Jiang, Wei; Charalambous, Themistoklis

Finite-Time Consensus Dual Algorithm for Distributed Optimization over Digraphs

Published in:
IFAC-PapersOnLine

DOI:
10.1016/j.ifacol.2023.10.1083

Published: 01/07/2023

Document Version
Publisher's PDF, also known as Version of record

Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-2 (2023) 1926–1931

Finite-Time Consensus Dual Algorithm for Distributed Optimization over Digraphs

Wei Jiang * **Themistoklis Charalambous** *,**

* *Department of Electrical Engineering and Automation, School of Electrical Engineering, Aalto University, 02150, Espoo, Finland.*
E-mail: wei.jiang@aalto.fi

** *Department of Electrical and Computer Engineering, School of Engineering, University of Cyprus, Nicosia, Cyprus.*
E-mail: charalambous.themistoklis@ucy.ac.cy

Abstract: In this paper, we solve the convex distributed optimization problems, which include unconstrained optimization and a special constrained problem commonly known as a resource allocation problem, over a network of agents among which the communication can be represented by directed graphs (digraphs), by using the finite-time consensus-based and dual-based first-order gradient descent (GD) techniques. The key point is that a special consensus matrix is utilized for problem reformulation to make our dual-based algorithm suitable for digraphs. By the property of distributed finite-time exact (not approximate) consensus, the classical centralized optimization techniques (e.g., Nesterov accelerated GD) can be embedded into our dual-based algorithm conveniently, which means our distributed algorithm can inherit performance of classical centralized algorithms that has been proved to have optimal convergence performance. As a result, our proposed algorithm has faster convergence rate related to the optimization iteration number compared with other distributed optimization algorithms in literature. Since there are finite consensus communication steps inside each consensus process, when the time needed to communicate values between two neighbors is less than a threshold of the time needed to perform local computations, our proposed algorithm is also faster related to the time, as demonstrated in the simulations.

Keywords: Distributed optimization, dual, finite-time consensus, digraphs.

1. INTRODUCTION

With the proliferation of emerging applications such as wireless sensor networks and machine learning, distributed optimization has attracted a lot of attention from the research community. Many algorithms have been proposed based on different assumptions related to the objective functions and network communication graphs. There are mainly two research lines in the literature: *(i)* primal-based, e.g., gradient descent (GD) and *(ii)* dual-based, e.g., alternating direction method of multipliers (ADMM) and dual-based GD.

For primal-based methods, one research line goes from proposing algorithms with sub-linear (e.g., $\mathcal{O}(\frac{\ln k}{k})$ with k being the optimization iteration number) convergence rate, e.g., in Nedic and Ozdaglar (2009); Chen and Ozdaglar (2012); Qu and Li (2018), to linear/geometric [1] convergence rate, e.g., EXTRA in Shi et al. (2015) and DIGing in Nedic et al. (2017).

All aforementioned works are only applicable to undirected or balanced graphs in the convenience of constructing a doubly stochastic matrix [2] for average consensus. Inspired by the push-sum technique (Kempe et al., 2003) for digraphs, researchers integrated push-sum in GD based algorithms for digraphs, e.g., Push-DIGing in Nedic et al. (2017), AB in Xin and Khan (2018) and Push-Pull in Pu et al. (2020). Since the dynamic average consensus technique used in AB/Push-Pull can only converge asymptotically, AB/Push-Pull converge linearly with a sufficiently small step-size. As having the maximum step-size is still an open challenge from (Pu et al., 2020, Remark 5), Jiang and Charalambous (2022) proposed Pull-FTERC algorithm in order to have a larger step-size. However, only a sufficient condition $L < 3\mu$ (L and μ are respectively the smooth and strongly convex parameters (see Definitions 4 and 5)) can be provided to guarantee that the interval of GD step-size is not empty. In other words, if the global objective function is ill-conditioned, $L < 3\mu$ would not hold anymore.

For dual-based methods, related to ADMM-based algorithms, one research line also goes from making algorithms

[1] Suppose that a sequence x^k converges to x^* in some norm $\|\cdot\|$. It is said that the convergence is (i) Q-linear if there exists $\lambda \in (0,1)$ such that $\frac{\|x^{k+1}-x^*\|}{\|x^k-x^*\|} \leq \lambda, \forall k$; (ii) R-linear if there exists $\lambda \in (0,1)$ and some positive constant c such that $\|x^k - x^*\| \leq c\lambda^k, \forall k$. Both of these rates are geometric (Nedic et al., 2017).

[2] A nonnegative matrix is such that all of its elements are nonnegative. A row (column) stochastic matrix is a real square nonnegative matrix of which each row (column) sums to 1. The doubly stochastic matrix is both row and column stochastic.

10.1016/j.ifacol.2023.10.1083

work for undirected and connected graphs (see, for example, Wei and Ozdaglar (2012); Shi et al. (2014); Falsone et al. (2020)) to making them for digraphs, e.g., in Khatana and Salapaka (2020); Jiang and Charalambous (2021); Jiang et al. (2022). For more details about the above algorithms, please refer to Jiang et al. (2022) and references therein. We do not go into details about ADMM-based methods. However, the algorithm performance comparison will be demonstrated in simulations.

This work focuses on the dual-based GD methods for unconstrained and constrained optimization problems. Similarly, previously, researchers, e.g., Scaman et al. (2017), proposed algorithms for undirected graphs. By using the finite-time exact ratio consensus (FTERC) technique from Charalambous et al. (2013, 2015) and constructing a consensus matrix, the dual-based algorithm (we call it Dual-FTERC) proposed in this paper can be applied for digraphs, which is an improvement. Another contribution is that classical *centralized* optimization techniques in literature (e.g., Nesterov accelerated GD) can be embedded into our Dual-FTERC directly which, as a result, makes Dual-FTERC capable of having a large value for GD step-size (compared to other distributed methods which can only have linear convergence rate with a sufficiently small step-size) and converge faster related to the optimization iteration number k.

2. PRELIMINARIES

Notation: the sets of real, integer, and positive integer numbers are denoted as $\mathbb{R}, \mathbb{Z}, \mathbb{Z}_+$, respectively and $\mathbb{R}^n$ denotes the n-dimensional real space. A^{T} and x^{T} are respectively the transpose of matrix A and vector x. $\mathbf{1}$ and I represent respectively the all-ones vector and the identity matrix (of appropriate dimensions). e_j is a column vector of all 0s but a 1 in the jth entry. $\langle a, b\rangle$ denotes the Euclidean inner product $a^{\mathrm{T}}b$. $\|x\|$ denotes the Euclidean norm of a vector x.

Graph theory: in a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ of order n, the set of nodes and edges are $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, respectively. An edge from node v_i to node v_j is represented as $\varepsilon_{ji} = (v_j, v_i) \in \mathcal{E}$ which means node v_j can receive information from node v_i. A graph is undirected if and only if $\varepsilon_{ji} \in \mathcal{E}$ implies $\varepsilon_{ij} \in \mathcal{E}$. A directed graph (digraph) is said to be *strongly* connected if there exists a path from each node v_i to each other node v_j ($v_j \neq v_i$). The graph diameter D is the longest shortest path between any two nodes in the network. Nodes that can send information to node v_j directly are the in-neighbors of node v_j, denoted by $\mathcal{N}_j^- = \{v_i \in \mathcal{V} \mid \varepsilon_{ji} \in \mathcal{E}, i \neq j\}$. Nodes that receive information from node v_j belong to the set of out-neighbors of node v_j and belong to the set $\mathcal{N}_j^+ = \{v_l \in \mathcal{V} \mid \varepsilon_{lj} \in \mathcal{E}, l \neq j\}$. The cardinality of $\mathcal{N}_j^+$, is called the *out-degree* of node v_j and is denoted as $\mathcal{D}_j^+ = |\mathcal{N}_j^+|$.

Ratio Consensus:

Assumption 1. The directed communication graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is strongly connected.

Lemma 1. (Domínguez-García and Hadjicostis (2010)). $\forall v_j \in \mathcal{V}$ under Assumption 1 and $t = 0, 1, 2, \ldots$, y_j^t and x_j^t are the result of the following ratio consensus iterations

$$y_j^{t+1} = p_{jj} y_j^t + \sum_{v_i \in \mathcal{N}_j^-} p_{ji} y_i^t, \tag{1a}$$

$$x_j^{t+1} = p_{jj} x_j^t + \sum_{v_i \in \mathcal{N}_j^-} p_{ji} x_i^t, \tag{1b}$$

where $p_{lj} = \frac{1}{1+\mathcal{D}_j^+}$ for $v_l \in \mathcal{N}_j^+ \cup \{v_j\}$ (zeros otherwise); $y^0 = (y_1^0 \quad y_2^0 \quad \ldots \quad y_n^0)^{\mathrm{T}} \triangleq y_0$ and $x^0 = \mathbf{1}$ are the initial conditions. As a result, the solution to the average consensus problem can be obtained asymptotically as $\lim_{t\to\infty} \mu_j^t = \frac{1}{n} \sum_{v_i \in \mathcal{V}} y_i^0, \forall v_j \in \mathcal{V}$, where $\mu_j^t = y_j^t / x_j^t$.

Finite-Time Exact Ratio Consensus (FTERC): In what follows, Charalambous et al. (2013, 2015) propose the FTERC algorithm in which every node can compute $\mu_j \triangleq \lim_{t\to\infty} \mu_j^t$ in a *minimum* number of iteration steps.

Lemma 2. (Charalambous et al. (2013)). $\forall v_j \in \mathcal{V}$ under Assumption 1 and $t = 0, 1, 2, \ldots$, y_j^t and x_j^t are the result of the iteration (1), where $P = [p_{ji}] \in \mathbb{R}^{n\times n}$ is a primitive column stochastic weight matrix adhere to the graph structure. Then, the solution to the average consensus problem for each node v_j can be distributively obtained in finite-time by

$$\mu_j \triangleq \lim_{t\to\infty} \frac{y_j^t}{x_j^t} = \frac{\phi_y(j)}{\phi_x(j)} = \frac{y_{M_j}^{\mathrm{T}} \boldsymbol{\beta}_j}{x_{M_j}^{\mathrm{T}} \boldsymbol{\beta}_j}, \tag{2}$$

where the details of $\phi_y(j)$, $\phi_x(j)$ and $\boldsymbol{\beta}_j$ (the coefficient vector) can be referred to Section II-C in Jiang and Charalambous (2022).

Distributed FTERC in Networked Systems;

From (2), we know $\boldsymbol{\beta}_j$ and M_j can be different for each node v_j. To implement FTERC for networked systems in a distributed way, all nodes need to know when to terminate ratio consensus (1); also, Assumption 1 and the following assumption are also needed.

Algorithm 1 Distributed FTERC

1: **Initialization:** n' (upper bound on n)
2: **Input:** Node $v_j \in \mathcal{V}$ sets y_j^0 and $k = 0$
3: **if** $k = 0$ **then**
4: Run FTERC for $2n'$ steps to compute y_j^1 and determine M_j and $\boldsymbol{\beta}_j$
5: **else if** $k = 1$ **then**
6: Run max$-$consensus from $M_j + 1, v_j \in \mathcal{V}$ to determine $M_{\max}$; run ratio consensus (1) for n' steps to compute y_j^2 with the same $\boldsymbol{\beta}_j$
7: **else**
8: Run ratio consensus (1) for $t_{\max} \coloneqq M_{\max} + 1$ steps with the same $\boldsymbol{\beta}_j$ to compute y_j^{k+1}
9: **end if**
10: **Output:** Node v_j obtains the inforamtion: $\frac{1}{n} \sum_{v_i \in \mathcal{V}} y_i^0$

Assumption 2. Each node $v_j \in \mathcal{V}$ knows an upper bound on the number of nodes in the network n' (i.e., $n' \geq n$).

Here, we describe Algorithm 1 that we proposed in Jiang and Charalambous (2021) for distributed FTERC termination:

1) When $k = 0$, node v_j computes y_j^1 by using FTERC algorithm which runs for $2n'$ steps. At that moment, it is guaranteed that every node has computed its final value y_j^1, which needs computing $\boldsymbol{\beta}_j$ and as a result, M_j is determined.
2) When $k = 1$, node v_j runs ratio consensus (1) for n' steps and computes y_j^2 with the same $\boldsymbol{\beta}_j$ obtained at $k = 0$ (i.e., there is no need to compute the defective Hankel matrices again). Simultaneously, it runs a max$-$consensus algorithm with the initial condition $u_j^0 = M_j + 1$. Note that the max$-$consensus algorithm converges in s steps ($s \leq D \leq n-1 < n'$). Hence, at the n' step for ratio consensus (1), node v_j not only computes y_j^2, but also the maximum number of steps needed $t_{\max} \coloneqq M_{\max} + 1$ by each node v_j to compute their $y_j^{k+1}, k \geq 2$.
3) When $k \geq 2$, each node v_j computes y_j^{k+1} via ratio consensus (1) for $t_{\max}$ steps with the same $\boldsymbol{\beta}_j$.

Algorithm 1 guarantees that the ratio consensus step at every k, $k \geq 2$ is the *minimum* (see properties of FTERC) and that the solution is precise.

3. MAIN RESULTS

3.1 Dual of unconstrained optimization

In this work, we investigate a networked system with n agents whose objective is to address the following additive cost optimization problem collaboratively over a digraph in a distributed fashion:

$$\min_{y\in\mathbb{R}^p} \sum_{i=1}^{n} f_i(y), \tag{3}$$

where $y \in \mathbb{R}^p$ is a common decision variable and each individual cost $f_i : \mathbb{R}^p \to \mathbb{R}$ is only known to the node v_i.

Assumption 3. The cost function $f_i : \mathbb{R}^p \to \mathbb{R}$ is convex, L_i-smooth and μ_i-strongly convex.

Definition 4. A function $f_i : \mathbb{R}^p \to \mathbb{R}$ is L_i-smooth if f_i is differentiable and its gradient is L_i-Lipschitz continuous, i.e., $\forall x, y \in \mathbb{R}^p$,

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_i \|x - y\|. \tag{4}$$

Definition 5. A function $f_i : \mathbb{R}^p \to \mathbb{R}$ is μ_i-strongly convex if $\forall x, y \in \mathbb{R}^p$,

$$f_i(y) \geq f_i(x) + \langle \nabla f_i(x), y - x\rangle + \frac{\mu_i}{2}\|y - x\|^2. \tag{5}$$

Denote $\mathbf{y} = [y_1^{\mathrm{T}}, \ldots, y_n^{\mathrm{T}}]^{\mathrm{T}}$. Then, it is equal to solve the following problem:

$$\min_{\mathbf{y}\in\mathbb{R}^{np}:\ y_1=\ldots=y_n} F(\mathbf{y}) \coloneqq \sum_{i=1}^{n} f_i(y_i). \tag{6}$$

Denote a matrix

$$\Gamma \coloneqq (I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}) \times I_p. \tag{7}$$

It is easy to see that 0 is a simple eigenvalue of $I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}$ with $\mathbf{1}$ as the corresponding right eigenvector, and 1 is the other eigenvalue with multiplicity $n-1$. Then, it follows in (Li et al., 2013, Theorem 1) that $\Gamma\mathbf{y} = \mathbf{0}$ if and only if $y_1 = y_2 = \ldots = y_n$. As a result, one can regard the matrix Γ as *consensus matrix* and problem (3) is transformed to

$$\min_{\mathbf{y}\in\mathbb{R}^{np}:\ \Gamma\mathbf{y}=\mathbf{0}} F(\mathbf{y}). \tag{8}$$

Remark 1. There are some works on solving the unconstrained problem (3) over undirected graphs, e.g., see Scaman et al. (2017). For the convenience of comparison presentation here, suppose the common decision variable $y \in \mathbb{R}$ as a scalar and thus $\mathbf{y} \in \mathbb{R}^n$ accordingly, the idea behind the above two works is to construct a matrix $\bar{\Gamma}$ satisfying the following conditions: (I) $\bar{\Gamma}$ is an $n \times n$ symmetric matrix; (II) $\bar{\Gamma}$ is positive semi-definite; (III) The kernel of $\bar{\Gamma}$ is the set of constant vectors: $\ker(\bar{\Gamma}) = \mathrm{span}(\mathbf{1})$; (IV) $\bar{\Gamma}$ is defined on the edges of the network: $\bar{\Gamma}_{ij} \neq 0$ only if $i = j$ or $(i, j) \in \mathcal{E}$. In addition, $\bar{\Gamma}$ is used for consensus requirement as $\bar{\Gamma}\mathbf{y} = \mathbf{0}$ by using the one-step or multi-step gossiping algorithm (Boyd et al., 2006). Due to the requirement $\bar{\Gamma}\mathbf{y} = \mathbf{0}$ and the condition (IV) which links $\bar{\Gamma}$ to the graph structure, the graph considered in those papers are undirected. As one can see, by proposing the specific matrix $I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}$ in the consensus matrix Γ in (7), the conditions (I-III) are naturally satisfied and we decouple the design of Γ from the graph structure (i.e., get rid of condition (IV)). In such a way, we can have a freedom to use another matrix (P in Lemma 2) in the FTERC technique to achieve the consensus stage $(I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}})\mathbf{y} = \mathbf{0}$, thus enabling our work for digraphs.

Define the Lagrangian associated with problem (8) as

$$L(y_i, x) = \sum_{i=1}^{n} f_i(y_i) + x^{\mathrm{T}}\Gamma\mathbf{y}, \tag{9}$$

where $x \in \mathbb{R}^{np}$ is the Lagrange multiplier (or dual variable) associated with (8). Denote $\Gamma_i \in \mathbb{R}^{np\times p}$ as a matrix constructed from the $((i-1)p+1)$-th column to the $((i-1)p+p)$-th column of $\Gamma \in \mathbb{R}^{np\times np}$ in (7). Thus, it is easy to derive the Lagrangian dual function as follows:

$$\begin{aligned}\inf_{y_i\in\mathbb{R}^p} L(y_i, x) &= -\sup_{y_i} \sum_{i=1}^{n}((-\Gamma_i^{\mathrm{T}}x)^{\mathrm{T}}y_i - f_i(y_i)) \\ &= -\sum_{i=1}^{n} f_i^*(-\Gamma_i^{\mathrm{T}}x),\end{aligned} \tag{10}$$

where f_i^* is the Legendre-Fenchel *conjugate* [3] of f_i. Since f_i is convex from Assumption 3 and there is no inequality constraint in problem (8), *Slater's condition* holds (Boyd et al., 2004, Section 5.2.3) which means strong duality holds from Slater's theory, i.e., the duality gap is zero. As a result, the Lagrangian dual of problem (8) becomes

$$\max_{x\in\mathbb{R}^{np}} -F^* \coloneqq -\sum_{i=1}^{n} f_i^*(-\Gamma_i^{\mathrm{T}}x). \tag{11}$$

Then, the Lagrangian dual (11) can be changed to

$$\min_{x\in\mathbb{R}^{np}} F^* \coloneqq \sum_{i=1}^{n} f_i^*(-\Gamma_i^{\mathrm{T}}x). \tag{12}$$

From the construction of Γ in (7), one can see each node needs to know the network size n. Similar as Assumption 1, there are distributed methods to compute the network size; see, e.g., Shames et al. (2012).

3.2 Constrained optimization problem transformation

The following type of problem is considered:

[3] Let $\phi(y) : \mathbb{R}^p \to \mathbb{R}$. The function $\phi^* : \mathbb{R}^p \to \mathbb{R}$ defined as $\phi^*(x) = \sup_{y\in\mathbb{R}^p}(x^{\mathrm{T}}y - \phi(y))$ is called the conjugate of the function $\phi(y)$.

$$\min_{z_i\in\mathbb{R}^{n_i}} \sum_{i=1}^{n} \phi_i(z_i), \quad \text{s.t.} \sum_{i=1}^{n}(A_i z_i - b_i) = 0, \tag{13}$$

where $z_i \in \mathbb{R}^{n_i}$ is the common decision variable and each individual cost $\phi_i : \mathbb{R}^{n_i} \to \mathbb{R}$ is only known to the node v_i and is convex, closed and proper, $A_i \in \mathbb{R}^{p\times n_i}$ and $b_i \in \mathbb{R}^p$. This kind of problem has many applications in reality, e.g., distributed resource allocation in Banjac et al. (2019); Jiang et al. (2022).

The objective in this subsection is to transform the constrained optimization problem (13) into the unconstrained one (3). To achieve that, similarly, by using the Lagrangian dual, its corresponding Lagrangian with the dual variable $y \in \mathbb{R}^p$ is $L(z_i, y) = \sum_{i=1}^{n} \phi_i(z_i) + y^{\mathrm{T}} \sum_{i=1}^{n}(A_i z_i - b_i)$. Similar as (10), the dual function is

$$f(y) = \inf_{z_i} L(z_i, y) = -\sum_{i=1}^{n}(\phi_i^*(-A_i^{\mathrm{T}} y) + y^{\mathrm{T}} b_i).$$

Also similar as Section 3.1, one can check the duality gap is zero. Consequently, the dual problem is to maximize the dual function as

$$\max_{y\in\mathbb{R}^p} -\sum_{i=1}^{n} \underbrace{(\phi_i^*(-A_i^{\mathrm{T}} y) + y^{\mathrm{T}} b_i)}_{=:f_i(y)}. \tag{14}$$

Then, problem (14) is equal to the unconstrained problem (3) with the specific f_i in the above.

Now, we need to check the smoothness and strongly convexity of $f_i(y)$ in (14). Assume ϕ_i is strongly convex and smooth, as a result, $\phi_i^*(-A_i^{\mathrm{T}} y) + y^{\mathrm{T}} b_i$ is strongly convex and smooth (Becker et al., 2011, Section 2.3).

Next, we will propose algorithms focusing on solving problem (3), which goes down to solve problem (12).

3.3 Dual-FTERC algorithm

First, we recall the centralized GD method for unconstrained minimization of a function $f(x)$ as

$$x^{k+1} = x^k - \alpha \nabla f(x^k), \tag{15}$$

where α is a fixed step-size parameter. When $f(x)$ is L-smooth (not necessarily convex) and $\alpha \in (0, 1/L)$, then the sequence $\{x^k\}$ converges to a minimizer $x^\star$ at linear rate (Xin et al., 2020).

Similarly, in distributed setting, for each node v_i with the function $f_i^*(-\Gamma_i^{\mathrm{T}} x_i)$ in (12), if each node updates as centralized GD (15) as $x_i^{k+1} = x_i^k - \alpha \nabla f_i^*(-\Gamma_i^{\mathrm{T}} x_i^k)$ [4], then, when $x_i^k = x^\star, \forall i$ and for some k, we have

$$x_i^{k+1} = x^\star - \alpha \nabla f_i^*(-\Gamma_i^{\mathrm{T}} x^\star) \neq x^\star, \tag{16}$$

because $\nabla f_i^*(-\Gamma_i^{\mathrm{T}} x^\star) \neq \mathbf{0}$, i.e., the minimizer $x^\star$ of the global cost function $\sum_{i=1}^{n} f_i^*(-\Gamma_i^{\mathrm{T}} x)$ dose not necessarily minimize the local functions $f_i^*(-\Gamma_i^{\mathrm{T}} x)$ (Xin et al., 2020). Therefore, to solve problem (12), if $\nabla f_i^*(-\Gamma_i^{\mathrm{T}} x_i^k)$ in (16) is replaced by $\sum_{i=1}^{n} \nabla f_i^*(-\Gamma_i^{\mathrm{T}} x_i^k)$, then the issue disappears. Next, we will show that this strategy is possible. We propose Algorithm 2 to solve problem (12) as follows:

1) In Step 7 of Algorithm 2, node v_i runs distributed FTERC Algorithm 1$(\nabla f_i^*(-\Gamma_i^{\mathrm{T}} x_i^{k+1}), k)$. Then, after finite iteration steps, v_i will get $\frac{1}{n}\sum_{j=1}^{n} \nabla f_j^*(-\Gamma_j^{\mathrm{T}} x_j^{k+1})$.
2) As a result, in Step 5 of Algorithm 2, for node v_i, the centralized GD method (15) or its accelerated versions can be adopted for update.

Algorithm 2 Dual-FTERC

1: **Initialization:** n (network size), Γ (consensus matrix), α_k (step-size), β_k (momentum parameter), $k_{\max}$ (maximum number of iterations)
2: **Input:** Node $v_i \in \mathcal{V}$ sets s_i^0, x_i^0 and $k = 0$
3: Node v_i does the following:
4: **while** $k \leq k_{\max}$ **do**
5: $x_i^{k+1} \leftarrow$ Centralized (accelerated) GD algorithms $\leftarrow (x_i^k, \alpha_k, (\beta_k), ns_i^k)$
6: Calculate $\nabla f_i^*(-\Gamma_i^{\mathrm{T}} x_i^{k+1})$
7: Put $(\nabla f_i^*(-\Gamma_i^{\mathrm{T}} x_i^{k+1}), k)$ as input to distributed FTERC Algorithm 1 and get output s_i^{k+1}
8: $k \leftarrow k + 1$
9: **end while**
10: **Output:** Node $v_i \in \mathcal{V}$ obtains the dual solution: $x^\star$; then, recovers the optimal solution $y^\star$ by calculating $\nabla f_i^*(-\Gamma_i^{\mathrm{T}} x^\star)$

Recall that Algorithm 2 is used to solve problem (3) distributively. By using the finite-time characteristic and exact average property of our distributed FTERC Algorithm 1, traditional accelerated GD algorithms can be adopted perfectly. For accelerated GD algorithms, in literature, e.g., there are Nesterov accelerated gradient (NAG) (Nesterov, 2004) and its variants, such as Sutskever Nesterov Momentum and heavy-ball momentum. For example, for NAG, Step 5 of Algorithm 2 becomes

$$\begin{aligned} r_i^{k+1} &= x_i^k - \alpha_k n s_i^k, \\ x_i^{k+1} &= r_i^{k+1} + \beta_k (r_i^{k+1} - r_i^k), k \geq 0, \end{aligned} \tag{17}$$

where the initial condition for variable r_i can be $r_i^0 = x_i^0$ and β_k is the momentum parameter.

To prove the convergence of Algorithm 2, for example, we take NAG (17) as the accelerated GD algorithm in Step 5. Before presenting our convergence proof, denote $\mu = \min_{i=1}^{n} \mu_i, L = \max_{i=1}^{n} L_i$. Then, $\sum_{i=1}^{n} f_i(y)$ in (3) is μ-strongly convex and L-smooth (Ghadimi et al., 2013). As a result, $\sum_{i=1}^{n} f_i^*(-\Gamma_i^{\mathrm{T}} x)$ in (12) is $\frac{1}{L}$-strongly convex and $\frac{1}{\mu}$-smooth (Becker et al., 2011, Section 2.3). Denote $F^{\star\star}$ as the minimal value of problem (12).

Theorem 1. Under Assumptions 1, 2 and 3, for Dual-FTERC Algorithm 2 with NAG (17) used in Step 5, by setting $\alpha_k = \mu$ and $\beta_k = \frac{1-\sqrt{\mu/L}}{1+\sqrt{\mu/L}}$, F^* converges linearly as

$$F^{*k} - F^{\star\star} \leq \frac{1}{\mu}(1 - \sqrt{\frac{\mu}{L}})^k \|x^0 - x^\star\|^2. \tag{18}$$

With this convergence rate one can achieve an accuracy of ϵ in $\mathcal{O}(\sqrt{\frac{L}{\mu}} \log \frac{1}{\epsilon})$.

Proof. In Step 7 of Algorithm 2, by using Algorithm 1, we have $ns_i^{k+1} = \sum_{j=1}^{n} \nabla f_j^*(-\Gamma_j^{\mathrm{T}} x_j^{k+1}), \forall i$ in finite-time, which means, Step 5 of Algorithm 2 equals to classical NAG method (Nesterov, 2004). Thus, by setting $\alpha_k = \mu$ and $\beta_k = \frac{1-\sqrt{\mu/L}}{1+\sqrt{\mu/L}}$, the convergence rate (18) is achieved

[4] If $f(y)$ is closed and strongly convex then its Lagrangian dual f^* is differentiable with gradient as $\nabla f^*(x) = \arg\min_y (f(y) - x^{\mathrm{T}} y)$.

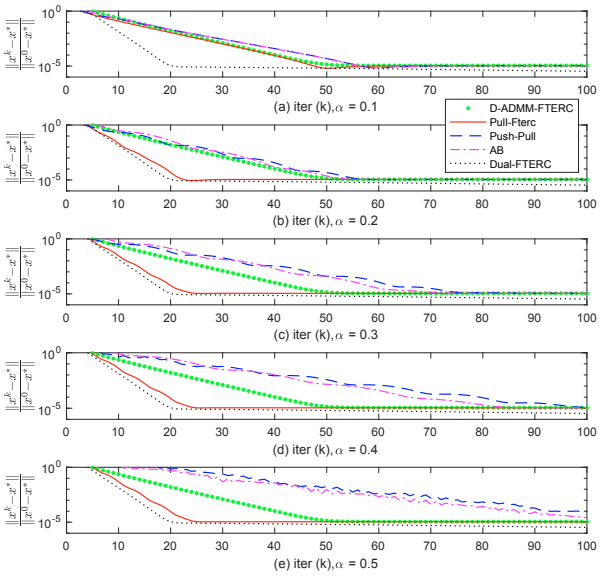


Fig. 1. Convergence rate (related to the iteration number k) performance of (normalized) residuals from different algorithms, i.e., D-ADMM-FTERC, Pull-FTERC, Push-Pull, AB with different values of step-size α and Dual-FTERC with $\alpha_k = 1.0198, \beta_k = 0.3134$.

and is optimal in the sense of the lower complexity bounds (O'donoghue and Candes, 2015, Section 2).

After the optimal solution $x^\star$ to the Lagrangian dual problem (12) is obtained in Dual-FTERC Algorithm 2, each node can recover the optimal solution $y^\star$ to problem (3) by calculating $\nabla f_i^*(-\Gamma_i^{\mathrm{T}} x^\star)$ in Step 6 of Algorithm 2.

4. EXAMPLES

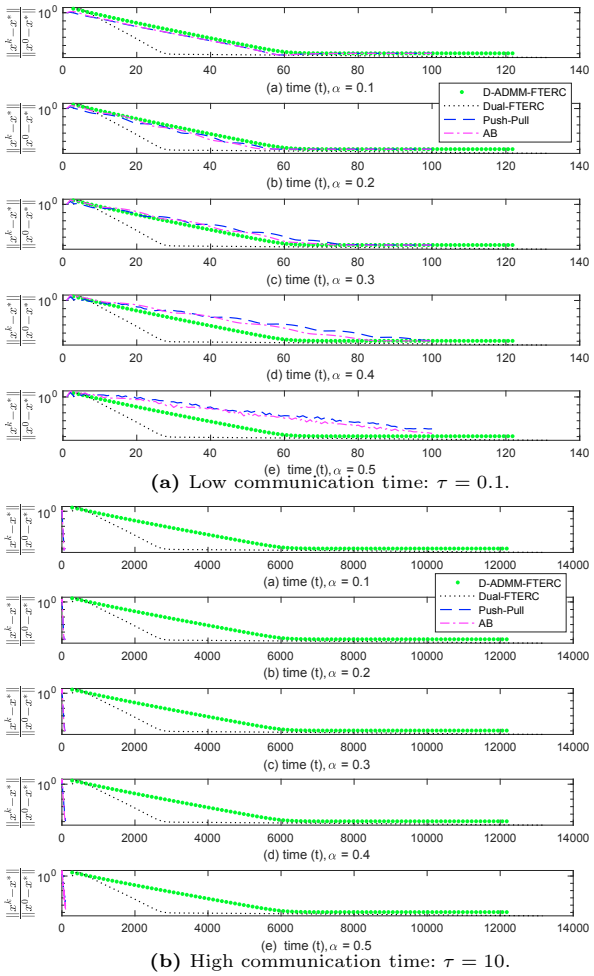
The distributed least squares problem is considered as

$$\min_{y\in\mathbb{R}^p} \sum_{i=1}^{n} \frac{1}{2}\|A_i y - b_i\|^2, \tag{19}$$

where $n = 6, A_i \in \mathbb{R}^{q\times p}$ is only known to node v_i, $b_i \in \mathbb{R}^q$ is the measured data and $y \in \mathbb{R}^p$ is the common decision variable that needs to be optimized. In this example, we take the same setting as in Jiang and Charalambous (2022), except that we add our newly proposed Dual-FTERC for comparison with algorithms: D-ADMM-FTERC in Jiang and Charalambous (2021), Pull-FTERC in Jiang and Charalambous (2022), Push-Pull in Pu et al. (2020) and AB in Xin and Khan (2018). For detailed settings of these algorithms, please refer to Jiang and Charalambous (2022). For Dual-FTERC, based on Theorem 1, we have $\alpha_k = 1.0198, \beta_k = 0.3134$.

Fig. 1 shows the convergence rate performance comparison. One can see Dual-FTERC always outperforms other algorithms. It is worth noting that it is not convincing to compare the convergence rate between FTERC-based algorithms and Push-Pull/AB directly in Fig. 1. The reason is that there are multiple consensus steps inside each FTERC iteration step (means multiple communication rounds which consume more time) while only one consensus step inside each Push-Pull/AB iteration step. Specifically, FTERC stage consists of $2n' = 28\,(k = 0), n' = 14\,(k = 1)$ and $t_{\max} = 13\,(k \geq 2)$ communication steps inside each Dual-FTERC iteration. Inspired by Scaman et al. (2017), we denote τ (resp. 1) is the time needed to communicate values between two neighbors (resp. perform local computations). More specifically, we borrow from (Scaman et al., 2017, Sec. 2.1) that assume:

1) Each computing unit can compute first-order characteristics, such as the gradient of its own function. By renormalization of the time axis, and without loss of generality, we assume that this computation is performed in one unit of time.
2) Each computing unit can communicate values (i.e. vectors in $\mathbb{R}^p$) to its neighbors. This communication requires time τ (may be smaller or greater than 1).



(a) Low communication time: $\tau = 0.1$.

(b) High communication time: $\tau = 10$.

Fig. 2. Convergence (related to time) performance of (normalized) residuals with different values of step-size α with $\alpha_k = 1.0198, \beta_k = 0.3134$.

Fig. 2 shows the performance of the compared algorithms. Fig. 2a shows that when the communication time is smaller than the computation time ($\tau \ll 1$), Dual-FTERC always have a better performance. though it performs multiple communication rounds per FTERC iteration. When $\tau \gg 1$, Dual-FTERC is less efficiency in time compared with Push-Pull and AB as shown in Fig. 2b.

5. CONCLUSIONS

A distributed finite-time consensus based dual gradient descent algorithm is proposed to solve the additive cost optimization problem over a digraph. Compared to the newest algorithms in the literature, the proposed one

has a faster convergence rate related to the optimization iteration number. When the time needed to communicate values between two neighbors is less than a threshold of the time needed to perform local computations, the proposed one is also faster related to the consumed time.

Future work will target non-convex objective functions and considering more (possible non-convex) constraints such as additional inequality constraints. Also, quantized optimization is very useful for decreasing communication burdens. Consensus-based optimization methods with a single gradient tracking step are also considered.

REFERENCES

Banjac, G., Rey, F., Goulart, P., and Lygeros, J. (2019). Decentralized resource allocation via dual consensus ADMM. In *American Control Conference*, 2789–2794. IEEE.

Becker, S.R., Candès, E.J., and Grant, M.C. (2011). Templates for convex cone problems with applications to sparse signal recovery. *Mathematical programming computation*, 3(3), 165–218.

Boyd, S., Boyd, S.P., and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2006). Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6), 2508–2530.

Charalambous, T., Yuan, Y., Yang, T., Pan, W., Hadjicostis, C.N., and Johansson, M. (2013). Decentralised minimum-time average consensus in digraphs. In *52nd IEEE Conference on Decision and Control*, 2617–2622.

Charalambous, T., Yuan, Y., Yang, T., Pan, W., Hadjicostis, C.N., and Johansson, M. (2015). Distributed finite-time average consensus in digraphs in the presence of time delays. *IEEE Transactions on Control of Network Systems*, 2(4), 370–381.

Chen, A.I. and Ozdaglar, A. (2012). A fast distributed proximal-gradient method. In *50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 601–608.

Domínguez-García, A.D. and Hadjicostis, C.N. (2010). Coordination and control of distributed energy resources for provision of ancillary services. In *IEEE International Conference on Smart Grid Communications*, 537–542.

Falsone, A., Notarnicola, I., Notarstefano, G., and Prandini, M. (2020). Tracking-ADMM for distributed constraint-coupled optimization. *Automatica*, 117, 108962.

Ghadimi, E., Shames, I., and Johansson, M. (2013). Multi-step gradient methods for networked optimization. *IEEE Transactions on Signal Processing*, 61(21), 5417–5429.

Jiang, W. and Charalambous, T. (2021). Distributed alternating direction method of multipliers using finite-time exact ratio consensus in digraphs. In *European Control Conference*, 2205–2212.

Jiang, W. and Charalambous, T. (2022). A fast finite-time consensus based gradient method for distributed optimization over digraphs. In *IEEE 61st Conference on Decision and Control (CDC)*, 6848–6854. IEEE.

Jiang, W., Doostmohammadian, M., and Charalambous, T. (2022). Distributed resource allocation via admm over digraphs. In *IEEE 61st Conference on Decision and Control (CDC)*, 5645–5651. IEEE.

Kempe, D., Dobra, A., and Gehrke, J. (2003). Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science*, 482–491.

Khatana, V. and Salapaka, M.V. (2020). D-DistADMM: A $O(1/k)$ distributed ADMM for distributed optimization in directed graph topologies. In *59th IEEE Conference on Decision and Control*, 2992–2997.

Li, Z., Ren, W., Liu, X., and Fu, M. (2013). Consensus of multi-agent systems with general linear and lipschitz nonlinear dynamics using distributed adaptive protocols. *IEEE Transactions on Automatic Control*, 58(7), 1786–1791.

Nedic, A., Olshevsky, A., and Shi, W. (2017). Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4), 2597–2633.

Nedic, A. and Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1), 48–61.

Nesterov, Y. (2004). *Introductory lectures on convex optimization: A basic course*, volume 87. Kluwer Academic, Dordrecht.

O'donoghue, B. and Candes, E. (2015). Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15(3), 715–732.

Pu, S., Shi, W., Xu, J., and Nedić, A. (2020). Push–pull gradient methods for distributed optimization in networks. *IEEE Transactions on Automatic Control*, 66(1), 1–16.

Qu, G. and Li, N. (2018). Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3), 1245–1260.

Scaman, K., Bach, F., Bubeck, S., Lee, Y.T., and Massoulié, L. (2017). Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *International Conference on Machine Learning*, 3027–3036. PMLR.

Shames, I., Charalambous, T., Hadjicostis, C., and Johansson, M. (2012). Distributed network size estimation and average degree estimation and control in networks isomorphic to directed graphs. In *50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 1885–1892.

Shi, W., Ling, Q., Yuan, K., Wu, G., and Yin, W. (2014). On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 62(7), 1750–1761.

Shi, W., Ling, Q., Wu, G., and Yin, W. (2015). Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2), 944–966.

Wei, E. and Ozdaglar, A. (2012). Distributed alternating direction method of multipliers. In *51st IEEE Conference on Decision and Control*, 5445–5450.

Xin, R. and Khan, U.A. (2018). A linear algorithm for optimization over directed graphs with geometric convergence. *IEEE Control Systems Letters*, 2(3), 315–320.

Xin, R., Pu, S., Nedić, A., and Khan, U.A. (2020). A general framework for decentralized optimization with first-order methods. *Proceedings of the IEEE*, 108(11), 1869–1889.