with the agent i activated, the proximal operator is evaluated only once (step 9). Agent i communicates only with its out-neighbors (collect in step 8 and send in step 13). The expected numbers of one-way communications and the expected number of scalar variables transmitted in each round are the same as those of Algorithm 3.

Algorithm 4 Asynchronous Dual D-R Algorithm

1: Choose any $\mathbf{w}^0$, and let $k \leftarrow 0$
2: **for** $i = 1, \ldots, m$ **do**
3: $\quad \bar{w}_i^0 \leftarrow (w_i^0 + \sum_{\ell \in \mathcal{N}_i^-} w_{\ell i}^0)/(|\mathcal{N}_i^-| + 1)$
4: **repeat**
5: $\quad$ Pick $i \in [m]$ with i.i.d. probability $p_i > 0$
6: $\quad u_i^{k+1} \leftarrow \bar{w}_i^k$
7: $\quad$ **for** $j \in \mathcal{N}_i^+$ **do**
8: $\quad\quad u_{ij}^{k+1} \leftarrow \bar{w}_j^k$
9: $\quad \mathbf{v}_i^{k+1} \leftarrow \mathrm{prox}_{\eta f_i}(\eta \mathbf{w}_i^k - 2\eta \mathbf{u}_i^{k+1})$
10: $\quad \mathbf{w}_i^{k+1} \leftarrow \mathbf{w}_i^k - 2\alpha \mathbf{u}_i^{k+1} - 2\alpha\eta^{-1}\mathbf{v}_i^{k+1}$
11: $\quad \bar{w}_i^{k+1} \leftarrow \bar{w}_i^k + (w_i^{k+1} - w_i^k)/(|\mathcal{N}_i^-| + 1)$
12: $\quad$ **for** $j \in \mathcal{N}_i^+$ **do**
13: $\quad\quad \bar{w}_j^{k+1} \leftarrow \bar{w}_j^k + (w_{ij}^{k+1} - w_{ij}^k)/(|\mathcal{N}_j^-| + 1)$
14: $\quad k \leftarrow k + 1$
15: **until** k is sufficiently large
16: **return** $\mathbf{w}^k - \bar{\mathbf{w}}^k$

Similar to Corollary 1, the following result can be proved.

Corollary 2: Suppose $\alpha \in (0, 1)$, $\rho > 0$, and $p_i > 0$ for $i \in [m]$. Starting from any initial $\mathbf{w}^0$, the sequence $\mathbf{w}^k - \bar{\mathbf{w}}^k$ obtained by Algorithm 4 converges with probability one to a solution to the dual problem (10).



(a) Ground truth (b) Initial guess

Fig. 1: A network localization problem with 2 anchors and 28 free agents.

VI. Network Localization Application

In this section the network localization problem is used to demonstrate the effectiveness of the proposed algorithms. A number of agents are randomly placed inside a planar region. Among them, some are anchors indexed by $\mathcal{I}_a$ who know their exact locations $(x_i^*)_{i \in \mathcal{I}_a}$, while the other are free agents indexed by $\mathcal{I}_f$ who need to estimate their positions $(x_i^*)_{i \in \mathcal{I}_f}$ based on the relative orientation measurements between pairs of agents within a certain measurement range. This is an instance of the convex feasibility problem in Example 3 where the local feasibility constraint for agent $i \in \mathcal{I}_f$ is of the form $\angle(x_i^* - x_j^*) = \theta_{ij}$ for all neighboring agents $j \in \mathcal{I}_a \cup \mathcal{I}_f$. If there are at least two anchors and the whole network has an infinitesimally rigid graph, then the network is localizable, i.e., there is a unique solution $(x_i^*)_{i \in \mathcal{I}_f}$ satisfying all the relative orientation constraints [13, Thm. 15]. Fig. 1 (a) shows the true locations of agents in a network with two anchors (solid circles) and 28 free agents (hollow circles), with each edge representing a relative orientation constraint between two agents. All tested algorithms start from the same random initial guess depicted in Fig. 1 (b).

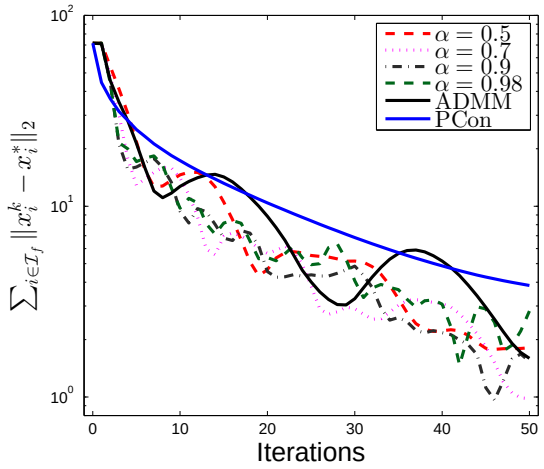


Fig. 2: Comparison of convergence rates of tested algorithms.

In Fig. 2, we compare the convergence rates of three synchronous algorithms, Algorithms 1, the ADMM algo-

rithm (11), and the PCon Algorithm [26] based on para-contractions. For Algorithm 1, the parameter α is set to 0.5, 0.7, 0.9, 0.98, respectively, while the parameter ρ has no effect on the algorithm as each f_i is an indicator function. For the ADMM algorithm (11), $\rho = 0.01, 0.1, 1, 10, 1000$ are tested and the best result ($\rho = 0.1$) is plotted in Fig. 2. For the PCon Algorithm, the result with the best parameter value $\alpha_i \equiv 1.9$ based on experiments is included here. For this example, the Algorithm 1 converges at a similar rate as the ADMM algorithm (11) (and with less performance oscillations), and faster than the PCon algorithm.

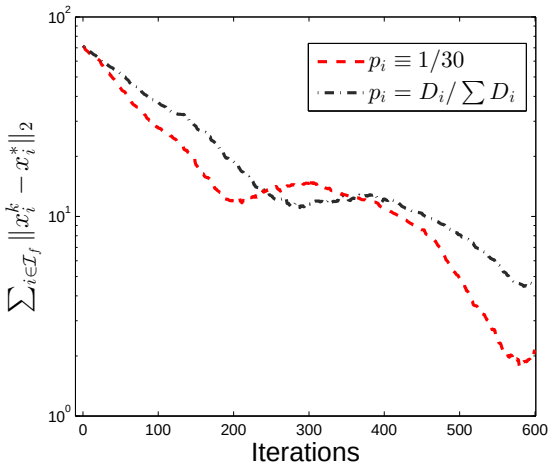


Fig. 3: Random outcomes of Algorithm 3 with $\alpha = 0.5$ and two different sets of probabilities p_i's.

Fig. 3 shows representative random outcomes of applying Algorithm 3 to the same localization problem with $\alpha = 0.5$ and two different sets of probabilities: $p_i = 1/30$, and $p_i = D_i/\sum D_i$ where D_i is the degree of node i in the dependency graph. It is observed that the iterative results indeed tend to the optimal solution. Further, updating highly connected nodes more frequently does not seem to speed up convergence in this example. Note that, compared to Algorithm 1, the number of iterations needed for achieving the same convergence performance is much larger due to the fact that at each round only one agent is performing computation as opposed to 28 agents in Algorithm 1.

VII. CONCLUSIONS AND FUTURE WORKS

Several synchronous algorithms are proposed for solving the optimization problems on agent networks with locally coupled objective functions and constraints. Asynchronous implementations of the proposed algorithms are presented. Convergence of the proposed algorithms to optimal solutions are demonstrated via examples.

REFERENCES

[1] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Trans. Industrial informatics*, vol. 9, no. 1, pp. 427–438, 2013.

[2] L. Xiao, M. Johansson, and S. P. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Trans. Communications*, vol. 52, no. 7, pp. 1136–1144, 2004.

[3] A. N. Venkat, J. B. Rawlings, and S. J. Wright, "Stability and optimality of distributed model predictive control," in *IEEE Int. Conf. Decision and Control*. IEEE, 2005, pp. 6680–6685.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

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

[6] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[7] H. Terelius, U. Topcu, and R. M. Murray, "Decentralized multi-agent optimization via dual decomposition," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 11 245–11 251, 2011.

[8] W. Shi, Q. Ling, G. Wu, and W. Yin, "A proximal gradient algorithm for decentralized composite optimization," *IEEE Trans. Signal Processing*, vol. 63, no. 22, pp. 6013–6023, 2015.

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

[10] 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.

[11] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Processing*, vol. 58, no. 10, pp. 5262–5276, 2010.

[12] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Trans. Robot. Automat.*, vol. 17, no. 6, pp. 947–951, 2001.

[13] G. Zhu and J. Hu, "A distributed continuous-time algorithm for network localization using angle-of-arrival information," *Automatica*, vol. 50, no. 1, pp. 53–63, 2014.

[14] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2017.

[15] E. K. Ryu and S. Boyd, "A primer on monotone operator methods," *Appl. Comput. Math*, vol. 15, no. 1, pp. 3–43, 2016.

[16] E. Wei and A. Ozdaglar, "On the O(1/k) convergence of asynchronous distributed alternating direction method of multipliers," in *IEEE Global Conf. Signal and Information Processing*. IEEE, 2013, pp. 551–554.

[17] P. Bianchi, W. Hachem, and F. Iutzeler, "A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization," *IEEE Trans. Automatic Control*, vol. 61, no. 10, pp. 2947–2957, 2016.

[18] Z. Peng, T. Wu, Y. Xu, M. Yan, and W. Yin, "Coordinate friendly structures, algorithms and applications," *arXiv Preprint 1601.00863*, 2016.

[19] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," in *IEEE Int. Conf. Decision and Control*. IEEE, 2013, pp. 3671–3676.

[20] T. Wu, K. Yuan, Q. Ling, W. Yin, and A. H. Sayed, "Decentralized consensus optimization with asynchrony and delays," in *50th Asilomar Conf. Signals, Systems and Computers*. IEEE, 2016, pp. 992–996.

[21] Z. Peng, Y. Xu, M. Yan, and W. Yin, "ARock: an algorithmic framework for asynchronous parallel coordinate updates," *SIAM Journal on Scientific Computing*, vol. 38, no. 5, pp. A2851–A2879, 2016.

[22] T.-H. Chang, M. Hong, W.-C. Liao, and X. Wang, "Asynchronous distributed ADMM for large-scale optimization – Part I: algorithm and convergence analysis," *IEEE Trans. Signal Processing*, vol. 64, no. 12, pp. 3118–3130, 2016.

[23] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[24] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, no. 1, pp. 293–318, 1992.

[25] B. He and X. Yuan, "On the O(1/n) convergence rate of the douglas–rachford alternating direction method," *SIAM Journal on Numerical Analysis*, vol. 50, no. 2, pp. 700–709, 2012.

[26] Y. Xiao and J. Hu, "Distributed solutions of convex feasibility problems with sparsely coupled constraints," in *IEEE Int. Conf. Decision and Control*. IEEE, 2017, pp. 3386–3392.