Distributed Aggregative Optimization over Multi-Agent Networks

Xiuxian Li, Lihua Xie, and Yiguang Hong

Abstract—This paper proposes a new framework for distributed optimization, called distributed aggregative optimization, which allows local objective functions to be dependent not only on their own decision variables, but also on the average of summable functions of decision variables of all other agents. To handle this problem, a distributed algorithm, called distributed gradient tracking (DGT), is proposed and analyzed, where the global objective function is strongly convex, and the communication graph is balanced and strongly connected. It is shown that the algorithm can converge to the optimal variable at a linear rate. A numerical example is provided to corroborate the theoretical result.

Index Terms—Distributed algorithm, aggregative optimization, multiagent networks, strongly convex function, linear convergence rate.

I. INTRODUCTION

Distributed optimization has received immense attention in the past decade, mostly inspired by advanced and inexpensive sensors, big data, and large-scale networks, and so on. In distributed optimization, a network consisting of a family of agents is usually introduced to capture the communication pattern among all agents, where each agent is only accessible to partial (and maybe private) information on the global optimization problem. In this case, the agents in the network aim to cooperatively, by local information exchange, solve the global optimization problem.

To date, a large volume of algorithms have been devised for distributed optimization problems. Generally speaking, the existing algorithms can be roughly summarized as two classes: consensusbased algorithms and dual-decomposition-based algorithms. Wherein, consensus-based algorithms employ the consensus idea to align the estimated variables of all agents, for which existing algorithms include distributed subgradient [1], diffusion adaptation strategy [2], fast distributed gradient [3], asynchronous distributed gradient [4], stochastic mirror descent [5], and distributed quasi-monotone subgradient algorithm [6], etc. With regard to dual-decomposedbased algorithms, dual variables are usually introduced by viewing the synchronization of all local variables as equality constraints, including alternating direction method of multipliers (ADMM) [7], EXTRA [8], augmented Lagrangian method [9], distributed dual proximal gradient [10], and distributed forward-backward Bregman splitting [11].

From another viewpoint, a variety of scenarios have so far been considered for distributed optimization. The simplest case is to minimize an objective/cost function without any constraints [1], [8], [12], including feasible set constraints, equality and inequality constraints, where the objective function is separable and composed of local objective functions. A little more complex case is to address distributed optimization with global/local feasible set constraints [13]–[15], that is, the decision variable must stay within some prespecified nonempty set that is often assumed to be closed and convex.

Moreover, the scenario with local (affine) equality constraints are addressed, for example, in [16], while local inequality constraints are investigated such as in [17], and global inequality constraints that can be realized by all agents are taken into account in the literature, see [18] for an example. Furthermore, the case with globally coupled inequality constraints, where individual agent is only capable of accessing partial information on the global inequality constraints, is studied such as in [19]–[24], and meanwhile, time-varying objective functions and/or constraint functions are also considered in recent years [25]–[28].

With careful observation, it can be found that distributed optimization studied in the aforementioned works focus on the case where a global objective function is a sum of local objective functions, which are dependent only on their own decision variables. To be specific, the problem is in the form $\sum_{i=1}^{N} f_i(x_i)$ such that $x_i = x_j$ for all $i \neq j$, maybe subject to inequality constraints, from which it is easy to see that each f_i is a function with respect to only x_i , independent of any other variables $x_j, j \neq i$. However, in a multitude of practical applications, local objective functions are also determined by other agents' variables. For example, in multi-agent formation control, each objective function often relies on variables (such as positions or velocities) of all its neighbors, and this scenario has been considered such as in [29] and [30] (cf. Remark 4). As another example, the average of all variables, i.e., $\sum_{i=1}^{N} x_i/N$, is a vital parameter for all agents in a network, which can be discovered from a large number of applications, such as optimal placement problem, transportation network, and formation control, etc. For instance, in formation control, a group of networked agents desire to achieve a geometric pattern, and simultaneously, they may plan to encircle an important target, which can be cast as a target tracking problem for the center of all agents. Therefore, it is significant to deal with the scenario where the average of all variables is involved in local objective functions. From the theoretical perspective, when each local function f_i also depends on variables of other agents (such as the average $\sum_{i=1}^{N} x_i/N$), the problem will be more challenging since other variables (such as the average $\sum_{i=1}^{N} x_i/N$) and related gradients are unavailable to agent *i*.

Motivated by the above facts, this paper aims to formulate and study a new framework for distributed optimization, called distributed aggregative optimization, for which a distributed algorithm, called distributed gradient tracking (DGT), is developed and analyzed. It is shown that the proposed algorithm has a linear convergence speed under mild assumptions, such as strong convexity of the global objective function and a directed balanced communication graph. The contributions of this paper are as follows: (1) a new distributed aggregative optimization is formulated for the first time; (2) a linearly convergent distributed algorithm is proposed and analyzed rigorously; and (3) a numerical example is provided to support the theoretical result.

The rest of this paper is structured as follows. Some preliminaries and the problem formulation are provided in Section II, followed by the main result in Section III. In Section IV, a numerical example is presented to corroborate the theoretical result, and the conclusion is drawn in Section V.

X. Li and L. Xie are with School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798 (e-mail: xxli@ieee.org; elhxie@ntu.edu.sg).

Y. Hong is with Key Laboratory of Systems and Control, Institute of Systems Science, Chinese Academy of Sciences, Beijing, 100190, China (email: yghong@iss.ac.cn).

Notations: Let \mathbb{R}^n and \mathbb{C} be the set of vectors with dimension n > 0 and the set of complex numbers, respectively. Define $[k] = \{1, 2, \ldots, k\}$ for an integer k > 0. Denote by $col(z_1, \ldots, z_k)$ the column vector by stacking up z_1, \ldots, z_k . Let $\|\cdot\|, x^{\top}$, and $\langle x, y \rangle$ be the standard Euclidean norm, the transpose of $x \in \mathbb{R}^n$, and standard inner product of $x, y \in \mathbb{R}^n$. Let 1 and 0 be column vectors of compatible dimension with all entries being 1 and 0, respectively, and I be the compatible identity matrix. Let $\rho(M)$ be the spectral radius of a square matrix M. \otimes is the Kronecker product. Let $J := \frac{1}{N} \mathbf{11}^{\top}$ and $\mathcal{J} := J \otimes I$ with compatible dimension.

II. PRELIMINARIES

A. Graph Theory

The communication pattern among all agents is captured by a simple graph in this paper, denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the node set $\mathcal{V} = \{1, \ldots, N\}$ and the edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. An edge $(j, i) \in \mathcal{E}$ means that node j can send information to node i, where j is called an in-neighbor of i. Denote by $\mathcal{N}_i = \{j : (j, i) \in \mathcal{E}\}$ the in-neighbor set of node i. The graph \mathcal{G} is called undirected if $(i, j) \in \mathcal{E}$ is equivalent to $(j, i) \in \mathcal{E}$, and directed otherwise. The communication matrix $A = (a_{ij}) \in \mathbb{R}^{N \times N}$ is defined by: $a_{ij} > 0$ if $(j, i) \in \mathcal{E}$, and $a_{ij} = 0$ otherwise.

The following standard assumptions on the communication graph are postulated.

Assumption 1. The following hold for the interaction graph:

- 1) The graph G is strongly connected;
- 2) The matrix A is doubly stochastic, i.e., $\sum_{j=1}^{N} a_{ij} = 1$ and $\sum_{i=1}^{N} a_{ij} = 1$ for all $i, j \in [N]$.

It should be noted that some approaches have been brought forward in the literature in order to hold the double stochasticity condition, for example, the uniform weights [31] and the least-mean-square consensus weight rules [32]. In addition, some distributed strategies have been proposed in [33] for strongly connected directed graphs to compute a doubly stochastic assignment in finite time.

B. Convex Optimization

For a convex function $g : \mathbb{R}^n \to \mathbb{R}$, the *subdifferential*, denoted as $\partial g(x)$, of g at x is defined by

$$\partial g(x) = \{ s \in \mathbb{R}^n : g(y) - g(x) \ge s^\top (y - x), \ \forall y \in \mathbb{R}^n \},\$$

and each element in $\partial g(x)$ is called a *subgradient*. When g is differentiable at x, the subdifferential $\partial g(x)$ only contains one element, which is usually called *gradient*, denoted as $\nabla g(x)$.

The differentiable function g is called μ -strongly convex if for all $x, y \in \mathbb{R}^n$,

$$g(x) \ge g(y) + \nabla g(y)^{\top} (x - y) + \frac{\mu}{2} ||x - y||^2.$$
 (1)

C. Problem Formulation

This paper proposes a new framework for distributed optimization in a network composed of N agents, called *distributed aggregative optimization*, given as follows:

$$\min_{x \in \mathbb{R}^n} \quad f(x) := \sum_{i=1}^N f_i(x_i, \sigma(x)), \tag{2}$$

$$\sigma(x) := \frac{\sum_{i=1}^{N} \phi_i(x_i)}{N},\tag{3}$$

where $x = col(x_1, \ldots, x_N)$ is the global decision variable with $x_i \in \mathbb{R}^{n_i}$, $n := \sum_{i=1}^N n_i$, and $f_i : \mathbb{R}^n \to \mathbb{R}$ is the local objective function. In problem (2), the global function f is not known to any

agent, and each agent can only privately access the information on f_i . Moreover, each agent $i \in [N]$ is only aware of the decision variable x_i without any knowledge of x_j 's for all $j \neq i$. Moreover, the term $\sigma(x)$ is an aggregative information of all agents' variables, and the function $\phi_i : \mathbb{R}^{n_i} \to \mathbb{R}^d$ is only accessible to agent *i*. The goal is to design distributed algorithms to seek an optimal decision variable for problem (2).

Remark 1. It should be noted that distributed aggregative optimization is proposed here for the first time, to our best knowledge, which is different from aggregative games [34], [35]. The substantial difference lies in that all agents in problem (2) aim to cooperatively find an optimal variable for the sum of all local objective functions, while the objective of aggregative games is to find the Nash equilibrium in a noncooperative manner since each agent desires to minimize only its own objective function. This can be seen from the following simple example.

Example 1. As a simple example, let us consider two agents in a network in the scalar space \mathbb{R} without feasible set constraints. Let $f_1(x) = (x_1 - 1)^2 + \sigma^2(x) = (x_1 - 1)^2 + (x_1 + x_2)^2/4$ and $f_2(x) = (x_2 - 2)^2 + \sigma^2(x) = (x_2 - 2)^2 + (x_1 + x_2)^2/4$. As a result, for distributed aggregative optimization, the optimal variable of $f(x) = f_1(x) + f_2(x)$ can be easily calculated, by $\nabla_{x_1} f(x) = 0$ and $\nabla_{x_2} f(x) = 0$, as $x_1 = 1/4, x_2 = 5/4$. On the other hand, as for aggregative games, the Nash equilibrium can be computed, by $\nabla_{x_1} f_1(x) = 0$ and $\nabla_{x_2} f_2(x) = 0$, as $x_1 = 1/2, x_2 = 3/2$. It is apparent to see that the Nash equilibrium $x_1 = 1/2, x_2 = 3/2$ is not the same as the global optimizer of f(x), i.e., $x_1 = 1/4, x_2 = 5/4$. In other words, the Nash equilibrium is generally not the optimal decision variable due to the noncooperative nature of all agents in aggregative games.

To move forward, for brevity, let $\nabla_1 f_i(x_i, \sigma(x))$ and $\nabla_2 f_i(x_i, \sigma(x))$ denote $\nabla_{x_i} f_i(x_i, \sigma(x))$ and $\nabla_\sigma f_i(x_i, \sigma(x))$, respectively, for all $i \in [N]$. And for $x \in \mathbb{R}^n$ and $y = col(y_1, \ldots, y_N) \in \mathbb{R}^{Nd}$, define $f(x, y) := \sum_{i=1}^N f_i(x_i, y_i)$, $\nabla_1 f(x, y) := col(\nabla_1 f_1(x_1, y_1), \ldots, \nabla_1 f_N(x_N, y_N))$ and $\nabla_2 f(x, y) := col(\nabla_2 f_1(x_1, y_1), \ldots, \nabla_2 f_N(x_N, y_N))$. It is now necessary to list some assumptions.

Assumption 2. The following hold for problem (2):

- 1) The global objective function f(x) is differentiable, μ -strongly convex, and L_1 -smooth on \mathbb{R}^n , that is, $\|\nabla f(x) - \nabla f(x')\| \le L_1 \|x - x'\|$ for all $x, x' \in \mathbb{R}^n$. Also, $\nabla_1 f(x, y) + \nabla \phi(x) \mathbf{1}_N \otimes \frac{1}{N} \sum_{i=1}^N \nabla_2 f_i(x_i, y_i)$ is L_1 -Lipschitz;
- 2) $\nabla_2 f(x,y)$ is L_2 -Lipschitz continuous, that is, $\|\nabla_2 f(x,y) \nabla_2 f(x',y')\| \leq L_2(\|x-x'\| + \|y-y'\|)$ for all $x,x' \in \mathbb{R}^n$ and $y,y' \in \mathbb{R}^{Nd}$;
- 3) All ϕ_i 's are differentiable, and there exists a constant $L_3 > 0$ such that $\|\nabla \phi_i(x_i)\| \leq L_3$ for all $x_i \in \mathbb{R}^{n_i}$ and $i \in [N]$.

It should be noted that the Lipschitz property of $\nabla f(x)$ and $\nabla_1 f(x, y) + \nabla \phi(x) \mathbf{1}_N \otimes \frac{1}{N} \sum_{i=1}^N \nabla_2 f_i(x_i, y_i)$ in Assumption 2.1 can be ensured by Assumptions 2.2 and 2.3 along with the boundedness of $\nabla_2 f(x, y)$ and the Lipschitz property of $\nabla_1 f(x, y)$ and $\nabla \phi_i(x_i)$, which are standard in distributed optimization and game theory (e.g., [1], [3], [15], [19], [24], [34], [35]). Please also note that it is only assumed the strong convexity of the global objective functions f_i 's.

To conclude this section, it is useful to display a few lemmas.

Lemma 1 ([36]). For an irreducible nonnegative matrix $M \in \mathbb{R}^{n \times n}$, it is primitive if it has at least one non-zero diagonal entry.

Lemma 2 ([36]). For an irreducible nonnegative matrix $M \in \mathbb{R}^{n \times n}$, there hold (i) $\rho(M) > 0$ is an eigenvalue of M, (ii) $Mx = \rho(M)x$ for some positive vector x, and (iii) $\rho(M)$ is an algebraically simple eigenvalue.

Lemma 3. Let $F : \mathbb{R}^n \to \mathbb{R}$ be μ -strongly convex and L-smooth. Then $||x - \alpha \nabla F(x) - (y - \alpha \nabla F(y))|| \le (1 - \mu \alpha) ||x - y||$ for all $x, y \in \mathbb{R}^n$, where $\alpha \in (0, 1/L]$.

Proof. It is known that a convex function $f : \mathbb{R}^n \to \mathbb{R}$ is *l*-smooth is equivalent to the convexity of $\frac{l}{2} ||x||^2 - f(x)$. Thus, by L-smoothness of F, one has that $\frac{L}{2}||x||^2 - F(x)$ is convex, and then $\frac{1}{2\alpha}||x||^2 - F(x)$ is $(\frac{1}{2\alpha} - \frac{L}{2})$ -strongly convex for $\alpha \in (0, 1/L]$, which further implies that $H(x) := \frac{1}{2}||x||^2 - \alpha F(x)$ is $(\frac{1}{2} - \frac{L\alpha}{2})$ -strongly convex. Meanwhile, it is easy to verify that $\frac{1-\mu\alpha}{2}||x||^2 - H(x) = \alpha(F(x) - \frac{\mu}{2}||x||^2)$ is convex since $F(x) - \frac{\mu}{2}||x||^2$ is convex due to the μ -strong convertive of F. Therefore, H is $(1 - \mu\alpha)$ -smooth, i.e., $||\nabla H(x)| = \frac{\pi}{2}H(x)$ convexity of F. Therefore, H is $(1 - \mu\alpha)$ -smooth, i.e., $\|\nabla H(x) - \mu\alpha\|$ $\nabla H(y) \leq (1 - \mu \alpha) \|x - y\|$ for all $x, y \in \mathbb{R}^n$, thus ending the proof.

Lemma 4. Under Assumption 1, there hold (i) $\mathcal{AJ} = \mathcal{JA} = \mathcal{J}$, and (ii) $||\mathcal{A}x - \mathcal{J}x|| \leq \rho ||x - \mathcal{J}x||$ for any $x \in \mathbb{R}^{Nd}$, where $\mathcal{A} := A \otimes I_d$ and $\rho := ||A - J|| < 1.$

Proof. The assertion (i) is trivial to verify. For the assertion (ii), it is easy to see that $\|\mathcal{A}x - \mathcal{J}x\| = \|(\mathcal{A} - \mathcal{J})x - (\mathcal{A} - \mathcal{J})\mathcal{J}x\| \le 1$ $\|\mathcal{A} - \mathcal{J}\| \|x - \mathcal{J}x\| = \|A - J\| \|x - \mathcal{J}x\|$. Invoking the doublestochasticity of A and the Perron-Frobenius theorem [36], one has ||A - J|| < 1. This ends the proof.

Lemma 5. [37] Let $X, E \in \mathbb{R}^{n \times n}$ with λ being a simple eigenvalue of X. Let w and v be the left and right eigenvectors of X associated with the eigenvalue λ , respectively. Then,

- 1) for each $\epsilon > 0$, there exists a $\delta > 0$ such that, $\forall t \in \mathbb{C}$ with $|t| < \delta$, there is a unique eigenvalue $\lambda(t)$ of X + tE such that $\begin{aligned} &|\lambda(t) - \lambda - t \frac{w^{\top} E v}{w^{\top} v}| \leq |t|\epsilon, \\ &2) \ \lambda(t) \ is \ continuous \ at \ t = 0, \ and \ \lim_{t \to 0} \lambda(t) = \lambda, \\ &3) \ \lambda(t) \ is \ differentiable \ at \ t = 0, \ and \ \frac{d\lambda(t)}{dt}\Big|_{t=0} = \frac{w^{\top} E v}{w^{\top} v}. \end{aligned}$

III. MAIN RESULT

This section presents the algorithm design and analysis. In doing so, a distributed algorithm, called *distributed gradient tracking* (DGT for short), for solving (2) is proposed for each agent $i \in [N]$ as in Algorithm 1.

Algorithm 1 Distributed Gradient Tracking (DGT)

- 1: **Initialization:** Stepsize $\alpha > 0$, and initial conditions $x_{i,0} \in \mathbb{R}^{n_i}$, $\sigma_{i,0} = \phi_i(x_{i,0})$, and $y_{i,0} = \nabla_2 f_i(x_{i,0}, \sigma_{i,0})$ for all $i \in [N]$.
- 2: Iterations: Step $k \ge 0$: update for each $i \in [N]$:

$$x_{i,k+1} = x_{i,k} - \alpha [\nabla_1 f_i(x_{i,k}, \sigma_{i,k}) + \nabla \phi_i(x_{i,k}) y_{i,k}], \quad (4a)$$

$$\sigma_{i,k+1} = \sum_{\substack{j=1\\N}} a_{ij}\sigma_{j,k} + \phi_i(x_{i,k+1}) - \phi_i(x_{i,k}), \tag{4b}$$

$$y_{i,k+1} = \sum_{j=1}^{n} a_{ij} y_{j,k} + \nabla_2 f_i(x_{i,k+1}, \sigma_{i,k+1}) - \nabla_2 f_i(x_{i,k}, \sigma_{i,k}), \quad (4c)$$

In algorithm (4), $\sigma_{i,k}$ is leveraged for agent *i* to track the average (3) since $\sigma(x)$ is global information, which cannot be accessed directly for all agents, and meanwhile, $y_{i,k}$ is introduced for agent *i* to track the gradient sum $\frac{1}{N} \sum_{i=1}^{N} \nabla_2 f_i(x_i, \sigma(x))$, which is also 3

unavailable to all agents. The initial variable $x_{i,0}$ is arbitrary for all $i \in [N]$, and choosing $\sigma_{i,0} = \phi_i(x_{i,0})$ and $y_{i,0} = \nabla_2 f_i(x_{i,0}, \sigma_{i,0})$ for all $i \in [N]$.

The name "distributed gradient tracking" is attributed to the fact that algorithm (4) has combined the classical gradient descent algorithm with the variable tracking techniques.

To proceed, for a vector $x = col(x_1, \ldots, x_N) \in \mathbb{R}^n$, it is helpful to define $\phi(x) := col(\phi_1(x_1), \dots, \phi_N(x_N))$. Also, for a differentiable function $g(x) = col(g_1(x), \ldots, g_m(x))$, where g_i 's are real-valued functions, let us denote by $\nabla g(x) = (\nabla g_1(x), \dots, \nabla g_m(x)).$

With the above notations and those after Example 1, DGT (4) can be written in a compact form

$$x_{k+1} = x_k - \alpha [\nabla_1 f(x_k, \sigma_k) + \nabla \phi(x_k) y_k],$$
(5)

$$\sigma_{k+1} = \mathcal{A}\sigma_k + \phi(x_{k+1}) - \phi(x_k), \tag{6}$$

$$y_{k+1} = \mathcal{A}y_k + \nabla_2 f(x_{k+1}, \sigma_{k+1}) - \nabla_2 f(x_k, \sigma_k),$$
(7)

with $\mathcal{A} = A \otimes I_d$ as defined in Lemma 4, $x_k := col(x_{i,k}, \dots, x_{N,k})$, and similar notations for σ_k and y_k .

Before presenting the main result, it is necessary to first introduce a preliminary result.

Lemma 6. Under Assumption 1, there hold:

$$\bar{\sigma}_k := \frac{1}{N} \sum_{i=1}^N \sigma_{i,k} = \frac{1}{N} \sum_{i=1}^N \phi_i(x_{i,k}),$$
$$\bar{y}_k := \frac{1}{N} \sum_{i=1}^N y_{i,k} = \frac{1}{N} \sum_{i=1}^N \nabla_2 f_i(x_{i,k}, \sigma_{i,k})$$

Proof. In view of (6) and double-stochasticity in Assumption 1, multiplying $\mathbf{1}^{\top}/N$ on both sides of (6) can lead to that

$$\bar{\sigma}_{k+1} = \bar{\sigma}_k + \frac{1}{N} \sum_{i=1}^N \phi_i(x_{i,k+1}) - \frac{1}{N} \sum_{i=1}^N \phi_i(x_{i,k}),$$

which further implies that

$$\bar{\sigma}_k - \frac{1}{N} \sum_{i=1}^N \phi_i(x_{i,k}) = \bar{\sigma}_0 - \frac{1}{N} \sum_{i=1}^N \phi_i(x_{i,0}).$$

Combining the above equality and $\sigma_{i,0} = \phi_i(x_{i,0})$ yields the first assertion of this lemma. Similar arguments can obtain the second one, which completes the proof.

It is now ready to present the main result of this paper.

Theorem 1. Under Assumptions 1 and 2, if

$$0 < \alpha < \min\left\{\frac{1}{L_1}, \alpha_s\right\},\tag{8}$$

where

$$\alpha_s := \frac{\mu (1-\rho)^2}{L_3 L_\mu [(1-\rho)L_0 + 2L_2 L_3]},\tag{9}$$

 $L_{\mu} := \mu + L_1 + L_2 L_3$ and $L_0 := L_1 + L_2 + L_2 L_3$, then $x_k =$ $col(x_{1,k},\ldots,x_{N,k})$ generated by algorithm (4) can converge to the optimizer of problem (2) at a linear convergence rate.

Proof. Let us bound $||x_{k+1} - x^*||$, $||x_{k+1} - x_k||$, $||\sigma_{k+1} - \mathcal{J}\sigma_{k+1}||$, and $||y_{k+1} - \mathcal{J}y_{k+1}||$ in the sequel, where x^* is the optimal variable of problem (2). Denote $\sigma(x^*)$ as σ^* for brevity in this proof.

First, for $||x_{k+1} - x^*||$, invoking (5) yields that

$$\begin{aligned} \|x_{k+1} - x^*\| \\ &= \|x_k - x^* - \alpha [\nabla_1 f(x_k, \sigma_k) + \nabla \phi(x_k) y_k] \| \\ &\leq \|x_k - x^* - \alpha [\nabla_1 f(x_k, \mathbf{1}_N \otimes \bar{\sigma}_k) \\ &+ \nabla \phi(x_k) \mathbf{1}_N \otimes \frac{1}{N} \sum_{i=1}^N \nabla_2 f_i(x_{i,k}, \mathbf{1}_N \otimes \bar{\sigma}_k)] + \alpha \nabla f(x^*) \\ &+ \alpha \|\nabla_1 f(x_k, \sigma_k) + \nabla \phi(x_k) \mathbf{1}_N \otimes \bar{y}_k - \nabla_1 f(x_k, \mathbf{1}_N \otimes \bar{\sigma}_k) \\ &- \nabla \phi(x_k) \mathbf{1}_N \otimes \frac{1}{N} \sum_{i=1}^N \nabla_2 f_i(x_{i,k}, \mathbf{1}_N \otimes \bar{\sigma}_k) \| \\ &+ \alpha \|\nabla \phi(x_k) y_k - \nabla \phi(x_k) \mathbf{1}_N \otimes \bar{y}_k \| \\ &\leq (1 - \mu \alpha) \|x_k - x^*\| + \alpha L_1 \|\sigma_k - \mathbf{1}_N \otimes \bar{\sigma}_k \| \\ &+ \alpha \|\nabla \phi(x_k) \| \|y_k - \mathbf{1}_N \otimes \bar{y}_k \| \\ &\leq (1 - \mu \alpha) \|x_k - x^*\| + \alpha L_1 \|\sigma_k - \mathcal{J}\sigma_k \| \\ &+ \alpha L_3 \|y_k - \mathcal{J}y_k\|, \end{aligned}$$

where Assumption 2.1, Lemma 3, (5) and (8) have been utilized to obtain the second inequality, and the last inequality has applied the fact that $\|\nabla \phi(x_k)\| \leq \max_{i \in [N]} \|\phi_i(x_{i,k})\| \leq L_3$ by Assumption 2.3, $\mathbf{1}_N \otimes \bar{\sigma}_k = \mathcal{J} \sigma_k$, and $\mathbf{1}_N \otimes \bar{y}_k = \mathcal{J} y_k$.

Second, for $||x_{k+1} - x_k||$, by noting that

$$\nabla f(x^*) = \nabla_1 f(x^*, \mathbf{1}_N \otimes \sigma^*) + \nabla \phi(x^*) [\mathbf{1}_N \otimes \frac{1}{N} \sum_{i=1}^N \nabla_2 f_i(x^*, \mathbf{1}_N \otimes \sigma^*)] = \mathbf{0},$$

invoking (5) yields that

$$\begin{aligned} \|x_{k+1} - x_k\| \\ &= \alpha \|\nabla_1 f(x_k, \sigma_k) + \nabla \phi(x_k) y_k\| \\ &\leq \alpha \|\nabla_1 f(x_k, \sigma_k) + \nabla \phi(x_k) \mathcal{J} y_k - \nabla_1 f(x^*, \mathbf{1}_N \otimes \sigma^*) \\ &- \nabla \phi(x^*) [\mathbf{1}_N \otimes \frac{1}{N} \sum_{i=1}^N \nabla_2 f_i(x^*, \mathbf{1}_N \otimes \sigma^*)]\| \\ &+ \alpha \|\nabla \phi(x_k) (y_k - \mathcal{J} y_k)\| \\ &\leq \alpha L_1(\|x_k - x^*\| + \|\sigma_k - \mathbf{1}_N \otimes \sigma^*\|) + \alpha L_3 \|y_k - \mathcal{J} y_k\| \\ &\leq \alpha L_1(\|x_k - x^*\| + \|\sigma_k - \mathcal{J} \sigma_k\|) + \alpha L_3 \|y_k - \mathcal{J} y_k\| \\ &+ \alpha L_1 \|\mathcal{J} \sigma_k - \mathbf{1}_N \otimes \sigma^*\|, \end{aligned}$$
(11)

where Assumption 2.1 and $\|\nabla \phi(x_k)\| \leq L_3$ have been used in the second inequality. For the last term in (11), in view of Lemma 6, one has that

$$\begin{split} \|\mathcal{J}\sigma_{k} - \mathbf{1}_{N} \otimes \sigma^{*}\|^{2} &= \|\mathbf{1}_{N} \otimes (\bar{\sigma}_{k} - \sigma^{*})\|^{2} \\ &= N \|\frac{1}{N} \sum_{i=1}^{N} (\phi_{i}(x_{i,k}) - \phi_{i}(x_{i}^{*}))\|^{2} \\ &\leq \frac{1}{N} (\sum_{i=1}^{N} \|\phi_{i}(x_{i,k}) - \phi_{i}(x_{i}^{*})\|)^{2} \\ &\leq \frac{1}{N} (\sum_{i=1}^{N} L_{3} \|x_{i,k} - x_{i}^{*}\|)^{2} \\ &\leq L_{3}^{2} \sum_{i=1}^{N} \|x_{i,k} - x_{i}^{*}\|^{2} \\ &= L_{3}^{2} \|x_{k} - x^{*}\|^{2}, \end{split}$$

where Assumption 2.3 has been employed in the second inequality, and the last inequality has appealed to the fact that $(\sum_{i=1}^{N} a_i)^2 \leq$

 $N \sum_{i=1}^{N} a_i^2$ for any nonnegative scalars a_i 's. Therefore, combining the above inequality and (11) follows that

$$||x_{k+1} - x_k|| \le \alpha L_1 (1 + L_3) ||x_k - x^*|| + \alpha L_1 ||\sigma_k - \mathcal{J}\sigma_k|| + \alpha L_3 ||y_k - \mathcal{J}y_k||.$$
(12)

Third, regarding $\|\sigma_{k+1} - \mathcal{J}\sigma_{k+1}\|$, by noting that $\mathcal{J}\mathcal{A} = \mathcal{A}\mathcal{J} = \mathcal{J}$, in light of (6), one can obtain that

$$\begin{aligned} \|\sigma_{k+1} - \mathcal{J}\sigma_{k+1}\| \\ &= \|\mathcal{A}\sigma_k + \phi(x_{k+1}) - \phi(x_k) - \mathcal{J}\mathcal{A}\sigma_k \\ &- \mathcal{J}[\phi(x_{k+1}) - \phi(x_k)]\| \\ &\leq \rho \|\sigma_k - \mathcal{J}\sigma_k\| + \|I - \mathcal{J}\| \|\phi(x_{k+1}) - \phi(x_k)\| \\ &\leq \rho \|\sigma_k - \mathcal{J}\sigma_k\| + L_3 \|I - \mathcal{J}\| \|x_{k+1} - x_k\|, \end{aligned}$$
(13)

where Lemma 4 has been leveraged in the first inequality, and Assumption 2.3 has been exploited in the last inequality. By noticing that $||I - \mathcal{J}|| = 1$ and inserting (12) into (13), it can be obtained that

$$\begin{aligned} \|\sigma_{k+1} - \mathcal{J}\sigma_{k+1}\| \\ &\leq (\rho + \alpha L_1 L_3) \|\sigma_k - \mathcal{J}\sigma_k\| + \alpha L_1 L_3 (1 + L_3) \|x_k - x^*\| \\ &+ \alpha L_3^2 \|y_k - \mathcal{J}y_k\|. \end{aligned}$$
(14)

Fourth, for $||y_{k+1} - \mathcal{J}y_{k+1}||$, similar to (13), invoking (7) results in that

$$\begin{aligned} \|y_{k+1} - \mathcal{J}y_{k+1}\| \\ &\leq \rho \|y_k - \mathcal{J}y_k\| + \|\nabla_2 f(x_{k+1}, \sigma_{k+1}) - \nabla_2 f(x_k, \sigma_k)\|. \end{aligned}$$
(15)

At this step, by (6), one has that $\|\sigma_{k+1} - \sigma_k\| = \|(\mathcal{A} - I \otimes I_d)(\sigma_k - \mathcal{J}\sigma_k) + \phi(x_{k+1}) - \phi(x_k)\| \le \|\mathcal{A} - I\| \|\sigma_k - \mathcal{J}\sigma_k\| + L_3 \|x_{k+1} - x_k\|$, where the fact $\mathcal{A}\sigma_k - \sigma_k = (\mathcal{A} - I \otimes I_d)(\sigma_k - \mathcal{J}\sigma_k)$ has been used in the equality, and Assumption 2.3 has been leveraged in the inequality, which together with Assumption 2.2 yields that

$$\begin{aligned} \|\nabla_2 f(x_{k+1}, \sigma_{k+1}) - \nabla_2 f(x_k, \sigma_k)\| \\ &\leq L_3(\|x_{k+1} - x_k\| + \|\sigma_{k+1} - \sigma_k\|) \\ &\leq L_2(1 + L_3)\|x_{k+1} - x_k\| + L_2\|A - I\|\|\sigma_k - \mathcal{J}\sigma_k\|. \end{aligned}$$
(16)

Substituting (12) and (16) into (15) can give rise to

$$\begin{aligned} \|y_{k+1} - \mathcal{J}y_{k+1}\| \\ &\leq (\rho + \alpha L_2 L_3 (1 + L_3)) \|y_k - \mathcal{J}y_k\| \\ &+ \alpha L_1 L_2 (1 + L_3)^2 \|x_k - x^*\| \\ &+ (\alpha L_1 L_2 (1 + L_3) + L_2 \|A - I\|) \|\sigma_k - \mathcal{J}\sigma_k\|. \end{aligned}$$
(17)

Finally, define $\theta_k := col(||x_k - x^*||, ||\sigma_k - \mathcal{J}\sigma_k||, ||y_k - \mathcal{J}y_k||)$. By (10), (14), (17) and $||A - I|| \leq 2$, it can be concluded that

$$\theta_{k+1} \le M(\alpha)\theta_k,\tag{18}$$

where

and

$$M(\alpha) := X + \alpha E, \tag{19}$$

 $X := \begin{pmatrix} 1 & 0 & 0 \\ 0 & \rho & 0 \\ 0 & 2L_2 & \rho \end{pmatrix},$ (20)

$$E := \begin{pmatrix} -\mu & L_1 & L_3 \\ L_1 L_2 (1+L_3) & L_1 L_3 & L_3^2 \\ L_1 L_2 (1+L_3)^2 & L_1 L_2 (1+L_3) & L_2 L_3 (1+L_3) \end{pmatrix}.$$
 (21)

Denote by $\lambda(\alpha)$ the eigenvalues of $M(\alpha)$. It is easy to see that 1 is a simple eigenvalue of M(0), and its corresponding left and right

eigenvectors are both w = col(1, 0, 0). As a result, invoking Lemma 6 leads to that

$$\frac{d\lambda(\alpha)}{d\alpha}\Big|_{\alpha=0} = \frac{w^{\top}Ew}{w^{\top}w} = -\mu < 0, \tag{22}$$

which indicates that the spectral radius of $M(\alpha)$ will be less than 1 for sufficiently small positive α .

One can also see that the graph corresponding to $M(\alpha)$ is strongly connected, which together with Theorem C.3 in [38] implies that $M(\alpha)$ is irreducible. By Lemma 1, $M(\alpha)$ is primitive, which in combination with Lemma 2 can ensure that 1 will be a simple eigenvalue of $M(\alpha)$ when α increases from 0 to some value. By calculating $det(I - M(\alpha)) = 0$, one can obtain that $\alpha = \alpha_s$, where α_s is defined in (9). Therefore, all eigenvalues of $M(\alpha)$ have absolute values less than 1 when $\alpha \in (0, \alpha_s)$, which can guarantee the linear convergence rate of θ_k , thus ending the proof.

Remark 2. It is worth mentioning that, to our best knowledge, this paper is the first to investigate problem (2) in the presence of the aggregative term $\sigma(x)$, for which a linearly convergent distributed algorithm has been developed here.

IV. A NUMERICAL EXAMPLE

This section aims at presenting an optimal placement problem for supporting the designed algorithm. In an optimal placement problem in \mathbb{R}^2 , there are M entities that are located at fixed positions, and meanwhile, there are N free entities, each of which are only privately aware of some of the fixed M entities. The objective is to determine the optimal positions of N free entities in order to minimize the sum of all (square) distances from each free entity to its corresponding fixed entities. For example, the entities can represent warehouses, the links between each free entity and its associated fixed entities as well as the center of all free entities stand for the transportation routes, and the center of all free entities means a goods factory or a central warehouse. In this example, free entities are called agents.



Fig. 1. The communication graph.

For the above problem, let M = N = 5, and each agent *i* is only privately aware of the fixed entity *i*. In this case, the problem can be modeled as (2) by letting

$$f_i(x_i, \sigma(x)) = \gamma_i \|x_i - r_i\|^2 + \|x_i - \sigma(x)\|^2, \quad i \in [N]$$
(23)

where r_i 's are the fixed entities, and $\gamma_i > 0$ represents the weighting between the first and second terms. For the simulation, let ϕ_i be the identity mapping for all $i \in [N]$, $\alpha = 0.05$, $\gamma_i = i$, $r_1 = col(3, 5)$, $r_2 = col(6, 9)$, $r_3 = col(9, 8)$, $r_4 = col(6, 2)$, and $r_5 = col(9, 2)$, and the communication graph is shown in Fig. 1, which is strongly connected.

By randomly selecting the initial positions of agents, i.e., $x_{i,0}$'s, performing the developed DGT algorithm gives rise to evolutions of all $x_{i,k}$'s and $\sigma_{i,k}$'s, as shown in Figs. 2 and 3, respectively, showing that all agents can converge to their optimal positions very fast and the estimate $\sigma_{i,k}$ of each agent can converge to the optimal $\sigma(x^*)$, where $x^* = col(x_1^*, \ldots, x_N^*)$ is the optimal position. Therefore, the simulation results support the theoretical result.



Fig. 2. Evolutions of $x_{i,k}$'s, where squares and circles mean initial positions and final optimal positions of all agents, respectively.



Fig. 3. Evolutions of $\sigma_{i,k}$'s, where squares are the initial positions, and the pentagram means the optimal center, i.e., $\sigma(x^*) = \frac{1}{N} \sum_{i=1}^{N} x_i^*$.

V. CONCLUSION

This paper has proposed and investigated a new framework for distributed optimization, i.e., distributed aggregative optimization, which allows local objective functions to be dependent not only on their own decision variables but also on an aggregative term $\sigma(x)$, relying on decision variables of all other agents. To handle this problem, a distributed algorithm, i.e., DGT, has been developed and rigorously analyzed, where the global objective function is assumed to be strongly convex and smooth along with some Lipschitz property, and the communication graph is assumed to be fixed, balanced, and strongly connected. It has been shown that the algorithm can converge to the optimal variable at a linear rate. A numerical example has been provided to support the theoretical result. Basically, this paper opens up a new avenue to distributed optimization. Future works can be placed on various cases, such as unbalanced graphs, feasible constraint sets, and other interesting forms of objective functions, etc.

REFERENCES

- A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multiagent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.
- [2] J. Chen and A. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, 2012.

- [3] D. Jakovetić, J. Xavier, and J. Moura, "Fast distributed gradient methods," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1131– 1146, 2014.
- [4] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Convergence of asynchronous distributed gradient methods over stochastic networks," *IEEE Transactions* on Automatic Control, vol. 63, no. 2, pp. 434–448, 2018.
- [5] D. Yuan, Y. Hong, D. Ho, and G. Jiang, "Optimal distributed stochastic mirror descent for strongly convex optimization," *Automatica*, vol. 90, pp. 196–203, 2018.
- [6] S. Liang, L. Wang, and G. Yin, "Distributed quasi-monotone subgradient algorithm for nonsmooth convex optimization over directed graphs," *Automatica*, vol. 101, pp. 175–181, 2019.
- [7] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.
- [8] 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.
- [9] D. Jakovetić, J. Moura, and J. Xavier, "Linear convergence rate of a class of distributed augmented Lagrangian algorithms," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 922–936, 2015.
- [10] I. Notarnicola and G. Notarstefano, "Asynchronous distributed optimization via randomized dual proximal gradient," *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2095–2106, 2017.
- [11] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "A Bregman splitting scheme for distributed optimization over networks," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 3809–3824, 2018.
- [12] N. K. Jerinkić, D. Jakovetić, N. Krejić, and D. Bajović, "Distributed second order methods with increasing number of working nodes," *IEEE Transactions on Automatic Control*, in press, doi: 10.1109/TAC.2019.2922191, 2019.
- [13] A. Nedić, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.
- [14] P. Lin, W. Ren, and J. A. Farrell, "Distributed continuous-time optimization: Nonuniform gradient gains, finite-time convergence, and convex constraint set," *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2239–2253, 2017.
- [15] S. Liu, Z. Qiu, and L. Xie, "Convergence rate analysis of distributed optimization with projected subgradient algorithm," *Automatica*, vol. 83, pp. 162–169, 2017.
- [16] Q. Liu, S. Yang, and Y. Hong, "Constrained consensus algorithms with fixed step size for distributed convex optimization over multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 4259–4265, 2017.
- [17] S. Yang, Q. Liu, and J. Wang, "A multi-agent system with a proportionalintegral protocol for distributed constrained optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3461–3467, 2017.
- [18] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2012.
- [19] 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, 2014.
- [20] D. Mateos-Núnez and J. Cortés, "Distributed saddle-point subgradient algorithms with Laplacian averaging," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2720–2735, 2017.
- [21] 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, 2017.
- [22] I. Notarnicola and G. Notarstefano, "A duality-based approach for distributed optimization with coupling constraints," in *Proceedings of International Federation of Automatic Control World Congress*, Toulouse, France, 2017, pp. 14326–14331.
- [23] —, "Constraint-coupled distributed optimization: A relaxation and duality approach," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 1, pp. 483–492, 2020.
- [24] X. Li, X. Yi, and L. Xie, "Distributed online optimization for multiagent networks with coupled inequality constraints," arXiv preprint arXiv:1805.05573, 2018.
- [25] S. Lee and M. M. Zavlanos, "On the sublinear regret of distributed primal-dual algorithms for online constrained optimization," arXiv preprint arXiv:1705.11128, 2017.
- [26] X. Li, G. Feng, and L. Xie, "Distributed proximal algorithms for multiagent optimization with coupled inequality constraints," *IEEE Transac-*

tions on Automatic Control, in press, doi: 10.1109/TAC.2020.2989282, 2020.

- [27] X. Yi, X. Li, L. Xie, and K. H. Johansson, "Distributed online convex optimization with time-varying coupled inequality constraints," *IEEE Transactions on Signal Processing*, vol. 68, no. 1, pp. 731–746, 2020.
- [28] X. Yi, X. Li, T. Yang, L. Xie, K. H. Johansson, and T. Chai, "Distributed bandit online convex optimization with time-varying coupled inequality constraints," arXiv preprint arXiv:1912.03719, 2019.
- [29] X. Cao and K. J. R. Liu, "Distributed Newton's method for network cost minimization," *IEEE Transactions on Automatic Control*, in press, doi: 10.1109/TAC.2020.2989266, 2020.
- [30] X. Li, L. Xie, and Y. Hong, "Distributed continuous-time algorithm for a general nonsmooth monotropic optimization problem," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 10, pp. 3252– 3266, 2019.
- [31] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multi-agent coordination, consensus, and flocking," in *Proceedings of 44th IEEE Conference on Decision and Control*, Seville, Spain, 2005, pp. 2996–3000.
- [32] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33–46, 2007.
- [33] B. Gharesifard and J. Cortés, "Distributed strategies for generating weight-balanced and doubly stochastic digraphs," *European Journal of Control*, vol. 18, no. 6, pp. 539–557, 2012.
- [34] S. Liang, P. Yi, and Y. Hong, "Distributed Nash equilibrium seeking for aggregative games with coupled constraints," *Automatica*, vol. 85, pp. 179–185, 2017.
- [35] C. De Persis and S. Grammatico, "Continuous-time integral dynamics for a class of aggregative games with coupling constraints," *IEEE Transactions on Automatic Control*, in press, doi: 10.1109/TAC.2019.2939639, 2019.
- [36] R. A. Horn and C. R. Johnson, *Matrix Analysis, 2nd ed.* New York, NY: Cambridge University Press, 2012.
- [37] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 315–320, 2018.
- [38] W. Ren and R. W. Beard, Distributed Consensus in Multi-Vehicle Cooperative Control. London, U.K.: Springer-Verlag, 2008.