Robust Online Learning over Networks

Nicola Bastianello*, *Member, IEEE*, Diego Deplano*, *Member, IEEE*,
Mauro Franceschelli, *Senior, IEEE*, and Karl H. Johansson, *Fellow, IEEE*

***Abstract*— The recent deployment of multi-agent networks has enabled the distributed solution of learning problems, where agents cooperate to train a global model without sharing their local, private data. This work specifically targets some prevalent challenges inherent to distributed learning: (i) online training, i.e., the local data change over time; (ii) asynchronous agent computations; (iii) unreliable and limited communications; and (iv) inexact local computations. To tackle these challenges, we apply the Distributed Operator Theoretical (DOT) version of the Alternating Direction Method of Multipliers (ADMM), which we call "DOT-ADMM". We prove that if the DOT-ADMM operator is metric subregular, then it converges with a linear rate for a large class of (not necessarily strongly) convex learning problems toward a bounded neighborhood of the optimal time-varying solution, and characterize how such neighborhood depends on (i)–(iv). We first derive an easy-to-verify condition for ensuring the metric subregularity of an operator, followed by tutorial examples on linear and logistic regression problems. We corroborate the theoretical analysis with numerical simulations comparing DOT-ADMM with other state-of-the-art algorithms, showing that only the proposed algorithm exhibits robustness to (i)–(iv).**

***Index Terms*— Distributed learning, online learning, asynchronous networks, unreliable communications.**

I. Introduction

In recent years, significant technological advancements have enabled the deployment of multi-agent systems across various domains, including robotics, power grids, and traffic networks [1, 2]. These systems consist of interconnected agents that leverage their computational and communication capabilities to collaborate in performing assigned tasks. Many of these tasks – such as estimation [3, 4], coordination and control [5–7], resilient operation [8–10], and learning [11–13] – can be formulated as *distributed optimization problems*, see [2, 14–16] for some recent surveys. In this context, this work focuses specifically on distributed optimization algorithms for learning under network constraints.

Traditional machine learning methods require transmitting all the data collected by the agents to a single location, where they are processed to train a model. However, communicating raw data exposes agents to privacy breaches, which is inadmissible in many applications such as healthcare and industry [17]. Moreover, it is often the case that the agents collect new data over time, which requires another round of data transmission and, as a result, both an increased privacy vulnerability and the need of repeating the centralized training task. In the alternative distributed approach, the agents within the network first process their data to compute an approximate local model, and then refine such model by sharing it with their peers and agreeing upon a common model that better fits all the distributed data sets, with the goal of improving the overall accuracy. This strategy, however, poses some practical challenges – discussed in the next section – both at the computation level, such as computing with different speed and precision, and at the communication level, such as loss and corruption of packets.

The focus of this paper is thus on solving online learning problems in a distributed way while addressing these practical challenges – discussed in detail in Section I-A – by applying a distributed operator theoretical (DOT) version of the alternating direction method of multipliers (ADMM), which we call "DOT-ADMM". In principle, other different approaches developed in the distributed optimization literature can be applied to this problem, which are reviewed in Section I-B. Section I-C outlines the main technical contributions of the manuscript, regarding the linear convergence of DOT-ADMM for (not necessarily strongly) convex problems based on novel theoretical results on stochastic and metric subregular operators.

A. Practical challenges in online learning over networks

1) Asynchronous agents computations: The agents cooperating for the solution of a learning problem are oftentimes highly heterogeneous, in particular in terms of their computational resources [17]; consequently, the agents may perform local computations at different rates. The simple solution of synchronizing the network by enforcing that all agents terminate the k-th local computation at iteration k entails

* Nicola Bastianello and Diego Deplano are co-first authors as they contributed equally. Nicola Bastianello is the corresponding author.

The work of N. Bastianello and K. H. Johansson was partially supported by the European Union's Horizon Research and Innovation Actions programme under grant agreement No. 101070162, and partially by Swedish Research Council Distinguished Professor Grant 2017-01078 Knut and Alice Wallenberg Foundation Wallenberg Scholar Grant.

The work of D. Deplano was supported by the project e.INS- Ecosystem of Innovation for Next Generation Sardinia (cod. ECS 00000038) funded by the Italian Ministry for Research and Education (MUR) under the National Recovery and Resilience Plan (NRRP) - MISSION 4 COMPONENT 2, "From research to business" INVESTMENT 1.5, "Creation and strengthening of Ecosystems of innovation" and construction of "Territorial R& D Leaders".

N. Bastianello and K. H. Johansson are with the School of Electrical Engineering and Computer Science, and Digital Futures, KTH Royal Institute of Technology, Stockholm, Sweden. Emails: {nicolba,kallej}@kth.se

D. Deplano and M. Franceschelli are with DIEE, University of Cagliari, 09123 Cagliari, Italy. Emails: {diego.deplano,mauro.franceschelli}@unica.it

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/ DOI: 10.1109/TAC.2024.3441723

that better-performing agents must wait for the slower ones (cf. [18, Fig. 2]). Therefore, in this paper we allow the agents to perform local processing at their own pace, which is a more efficient strategy than enforcing synchronization.

2) Unreliable communications: In real-world scenarios, the agents have at their disposal imperfect channels to deliver the locally processed models to their peers. One problem that may occur, particularly when relying on wireless communications, is that transmissions from one agent to another may be lost in transit (e.g., due to interference [12]). When a transmission is lost, the newly processed local model of an agent is not delivered to its neighboring agents, but the algorithm needs to be robust to this occurrence. By "robust", we refer to the ability of the algorithm to mitigate the effect of the packet-losses without taking any specific action. Indeed, the design of specific procedures to address packet-losses may require some additional knowledge at the transmitter level, which is unfeasible in some applications. A second problem that must be faced, especially when the local models stored by the agents are high dimensional, is the impracticability of sharing the exact local model over limited channels (e.g., when training a deep neural network). To satisfy limited communications constraints, different approaches have been explored, foremost of which is *quantization/compression* of the messages exchanged by the agents [13]. But quantizing a communication implies that an inexact version of the local models is shared by the agents, which can be seen as a disturbance in the communication. Therefore, we consider both packet-loss and packet-corruption during the communications between agents, which allow us to deal with the two above-mentioned problems.

3) Inexact local computations: In learning applications, the local training performed by the agents may depend on large data-sets, and thus be computationally demanding. This is especially relevant in online set-ups, where training needs to be completed within the interval of time $[k, k+1)$. To solve this issue, so-called *stochastic gradients* are employed, which construct an approximation of the local gradients using a limited number of data points [19]. This implies that every time a stochastic gradient is applied by an agent, some error is introduced in the algorithm. The algorithm we propose in this manuscript needs to compute the proximal of the local cost function, where the proximal operator finds the point that minimizes a function while also being close to its argument. However, unless the cost is proximable [20] and there is a closed form, the proximal needs to be computed via an iterative scheme, such as gradient descent. But again due to the limited computational time $[k, k+1)$ allowed to the agent, the proximal can be computed only with a limited number of iterations, introducing an approximation. The issue may be further compounded by the use of stochastic gradients instead of full gradients.

B. Review of the state-of-the-art

Many algorithms in the state-of-the-art for solving optimization and learning problems are built upon (sub)gradient methods [2]. Despite their effectiveness, these methods are limited to achieving, at best, sub-linear convergence, necessitating the adoption of diminishing step-sizes, even in the case of strongly convex problems. Another class of suitable algorithms is that of gradient tracking, which can achieve convergence with the use of fixed step-sizes [21]. On the one hand, different gradient tracking methods have been proposed for application in an online learning context, see *e.g.* [22, 23]. On the other, the use of robust average consensus techniques has also enabled the deployment of gradient tracking for learning under the constraints of Section I-A, see *e.g.* [24–28]. However, gradient tracking algorithms may suffer from a lack of robustness to some of these challenges [29].

Our approach falls in a different branch of research, which is based on the alternating direction method of multipliers (ADMM). ADMM-based algorithms have turned out to be reliable and versatile for distributed optimization [11, 30]. In particular, ADMM has been shown to be robust to asynchronous computations [30–32], packet losses [33], and both [34]. Additionally, its convergence with inexact communications has been analyzed in [35], and the impact of inexact local computations has been studied in [36]. Moreover, the convergence of ADMM-based algorithms under network constraints has been usually shown to occur at a sub-linear rate, whereas a linear rate can be proved only under additional assumptions, such as strong convexity [34]).

Instead, the algorithm we propose is shown to converge with a linear rate without the strong convexity assumption, but under a milder set of assumptions, while facing at the same time all the challenges described in Section I-A.

C. Main contributions

DOT-ADMM has the following set of features:

- Convergence with a linear rate for a wide class of learning problems (e.g., linear and logistic regression problems);
- Applicability in an online scenario where the data sets available to the agents change over time;
- Robustness to asynchronous and inexact computations of the agents;
- Robustness to faulty and noisy communications.

The main theoretical results of the paper are as follows:

- Time-varying stochastic operators that are averaged and metric subregular operators converge linearly (in mean) and almost surely to a neighborhood of the time-varying set of fixed points without assuming that there exists a common fixed point (see Theorem 3).
- DOT-ADMM is proved to converge for a large class of online learning problems with (not necessarily strongly) convex local costs under the challenging scenario described in Section I-A by relying on the metric subregularity property of the DOT-ADMM operator (see Theorem 1). Complementary results are provided for simpler scenarios where some of the challenges do not come into play (see Corollary 1) and for the case in which metric subregularity holds in a subset of the state space (see Theorem 2).
- An easy-to-verify sufficient condition to ensure metric subregularity of an operator is provided in Proposition

Licensed under CC BY 4.0. DOI: 10.1109/TAC.2024.3441723

1, which basically requires the operator to be bounded by two affine operators. This result is then applied to standard learning problems, such as linear regression (see Proposition 2), robust linear regression (see Proposition 2), and logistic regression (see Proposition 4). These results can be also used as tutorial examples for other learning problems with different regression models.
- Extensive numerical simulations of DOT-ADMM together with a comparison with other state-of-the-art algorithms are provided in Section V, revealing the outperforming performance of DOT-ADMM in terms of convergence time and resilience to the challenging scenario considered in the manuscript.

D. Outline

Section II provides the notation used throughout the paper, gives useful preliminaries on graph theory and operator theory, and formalizes the online optimization problem of interest together with some technical working assumptions. In Section III we formalize the technical challenges usually faced when solving online learning problems, and we present DOT-ADMM as a suitable protocol to be implemented in networks to solve these problems in a distributed manner while facing all the challenges. Section III-C is devoted to the proof of the convergence result anticipated in Section III, which relies on a novel foundational result on stochastic operators that are metric subregular. Section IV outlines how the proposed algorithm can be applied to different learning problems, with a focus on linear and logistic regression problems. In Section V several numerical results are carried out, and Section VI gives some concluding remarks.

II. Preliminaries and Problem Formulation

The set of real and integer numbers are denoted by $\mathbb{R}$ and $\mathbb{Z}$, respectively, while $\mathbb{R}_+$ and $\mathbb{N}$ denote their restriction to positive entries. Matrices are denoted by uppercase letters, vectors and scalars by lowercase letters, while sets and spaces are denoted by uppercase calligraphic letters. The identity matrix is denoted by I_n, $n \in \mathbb{N}$, while the vectors of ones and zeros are denoted by $\mathbb{1}_n$ and $\mathbb{0}_n$; subscripts are omitted if clear from the context. Maximum and minimum of an n-element vector $u = [u_1, \dots, u_n]^\top$, are denoted by $\overline{u} = \max_{i=1,\dots,n} u_i$ and $\underline{u} = \min_{i=1,\dots,n} u_i$, respectively.

A. Preliminaries

1) Networks and graphs: We consider networks modeled by undirected graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, n\}$, $n \in \mathbb{N}$, is the set of *nodes*, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of *edges* connecting the nodes. An undirected graph $\mathcal{G}$ is said to be *connected* if there exists a sequence of consecutive edges between any pair of nodes $i, j \in \mathcal{V}$. Nodes i and j are *neighbors* if there exists an edge $(i, j) \in \mathcal{E}$. The set of neighbors of node i is denoted by $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. For the sake of simplicity, we consider graphs with self-loops, i.e., $i \in \mathcal{N}_i$, and denote by $\eta_i = |\mathcal{N}_i|$ the number of neighbors and by $\xi = 2|\mathcal{E}| = \eta_1 + \cdots + \eta_n$ twice the number of undirected edges in the network.

2) Operator theory: We introduce some key notions from operator theory in finite-dimensional Euclidean spaces, i.e., vector spaces $\mathbb{R}^n$ with $\|\cdot\|$ and distance d

$$\|x\| = \sqrt{x^\top x}, \qquad d(x, y) = \|x - y\|.$$

Operators $\mathsf{F} : \mathbb{R}^n \to \mathbb{R}^n$ are denoted with block capital letters. An operator is *affine* if there exist a matrix $A \in \mathbb{R}^{n \times n}$ and a vector $b \in \mathbb{R}^n$ such that $\mathsf{F} : x \mapsto Ax + b$, and *linear* if $b = \mathbb{0}$. The linear operator associated to the identity matrix I is defined by $\mathsf{Id} : x \mapsto Ix$. Given $\mathsf{F} : \mathbb{R}^n \to \mathbb{R}^n$, $\mathrm{fix}(\mathsf{F}) = \{x \in \mathbb{R}^n : \mathsf{F}(x) = x\}$ denotes its set of *fixed points*. By further defining the *projection operator* of a point x over a non-empty set $\mathcal{X}$ as

$$\mathrm{proj}_{\mathcal{X}}(x) = \operatorname*{arginf}_{y \in \mathcal{X}} \|x - y\|,$$

the distance of point x from the set $\mathcal{X}$ is denoted by

$$d_{\mathcal{X}}(x) = \|x - \mathrm{proj}_{\mathcal{X}}(x)\|$$

When $\mathcal{X}$ is the set of fixed points of an operator F, we use the shorthand notations $d_{\mathsf{F}} := d_{\mathrm{fix}(\mathsf{F})}$ and $\mathrm{proj}_{\mathsf{F}} := \mathrm{proj}_{\mathrm{fix}(\mathsf{F})}$. With this notation, we now define some properties of operators, which are pivotal in this paper.

Definition 1. *An operator $\mathsf{F} : \mathbb{R}^n \to \mathbb{R}^n$ is metric subregular if there is a positive constant $\gamma > 0$ such that*

$$d_{\mathsf{F}}(x) \leq \gamma \|(\mathsf{Id} - \mathsf{F})x\|, \qquad \forall x \in \mathbb{R}^n. \tag{1}$$

When γ is known, F is said to be γ-metric subregular.

Definition 2. *An operator $\mathsf{F} : \mathbb{R}^n \to \mathbb{R}^n$ is nonexpansive if*

$$\|\mathsf{F}(x) - \mathsf{F}(y)\| \leq \|x - y\|, \qquad \forall x, y \in \mathbb{R}^n.$$

Moreover, the operator $\mathsf{G} := (1 - \alpha)\mathsf{Id} + \alpha\mathsf{F}$ with $\alpha \in (0, 1)$ is α-averaged if F is nonexpansive, or, equivalently, if

$$\|\mathsf{G}(x) - \mathsf{G}(y)\|^2 \leq \|x - y\|^2 - \frac{1 - \alpha}{\alpha} \|(\mathsf{Id} - \mathsf{G})(x) - (\mathsf{Id} - \mathsf{G})(y)\|^2.$$

A function $f : \mathcal{X} \to [-\infty, +\infty]$ with $\mathcal{X} \subseteq \mathbb{R}^n$ is: *proper* if its domain $\{x \in \mathcal{X} \mid f(x) < +\infty\}$ is not empty and $-\infty \notin f(\mathcal{X})$ [37, Definition 1.4]; *lower semicontinuous* if its epigraph $\{(x, t) \in \mathcal{X} \times \mathbb{R} \mid f(x) \leq t\}$ with $\mathcal{X} \subseteq \mathbb{R}^n$ is closed in $\mathbb{R}^n \times \mathbb{R}$ [37, Lemma 1.24]; *convex* if its epigraph is a convex subset of $\mathcal{X} \times \mathbb{R}$ [37, Definition 8.1]. Let Γ_0^n be the set of proper, lower semicontinuous, convex functions f from $\mathcal{X} \subseteq \mathbb{R}^n$ to $\mathbb{R} \cup \{+\infty\}$. Then, the proximal operator of $f \in \Gamma_0^n$ is defined by

$$\mathrm{prox}_f^\rho(y) = \operatorname*{argmin}_x \left\{ f(x) + \frac{1}{2\rho} \|x - y\|^2 \right\},$$

where $\rho > 0$ is a penalty parameter; if $\rho = 1$, it is omitted. We note that the projection operator is a particular case of the proximal operator when applied to the indicator function $\iota : \mathbb{R}^n \to \mathbb{R}^n \cup \{+\infty\}$ defined as $\iota_{\mathcal{X}}(x) = 0$ if $x \in \mathcal{X}$, and $\iota_{\mathcal{X}}(x) = +\infty$ otherwise.

Licensed under CC BY 4.0. DOI: 10.1109/TAC.2024.3441723

B. Problem formulation

We consider a network of n agents linked according to an undirected, connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each agent $i \in \mathcal{V}$ has a vector state $x_i(k) \in \mathbb{R}^p$ with $p \in \mathbb{N}$, and has access to a *time-varying* local cost $f_{i,k} : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$, $k \in \mathbb{N}$. The objective of the network is to solve the optimization problem $\min_x \sum_{i\in\mathcal{V}} f_{i,k}(x)$, which can be reformulated in the following distributed form:

$$\begin{aligned} &\min_{x_i} \sum_{i\in\mathcal{V}} f_{i,k}(x_i) \\ &\text{s.t.} \quad x_i = x_j \text{ if } (i,j) \in \mathcal{E} \end{aligned} \tag{2}$$

whose set of solutions is denoted by $\mathcal{X}_k^\star$. We make the following two standing assumptions, which are standard assumptions in online optimization [38].

Assumption 1. *At each time $k \in \mathbb{N}$, the local cost functions $f_{i,k}$ of problem (2) are proper, lower semi-continuous, and convex, i.e., $f_{i,k} \in \Gamma_0^p$.*

Assumption 2. *The set of solutions $\mathcal{X}_k^\star$ to problem (2) is non-empty at each time $k \in \mathbb{N}$. In addition, the minimum distance between two solutions at consecutive times is upper-bounded by a nonnegative constant $\sigma \geq 0$, i.e.,*

$$\sup_{k\in\mathbb{N}} \inf_{x_k^\star \in \mathcal{X}_k^\star} d_{\mathcal{X}_{k-1}^\star}(x_k^\star) \leq \sigma.$$

Remark 1. *If one only considers Assumption 1, the set of solutions $\mathcal{X}_k^\star$ may be empty [37, Proposition 11.15]; thus, Assumption 2 requires that $\mathcal{X}_k^\star \neq \emptyset$. On the other hand, the set of solutions $\mathcal{X}_k^\star$ may also contain infinitely many solutions and be unbounded; thus, Assumption 2 requires that there exists an upper bound $\sigma \geq 0$ (that holds uniformly over time) to the distance between any solution $x_k^\star \in \mathcal{X}_k^\star$ and its projection onto the set of solutions at the previous step, i.e., $\mathcal{X}_{k-1}^\star$.*

The dynamic nature of the problem implies that the solution – in general – cannot be reached exactly, but rather that the agents' states will reach a neighborhood of it [38, 39]. Our goal is thus to quantify how closely the agents can track the optimal solution over networks characterized by the following challenging conditions, as discussed in Section I-A:

- asynchronous agents computations;
- unreliable communications;
- inexact local computations.

The next section introduces the proposed algorithm along with the formal description of the above challenges and the main working assumptions.

III. Proposed algorithm and convergence results

To solve the problem in eq. (2), we employ the Distributed Operator Theoretical (DOT) version of the Alternating Direction Method of Multipliers (ADMM), which we call "DOT-ADMM". It is derived by applying the relaxed Peaceman-Rachford splitting method to the dual of problem in eq. (2) (see [34] and references therein), and its distributed implementation is detailed in Algorithm 1.

Algorithm 1 Distributed Operator Theoretical (DOT) ADMM

Input: For each agent $i \in \mathcal{V}$ initialize the auxiliary variables $\{z_{ij}(0)\}_{j\in\mathcal{N}_i}$; choose the relaxation $\alpha \in (0,1)$ and the penalty $\rho > 0$.

Output: Each agent returns $x_i(k)$ that is an (approximated) solution to the distributed optimization problem in eq. (2).

for $k = 1, 2, \ldots$ **each active agent** $i \in \mathcal{V}$

// asynchronous computations and inexact updates

receives a local cost $f_{i,k}$ and applies the local update

$$x_i(k) = \mathsf{F}_{i,k}(z(k-1)) := \operatorname{prox}_{f_{i,k}}^{1/\rho\eta_i}\left(\frac{1}{\rho\eta_i}\sum_{j\in\mathcal{N}_i} z_{ij}(k-1)\right) \tag{3}$$

for each agent $j \in \mathcal{N}_i$

transmits the packet

$$y_{i\to j}(k) = 2\rho x_i(k) - z_{ij}(k-1)$$

end for

for each packet $y_{j\to i}$ **received by agent** $j \in \mathcal{N}_i$

// noisy communications with packet loss

updates the auxiliary variable

$$z_{ij}(k) = \mathsf{T}_{ij,k}(z(k-1)) := (1-\alpha) z_{ij}(k-1) + \alpha y_{j\to i}(k) \tag{4}$$

end for

end for

Each agent $i \in \mathcal{V}$ first updates its local state $x_i \in \mathbb{R}^p$ according to eq. (3), then sends some information to each neighbor $j \in \mathcal{N}_i$ within the packet $y_{i\to j}$. The agents are assumed to be heterogeneous in their computation capabilities, therefore at each time step only some of the agents are ready for the communication phase. In turn, after receiving the information from its neighbors, each agent updates its auxiliary state variables $z_{ij} \in \mathbb{R}^p$ with $j \in \mathcal{N}_i$ according to eq. (4). Note that the local update in eq. (3) depends only on information available to agent i, while for the auxiliary update in eq. (4) the agent needs to first receive the aggregate information $y_{j\to i}(k)$ from the neighbor j.

Algorithm 1 also makes explicit where the sources of stochasticity discussed in Section I-A come into play: asynchronous agents' computations and packet-loss prevent the updates in eqs. (3)-(4) to be performed at each time $k \in \mathbb{N}$, whereas inexact local computations and noisy communications make these updates inaccurate. We model these sources of stochasticity by means of the following random variables:

- $\beta_{ij}(k) \sim \text{Ber}(p_{ij})$ are Bernoulli random variables[1] with $p_{ij} \in (0,1]$ modeling the asynchronous agents' computations and packet-loss: p_{ij} denotes the probability that agent j has completed its computation and packet $y_{j\to i}$ successfully arrives to agent i;
- $u_i(k)$ and $v_{ij}(k)$ are i.i.d. random variables which represent, respectively, the additive error modeling inexact local updates of x_i and noisy transmission of $y_{j\to i}(k)$ sent by node j to node i.

With these definitions in place, we define the perturbed

[1] i.e., $\beta_{ij}(k) = 1$ with probability p_{ij}, $\beta_{ij}(k) = 0$ with probability $1 - p_{ij}$.

Licensed under CC BY 4.0. DOI: 10.1109/TAC.2024.3441723

variables as follows

$$\widetilde{x}_i(k) = x_i(k) + u_i(k)$$
$$\widetilde{z}_{ij}(k) = z_{ij}(k) + \alpha v_{ij}(k)$$

One can further notice that the additive error $u_i(k)$ on $x_i(k)$ is also an additive error on $y_{j\to i}$ (scaled by a factor equal to 2ρ). Therefore, one can consider only one source of error $e_{ij} = v_{ij}(k) + 2\rho u_i(k)$ and write the perturbed updates as

$$x_i(k) = \mathsf{F}_{i,k}(\widetilde{z}(k-1)) \tag{5a}$$
$$\widetilde{z}_{ij}(k) = \begin{cases} \mathsf{T}_{ij,k}(\widetilde{z}(k-1)) + \alpha e_{ij}(k) & \text{if } \beta_{ij}(k) = 1 \\ \widetilde{z}_{ij}(k-1) & \text{otherwise} \end{cases} \tag{5b}$$

where the operators $\mathsf{F}_{i,k}$, $\mathsf{T}_{ij,k}$ are those of eqs. (3)-(4).

With this notation, we formalize next the challenging assumptions under which the problem in eq. (2) must be solved.

Assumption 3. *At each time step, each node $j \in \mathcal{V}$ has completed its local computation and successfully transmits data to its neighbors $i \in \mathcal{N}_j$ with probability $p_{ij} \in (0, 1]$.*

The minimum and maximum among the probabilities of Assumption 3 are denoted by

$$\underline{p} = \min_{(i,j)\in\mathcal{E}} p_{ij}, \qquad \overline{p} = \max_{(i,j)\in\mathcal{E}} p_{ij}. \tag{6}$$

Assumption 4. *Each node $i \in \mathcal{V}$ updates its local variable $x_i(k)$ with an additive error $u_i(k) \in \mathbb{R}^p$ and receives information from neighbor $j \in \mathcal{N}_i$ with an additive noise $v_{ij}(k) \in \mathbb{R}^p$ such that the overall error $e_{ij}(k) = v_{ij}(k) + 2\rho u_i(k)$ on the update of the auxiliary variable $z_{ij}(k)$ is bounded by $\mathbb{E}[\|e_{ij}(k)\|] \le \nu_e < \infty$.*

A. Convergence results

For the convenience of the reader, we state next our main results, while we postpone their proofs to Section III-C. We begin with the following theorem, which characterizes the mean linear convergence of DOT-ADMM in the stochastic scenario described in Section I-A and formalized in Section III.

Theorem 1 (Linear convergence). *Consider the online distributed optimization in problem (2) under Assumptions 1-2, and a connected network of agents that solves it by running DOT-ADMM under Assumptions 3-4. If the DOT-ADMM operator T_k, defined block-wise by $\mathsf{T}_{ij,k}$ as in eq. (4), is γ-metric subregular, then:*

i) there is an upper bound to the distance between the current solution $x(k)$ and the set of optimal solutions $\mathcal{X}_k^\star$ that holds in mean for all $k \in \mathbb{N}$, and this bound has a linearly decaying dependence on the initial condition:

$$\mathbb{E}\left[d_{\mathcal{X}_k^\star}(x(k))\right] = O\left(\mu^k d_{\mathsf{T}_0}(x(0)) + \frac{1-\mu^k}{1-\mu}(\nu_e + \sigma)\right) \tag{7}$$

where the rate of convergence $\mu \in (0, 1)$ is given in (9).

ii) there is an upper bound for $d_{\mathcal{X}_k^\star}(x(k))$ that holds almost surely when $k \to \infty$, and this bound does not depend on the initial condition:

$$\limsup_{k\to+\infty} d_{\mathcal{X}_k^\star}(x(k)) = O\left(\frac{\nu_e + \sigma}{1-\mu}\right). \tag{8}$$

The following corollary makes explicit how the results of Theorem 1 become stronger when some challenges are not considered.

Corollary 1 (Particular cases). *Consider the scenario of Theorem 1 and the following simplified scenarios:*

(a) the cost functions $f_{i,k} = f_i$ are static, i.e., $\sigma = 0$;
(b) communications are noiseless and computations are exact, i.e., there are no additive errors $\nu_e = 0$;
(c) communications are synchronous.

Then the results of Theorem 1 become:

i) (a) implies that the distance $d_{\mathcal{X}_k^\star}(x(k))$ converges linearly to $O(\nu_e)$ in mean;
ii) (b) implies that the distance $d_{\mathcal{X}_k}(x(k))$ converges linearly to $O(\sigma)$ in mean;
iii) (a) $\wedge$ (b) implies that the distance $d_{\mathcal{X}_k}(x(k))$ converges linearly to zero in mean square with rate μ^2; moreover, $x(k)$ almost surely converges to the set of solutions $\mathcal{X}^\star$;
iv) (a) $\wedge$ (b) $\wedge$ (c) implies that $x(k)$ converges linearly with rate μ^2 and almost surely to the set of solutions $\mathcal{X}^\star$.

Building upon Theorem 1, we also prove that linear convergence holds for strongly convex and smooth costs as the iterative solution approaches a neighborhood of the optimal solutions. This result, which encompasses that of [34], is termed as *eventual linear convergence*.

Theorem 2 (Eventual linear convergence). *Consider the scenario of Theorem 1, when the cost functions $f_{i,k} = f_i$ are static and there are no additive errors $\nu_e = 0$. If the costs are strongly convex and twice continuously differentiable, then there is a finite time $k^\star \in \mathbb{N}$ such that, for any initial condition:*

i) (global) for $k \le k^\star$, the distance $d_{\mathcal{X}_k}(x(k))$ decays sublinearly in mean square;
ii) (local) for $k > k^\star$, the distance $d_{\mathcal{X}_k}(x(k))$ converges linearly to zero in mean square and $x(k)$ almost surely converges to the set of solutions $\mathcal{X}^\star$ for $k \to \infty$.

B. Discussion of the results

1) Convergence rate and error bounds: The value of the convergence rate $\mu \in (0, 1)$, resulting from the punctual upper bound to the tracking error in eq. (7) provided by Theorem 1, is

$$\mu = \sqrt{1 - \frac{(1-\alpha)\underline{p}}{\alpha\lambda}}, \quad \lambda > \max\left\{\gamma^2, \frac{(1-\alpha)\underline{p}}{\alpha}\right\}. \tag{9}$$

where $\underline{p} \in (0, 1]$ is the minimum probability as in eq. (6) that any node completes both the local computation and the transmission tasks, $\alpha \in (0, 1)$ is the relaxation parameter of DOT-ADMM and $\gamma > 0$ is the metric-subregularity constant of the operator T_k ruling the iterations of DOT-ADMM. The presence of random updates leads to a worse convergence rate compared to the convergence rate μ_s attained when all coordinates update at each iteration ($\underline{p} = 1$), indeed, $\mu_s \le \mu$. This is in line with the results proved in [40], and makes intuitive sense since less frequent updates (smaller values of $\underline{p}$) lead to slower convergence (higher values of μ).

On the other hand, it is possible to make the convergence arbitrarily faster by selecting higher values of the relaxation

Licensed under CC BY 4.0. DOI: 10.1109/TAC.2024.3441723

constant α, which, however, worsen the asymptotic error bound in eq. (8): therefore, α constitutes a trade-off between the convergence rate and the asymptotic error. Another important role is played by the additive noise (through ν_e) and by the time-variability of the costs (through σ), which prevent DOT-ADMM from converging to the optimal solution by introducing a non-zero term in both the punctual and asymptotic upper bounds: when these non-idealities are not considered, then one recovers exact convergence as pointed out in Corollary 1. We also remark that the scaling constants hidden by the $O(\cdot)$ notation depend on the specific structure of the network (through the number of edges $|\mathcal{E}|$).

2) Simplified scenarios and mean square convergence: The results of Corollary 1 particularize Theorem 1 for simplified scenarios in which some of the challenges faced in this work are not taken into account. Removing the time-variability of the costs ($\sigma = 0$) implies that the solution provided by DOT-ADMM converges asymptotically within an error from the optimal solution that is bounded by only ν_e; instead, removing the sources of error ($\nu_e = 0$) makes the error bounded by only σ. When both these challenges are not considered ($\sigma = \nu_e = 0$) then exact convergence to the set of optimal solutions can be achieved. Additionally, the linear convergence of DOT-ADMM holds not only in mean but also in mean square. This is a remarkable result since it holds even though the problem is not strongly convex and regardless of the challenging time-varying, unreliable, and asynchronous scenario (see Section III-B.3).

3) Comparison with [34] and strongly convex problems: The result of Theorem 2 clarifies the advantage of the metric-subregularity property against strong convexity of the problem. Indeed, Theorem 1 proves that metric subregularity of the DOT-ADMM operator is sufficient for linear convergence in convex optimization problems. Theorem 2 proves that strong convexity of the problem (under the additional assumption of twice differentiability of the costs) is sufficient for local linear convergence after a finite time $k^\star$, a behavior that we termed *eventual linear convergence*. The linear and logistic regression learning problems discussed in Section IV constitute examples of problems that are not strongly convex but for which DOT-ADMM converges linearly because the updates are ruled by a metric subregular operator. Additionally, sub-linear convergence can be experienced for some strongly convex problems; an example is given by the scalar regularized costs $f_i(x_i) = \sqrt[3]{x_i^4} + \frac{\epsilon}{2}x_i^2$. Therefore, strong convexity is neither necessary nor sufficient for global linear convergence; instead, metric subregularity of the DOT-ADMM operator is sufficient for convex problems due to Theorem 1.

Theorem 2 also shows how the analysis in this paper subsumes that of [34]. In particular, in the case of strongly convex and twice differentiable costs, DOT-ADMM converges sub-linearly until a sufficiently small neighborhood of the unique optimal solution $x^\star$ is reached. Thereafter, the costs are well approximated by a quadratic function around $x^\star$ that, in turn, implies metric subregularity of the DOT-ADMM operator T_k, and thus linear convergence follows by Theorem 1, finding as a special case the result of [34]. Thus, one of the main differences between this paper's contribution w.r.t. [34] is that [34] requires strong convexity and twice differentiability of the costs to ensure *local* linear convergence, while this paper only requires metric subregularity of the DOT-ADMM operator (which does not imply strong convexity of the problem) and derive a *global* linear convergence result. In addition, while DOT-ADMM indeed follows the blueprint of [34], it differs in that it allows for local updates to be inexact and the local costs to be time-varying.

C. Proofs of the results

The proofs of Theorem 1 and Theorem 2 make use of the following general convergence result for stochastic operators enjoying metric-subregularity, which is another original contribution of this manuscript.

Theorem 3. *Let $\widetilde{\mathsf{T}}_k^e : \mathbb{R}^m \to \mathbb{R}^m$ be a time-varying operator defined component-wise by*

$$\widetilde{\mathsf{T}}_{\ell,k}^e(z) := \begin{cases} \mathsf{T}_{\ell,k}(z) + e_{\ell,k} & \text{if } \beta_{\ell,k} = 1 \\ z_\ell & \text{otherwise} \end{cases} \tag{10}$$

for $\ell = 1, \dots, m$, where $e_{\ell,k}$ are i.i.d. random variables and $\beta_{\ell,k} \sim \text{Ber}(p_\ell)$ are Bernoulli i.i.d. random variables such that $p_\ell \in (0, 1]$. If at each time $k \in \mathbb{N}$ it holds:

i) $\exists \varsigma > 0$ such that $\|\text{proj}_{\mathsf{T}_k}(z) - \text{proj}_{\mathsf{T}_{k-1}}(z)\| \leq \varsigma$, $\forall z \in \mathbb{R}^m$;
ii) T_k is α-averaged;
iii) T_k is γ-metric subregular;

then the iteration $z(k) = \widetilde{\mathsf{T}}_k^e(z(k-1))$ converges linearly in mean according to

$$\mathbb{E}[d_{\mathsf{T}_k}(z(k))] \leq \sqrt{\frac{\overline{p}}{\underline{p}}} \left[\mu^k d_{\mathsf{T}_0}(z(0)) + \sum_{h=1}^{k} \mu^{k-h} (\mathbb{E}[\|e_h\|] + \mu\varsigma) \right]$$

where the convergence rate $\mu \in (0, 1)$ is given in eq. (9) and where $\overline{p} = \max_\ell p_\ell$ and $\underline{p} = \min_\ell p_\ell$ are the maximum and minimum error probabilities. Moreover, it almost surely holds that the iteration asymptotically converges to

$$\limsup_{k\to\infty} d_{\mathsf{T}_k}(z(k)) \leq \lim_{k\to\infty} \sqrt{\frac{\overline{p}}{\underline{p}}} \sum_{h=1}^{k} \mu^{k-h} (\mathbb{E}[\|e_h\|] + \mu\varsigma).$$

Proof: See Appendix A. ■

Remark 2. *The results of Theorem 3 are stronger than most of the literature in that they provide a punctual upper bound on the distance to the set of optimal solutions together with a linear convergence rate, in contrast to other state-of-the-art results, such as those in [40, 41]. Results in [40, 41] only rely on the averagedness property to prove sub-linear convergence of the iteration, where [40] does not provide a punctual upper bound to the error but only an asymptotic upper bound and where [41] provides both using a regret-style metric. In contrast, our Theorem 3 exploits the additional metric-subregularity property to prove linear convergence by using the distance from the set of fixed points as a metric. The results of Theorem 3 are also much more practical and can be exploited in more realistic scenarios. Firstly, we do not assume that the influence of additive errors vanishes over time as in [40], instead, we allow for persistent errors due, for instance, to the computational limitations of the*

Licensed under CC BY 4.0. DOI: 10.1109/TAC.2024.3441723

agents. Secondly, we only assume that the fixed point sets at two consecutive iterations are "similar enough" and not necessarily overlapping as assumed in [40].

We also provide a result to cover the case in which metric subregularity does not hold in the entire state space, but only in a subspace of it: this property is called locally metric subregularity.

Theorem 4. *In the scenario of Theorem 3, if it holds:*

i) $\mathsf{T}_k = \mathsf{T}$ *is not time-varying, i.e.,* $\varsigma = 0$*;*
ii) T *is* α*-averaged;*
iii) T *is metric subregular in a set* $\mathcal{X} \subset \mathbb{R}^n$*;*
iv) $\lim_{k\to\infty} \|e_k\| \to 0$*;*

then it almost surely holds $\limsup_{k\to\infty} d_{\mathsf{T}}(z(k)) = 0$*. Moreover, there is a finite time* $k^\star$ *such that for* $k \geq k^\star$ *linear convergence in mean is achieved with rate* μ *in eq.* (9)*.*

Proof: See Appendix B. ∎

Before proceeding with the proofs of our main results, let us conveniently rewrite the operators of the DOT-ADMM updates in eq. (5) in compact form as follows[2]

$$\begin{aligned} x(k) &= \mathsf{F}_k(z(k-1)) = \operatorname{prox}_{f_k}^{1/\rho\eta}(DA^\top z(k-1)) \\ z(k) &= \mathsf{T}_k(z(k-1)) = [(1-\alpha)I - \alpha P]z(k-1) + 2\alpha\rho PAx(k) \end{aligned} \tag{11}$$

where the operator $\operatorname{prox}_{f_k}^{1/\rho\eta} : \mathbb{R}^{np} \to \mathbb{R}^{np}$ applies blockwise the proximal of the time-varying local costs $f_{i,k}$; the matrix $A \in \mathbb{R}^{\xi p \times np}$ is given[3] by $A = \Lambda \otimes I_p$ with $\Lambda \in \{0,1\}^{\xi\times n}$ given by $\Lambda = \operatorname{blk\,diag}\{\mathbb{1}_{\eta_i}\}_{i=1}^n$; the matrix $D \in \mathbb{R}^{np\times np}$ is given by $D = \operatorname{blk\,diag}\{(\rho\eta_i)^{-1} I_p\}_{i=1}^n$; the matrix $P \in \{0,1\}^{\xi p \times \xi p}$ is given by $P = \Pi \otimes I_p$ with $\Pi \in \{0,1\}^{\xi\times\xi}$ being a permutation matrix swapping $(i,j) \in \mathcal{E}$ with $(j,i) \in \mathcal{E}$.

1) Proof of Theorem 1: By [34, Proposition 3], for each fixed point $z_k^\star \in \operatorname{fix}(\mathsf{T}_k)$ there is $x_k^\star = \mathsf{F}_k(z_k^\star)$ which is a solution to the problem in eq. (2). Thus, letting $\mathcal{X}_k^\star$ be the time-vaying set of solutions, we can write

$$\begin{aligned} d_{\mathcal{X}_k^\star}(x(k)) &= \inf_{y\in\mathcal{X}_k^\star} \|x(k) - y\| \overset{(i)}{\leq} \|x(k) - x_k^\star\| \\ &= \|\mathsf{F}_k(z(k-1)) - \mathsf{F}_k(z_k^\star)\| \\ &= \left\|\operatorname{prox}_{f_k}^{1/\rho\eta}(DA^\top z(k-1)) - \operatorname{prox}_{f_k}^{1/\rho\eta}(DA^\top z_k^\star)\right\| \\ &\overset{(ii)}{\leq} \left\|DA^\top(z(k-1) - z_k^\star)\right\| \leq \left\|DA^\top\right\| \|z(k-1) - z_k^\star\| \\ &\overset{(iii)}{=} \left\|DA^\top\right\| d_{\mathsf{T}_k}(z(k-1)) \end{aligned}$$

where (i) holds since $x_k^\star \in \mathcal{X}_k^\star$, (ii) follows by the nonexpansiveness of the proximal, and (iii) holds by choosing

$$z_k^\star = \operatorname*{arginf}_{y\in\operatorname{fix}(\mathsf{T}_k)} \|\!|z(k-1) - y|\!\|.$$

This means that the linear convergence of $x(k)$ to a neighborhood of $\mathcal{X}_k^\star$ is implied by that of $z(k)$ to a neighborhood of $\operatorname{fix}(\mathsf{T}_k)$, which can be proved by means of Theorem 3. Indeed, by Assumptions 3-4 the update of $z(k)$ can be described by a stochastically perturbed operator as in eq. (10) object of Theorem 3. We thus prove both statements of the theorem by checking all the conditions under which Theorem 3 holds:

i) By definition, the fixed points of DOT-ADMM are

$$\operatorname{fix}(\mathsf{T}_k) = \{z \mid (I+P)z = 2\rho PAx_k^\star,\ x_k^\star \in \mathcal{X}_k^\star\}.$$

Therefore the projection of a point z onto $\operatorname{fix}(\mathsf{T}_k)$ is [20, section 6.2.2]

$$\operatorname{proj}_{\mathsf{T}_k}(z) = z - (I+P)^\dagger\left((I+P)z - 2\rho PAx_k^\star\right).$$

With some simple algebra, we can then see that

$$\begin{aligned} \left\|\operatorname{proj}_{\mathsf{T}_k}(z) - \operatorname{proj}_{\mathsf{T}_{k-1}}(z)\right\| &= 2\rho\left\|(I+P)^\dagger PA(x_k^\star - x_{k-1}^\star)\right\| \\ &\overset{(i)}{\leq} 2\rho\left\|(I+P)^\dagger PA\right\|\sigma \end{aligned}$$

where (i) follows by sub-multiplicativity of the norm and Assumption 2. Thus assumption *i)* of Theorem 3 is verified for $\varsigma = 2\rho\left\|(I+P)^\dagger PA\right\|\sigma$.

ii) By Assumption 1 it follows that T_k are α-averaged for all $k \in \mathbb{N}$. Indeed, the operator T_k comes from the application of the Peaceman-Rachford operator to the dual problem of (2), which guarantees its α-averagedness when the cost functions $f_{i,k}$ are convex (cfr. [34]);

iii) T_k is γ-metric subregular by assumption.

2) Proof of Theorem 2: Since the local costs are strongly convex and twice differentiable, one can approximate the local costs by quadratic functions of the following kind

$$\begin{aligned} f_i(x) = \frac{1}{2}(x - x^\star)^\top \nabla^2 f_i(x^\star)(x - x^\star) \\ + \langle \nabla f_i(x^\star), x - x^\star\rangle + o(x - x^\star) \end{aligned}$$

where

$$x^\star = \operatorname*{argmin}_x \sum_{i=1}^n f_i(x) \quad \text{and} \quad \lim_{x\to x^\star} \frac{\|o(x-x^\star)\|}{\|x - x^\star\|} = 0.$$

Therefore, one can interpret DOT-ADMM as being characterized by an affine operator with an additive error that depends on the higher order terms $o(x - x^\star)$. But since affine operators are metric subregular [42, 43], then the DOT-ADMM operator is metric subregular around the optimal solution $x^\star$. Finally, since the additive error vanishes around $x^\star$, then we can apply Theorem 4 (and, specifically, the particular cases outlined in Corollary 1) and prove that linear convergence can be achieved locally in mean square.

IV. Tutorial Examples: Linear and Logistic Regression

This section discusses some tutorial examples of distributed learning problems for which DOT-ADMM is characterized by a metric subregular operator, and, as a consequence, Theorems 1 and 2 apply. For simplicity, the discussion is limited to the case of static local costs even though all the results apply straightforwardly to the online scenario with time-varying local costs. We consider *empirical risk minimization* (ERM)

[2]Note that since $x(k)$ is a function of $z(k-1)$, the update of $z(k)$ depends solely on $z(k-1)$ or, in other words, $x(k)$ is an internal variable of DOT-ADMM.

[3]The symbol $\otimes$ denotes the Kronecker product.

Licensed under CC BY 4.0. DOI: 10.1109/TAC.2024.3441723

problems, in which the local cost of each agent $i \in \mathcal{V}$ is defined over the data set $\{a_{i,h}, b_{i,h}\}_{h=1}^{m_i}$, $m_i \in \mathbb{N}$:

$$f_i(x) = \sum_{h=1}^{m_i} g(x, a_{i,h}, b_{i,h}) \tag{12}$$

where $g \in \Gamma_0^p$ is a suitable *loss function*. The goal of an ERM problem is that of computing online a solution $x_k^\star \in \mathbb{R}^p$ to eq. (2) with local costs in eq. (12) where $x_k^\star$ represents the vector of trained parameters of a model. Such a goal can be reached by employing DOT-ADMM, whose updates are ruled by the operator T in (11), recalled next

$$\mathsf{T}(z)=[(1-\alpha)I-\alpha P]z+2\alpha\rho PA\mathrm{prox}_f^{1/\rho\eta}(D^{-1}A^\top z) \tag{13}$$

where $\mathrm{prox}_f^{1/\rho\eta} : \mathbb{R}^{np} \to \mathbb{R}^{np}$ applies block-wise the proximal of the local costs f_i. The specific structure of the operator T allows for *regularized* versions of the local costs $f_i(x)+\frac{\epsilon}{2}\|x\|^2$ which only results in a scaling of the proximal, i.e. (cfr. [20, section 2.2]),

$$\mathrm{prox}^{\rho}_{f_i+\frac{\epsilon}{2}\|\cdot\|^2}(x) = \mathrm{prox}^{1/(\epsilon+1/\rho)}_{f_i}\left(x/(1+\rho\epsilon)\right).$$

To apply Theorems 1 and 2 one needs to prove that T is metric subregular given a specific loss function $g \in \Gamma_0^p$. To this end, we will make use of the following novel result, which provides an *operative way to verify metric subregularity of an operator*. We anticipate here that, differently from linear regression problems, the proximal of robust linear regression and logistic regression costs do not admit a closed-form solution. Nevertheless, resorting to the following Proposition 1 it is possible to show metric subregularity indirectly.

Proposition 1. *An operator $\mathsf{T} : \mathbb{R}^n \to \mathbb{R}^n$ is metric subregular if there is a matrix $A \in \mathbb{R}^{n\times n}$ and two offsets $b_\mathsf{L}, b_\mathsf{U} \in \mathbb{R}^n$ such that T is lower and upper bounded (component-wise) by the affine operators $\mathsf{L}(z) := Az + b_\mathsf{L}$ and $\mathsf{U}(z) := Az + b_\mathsf{U}$, respectively, yielding*

$$\mathsf{L}(z) \le \mathsf{T}(z) \le \mathsf{U}(z), \qquad \forall z \in \mathbb{R}^n. \tag{14}$$

Proof: See Appendix C. ∎

A. Linear regression

In linear regression problems, the data sets are such that $a_{i,h} \in \mathbb{R}^p$ and $b_{i,h} \in \mathbb{R}$ and the loss function g is given by

$$g(x, a_{i,h}, b_{i,h}) = \frac{1}{2}(a_{i,h}^\top x - b_{i,h})^2. \tag{15}$$

The following result holds for this class of problems.

Proposition 2. *Consider the operator T in (13) characterizing DOT-ADMM applied to a linear regression problem 2, that is, with local costs (12) and loss (15). Then, T is metric subregular.*

Proof: Denoting $A_i = [a_{i,1}, \cdots, a_{i,m_i}]^\top \in \mathbb{R}^{m_i\times p}$ and $b_i = [b_{i,1}, \cdots, b_{i,m_i}]^\top \in \mathbb{R}^{m_i}$ the local costs become $f_i(x) = \frac{1}{2}\|A_i x_i - b_i\|^2$. In this particular case, the proximals of the local costs have the following closed-form expression:

$$\mathrm{prox}_{f_i}^{1/\rho\eta_i}(w)=(A_i^\top A_i+\rho\eta_i I)^{-1}\left(\rho\eta_i w+A_i^\top b_i\right). \tag{16}$$

By noticing that the proximals are affine functions of their argument w, it follows that also the operator T in (13) is affine. Consequently, T is metric subregular by Proposition 1. ∎

B. Robust linear regression

Linear regression may be sensitive to outliers when using a quadratic loss. To remedy this, it is possible to formulate a *robust linear regression* problem by using the *Huber loss* in the local costs (12):

$$g(x,a_{i,h},b_{i,h})=\begin{cases}\frac{1}{2}(a_{i,h}^\top x-b_{i,h})^2 & \text{if } |a_{i,h}^\top x-b_{i,h}|\le\theta\\ \theta(|a_{i,h}^\top x-b_{i,h}|-\frac{\theta}{2}) & \text{otherwise}\end{cases} \tag{17}$$

with $\theta > 0$. The following result holds for this class of problems.

Proposition 3. *Let T be the operator characterizing DOT-ADMM applied to a robust linear regression problem, that is, with local costs (12) and loss (17). Then, T is metric subregular.*

Proof: The proximal of the local cost f_i is

$$\mathrm{prox}_{f_i}^{1/\rho\eta_i}(w)=\underset{x}{\mathrm{argmin}}\underbrace{\left\{\sum_{h=1}^{m_i} g(x,a_{i,h},b_{i,h})+\frac{1}{2\rho\eta_i}\|x-w\|^2\right\}}_{h(x)}.$$

Thus, the proximal is the (unique) stationary point of $h(x)$, which is the solution of

$$\frac{\partial}{\partial x}h(x) = \sum_{h=1}^{m_i}\frac{\partial}{\partial x}g(x, a_{i,h}, b_{i,h}) + \frac{1}{\rho\eta_i}(x - w) = 0,$$

Since by the definition of the Huber loss (17) it holds

$$\left|\frac{\partial}{\partial x}g(x, a_{i,h}, b_{i,h})\right| \le \theta|a_{i,h}|, \qquad \forall x \in \mathbb{R}^p,$$

it follows that

$$w - \rho\eta_i\theta|a_{i,h}| \le \mathrm{prox}_{f_i}^{1/\rho\eta_i}(w) \le w + \rho\eta_i\theta|a_{i,h}|.$$

i.e., the proximal is upper/lower bounded by the identity operator with offsets $\pm\rho\eta_i\theta|a_{i,h}|$. Consequently, also the operator T is bounded by two offsetted operators and we can apply Proposition 1 to prove that is metric subregular. ∎

C. Logistic regression

We turn now to classification problems using logistic regression. The data sets in this case are such that $a_{i,h} \in \mathbb{R}^p$ and $b_{i,h} \in \{-1, 1\}$, with the loss

$$g(x, a_{i,h}, b_{i,h}) = \log\left(1 + \exp\left(-b_{i,h}a_{i,h}^\top x\right)\right). \tag{18}$$

The following result holds for this class of problems.

Proposition 4. *Let T be the operator characterizing DOT-ADMM applied to a logistic regression problem, that is, with local costs (12) and loss (18). Then, T is metric subregular.*

Proof:

By definition of the logistic loss (18), it holds

$$\left|\frac{\partial}{\partial x}g(x, a_{i,h}, b_{i,h})\right| = |b_{i,h}a_{i,h}|.$$

The proof follows by similar arguments as those in Proposition 3. ∎

Licensed under CC BY 4.0. DOI: 10.1109/TAC.2024.3441723

V. Numerical results

In this section, we carry out numerical simulations corroborating the theoretical results of the previous sections; all simulations have been implemented in Python using the `tvopt` package [44], and run on a laptop with 12$^{\text{th}}$ generation Intel i7 CPU and 16 GB of RAM.

The considered set-up is a network of $N = 10$ nodes, exchanging information through a random graph topology of 20 edges, that want to solve an online logistic regression problem characterized by (2) and the local costs

$$f_{i,k}(x) = \sum_{h=1}^{m_i} \log\left(1 + \exp\left(-b_{i,h,k} a_{i,h,k} x\right)\right) + \frac{\epsilon}{2}\|x\|^2$$

where $x \in \mathbb{R}^p$, $p = 16$, is the vector of weights and intercept to be learned, and $\{(a_{i,h,k}, b_{i,h,k}\}_{h=1}^{m_i}\}$ with $a_{i,h,k} \in \mathbb{R}^{1\times p}$, $b_{i,h,k} \in \{-1, 1\}$ are the $m_i = 20$ feature vectors and class pairs available to the node at time $k \in \mathbb{N}$. Notice that we add a regularization term ($\epsilon = 5$) to ensure strong convexity.

In the following sections, we discuss the performance of DOT-ADMM in the different scenarios presented in Section I-A. The algorithm will then be compared to the gradient tracking methods [25] (designed to be robust to asynchrony), and [45] (designed to be robust to quantization).

A. Local updates for logistic regression

While running DOT-ADMM, each active node needs to compute the local update (3). However, when applied to logistic regression in eq. (18) the proximal of $f_{i,k}$ does not have a closed form – differently from the linear regression problem (15) – and therefore, the proximal needs to be computed approximately. In our set-up, a node computes an approximation of (3) via the accelerated gradient descent, terminating when the distance between consecutive iterates is smaller than a threshold $\theta > 0$. The error introduced by such an inexact local update is smaller the smaller θ is. However, smaller values of the threshold make the computational time required for a local update longer, presenting a trade-off.

To exemplify this trade-off, we apply DOT-ADMM to a static version of (18). In Table I we report the computational time required to compute the local updates for different choices of θ, as well as the corresponding asymptotic error (that is, the distance $\|x(k) - x^\star\|$ from the unique solution at the end of the simulation). The computational time is computed by averaging over 250 iterations of the algorithm.

Hereafter, unless otherwise stated we use DOT-ADMM with $\theta = 10^{-8}$, for a local update time of $\sim 2.42 \times 10^{-3} s$. For comparison, we note that a local update of DGT (with hand-tuned parameters to improve performance) requires $\sim 1.90 \times 10^{-3} s$, and in the simulations we allow DGT to run two iterations per each iteration of DOT-ADMM, to account for the longer time required in the latter local updates.

B. Quantized communications

As in the section above, we consider a static logistic regression problem, and assume that the agents can exchange quantized communications. In particular, an agent i can only send the quantized version $q(x)$ of a message $x \in \mathbb{R}^p$, as defined component-wise by

$$[q(x)]_j = \begin{cases} \underline{q} & \text{if } [x]_j < \underline{q} \\ \delta\lfloor [x]_j/\delta \rfloor & \text{if } \underline{q} \le [x]_j \le \overline{q} \\ \overline{q} & \text{if } [x]_j > \overline{q} \end{cases}, \quad j \in \{1, \dots, p\}$$

with $\overline{q} = -\underline{q} = 10$, and $\delta > 0$ the quantization level. Table II reports the asymptotic error of DOT-ADMM for different quantization levels δ.

C. Asynchrony

In Section V-A we discussed how the local updates (3) for a logistic regression problem need to be computed recursively as the proximal does not have a closed form solution. We then discussed how the threshold specifying the accuracy of the local update impacts the convergence of the algorithm. Here we consider how recursive local updates can lead to *asynchronous operations* of the agents, due to their heterogeneous computational capabilities.

We consider the following scenario: at iteration k each agent completes the local update (3) – using $\theta = 10^{-8}$ – with some probability, $\underline{p}$ or $\overline{p}$, $\underline{p} < \overline{p}$. The agents characterized by the smaller probability $\underline{p}$ are the "slow" nodes, which, having fewer computational resources, take on average a longer time to reach the threshold θ. Notice that all the nodes use the same threshold, and their more or less frequent updates mimic the effect of different resources.

In Figure 1 we report the mean tracking error (as averaged over 100 Monte Carlo iterations) for the asynchronous case with different numbers of slow nodes N_s. We also compare the result with the error in the synchronous case, in which all nodes complete an update at each iteration k. As discussed in Section III-B.1, asynchronous agent operations,

TABLE I
COMPUTATIONAL TIME OF LOCAL UPDATES AND ASYMPTOTIC ERROR FOR A STATIC LOGISTIC REGRESSION PROBLEM.

Threshold	Comp. time [s]	Asymptotic err.
$\theta = 10^{-14}$	3.47×10^{-3}	4.14×10^{-14}
$\theta = 10^{-12}$	2.84×10^{-3}	3.65×10^{-12}
$\theta = 10^{-10}$	2.42×10^{-3}	4.88×10^{-10}
$\theta = 10^{-8}$	1.95×10^{-3}	5.30×10^{-8}
$\theta = 10^{-6}$	1.39×10^{-3}	1.01×10^{-5}
$\theta = 10^{-4}$	8.88×10^{-4}	5.73×10^{-4}
$\theta = 10^{-2}$	4.12×10^{-4}	9.71×10^{-2}

TABLE II
ASYMPTOTIC ERROR FOR DIFFERENT QUANTIZATION LEVELS.

Quantization	Asymptotic error
No quantization	5.30×10^{-8}
$\delta = 10^{-10}$	5.30×10^{-8}
$\delta = 10^{-8}$	7.36×10^{-8}
$\delta = 10^{-6}$	4.74×10^{-6}
$\delta = 10^{-4}$	5.64×10^{-4}
$\delta = 10^{-2}$	5.32×10^{-2}
$\delta = 10^{-1}$	4.91×10^{-1}

Licensed under CC BY 4.0. DOI: 10.1109/TAC.2024.3441723

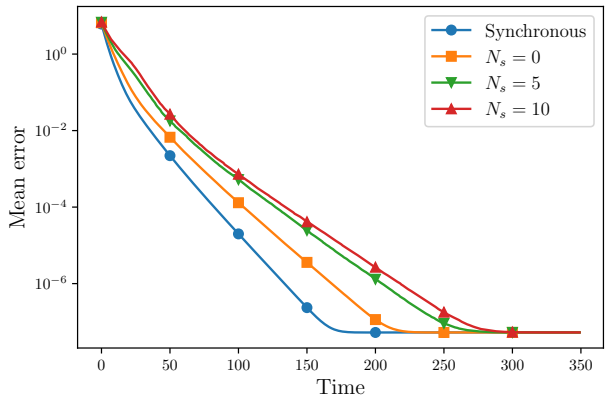


Fig. 1. Error trajectories of DOT-ADMM with synchronous and asynchronous updates for different numbers of slow nodes.

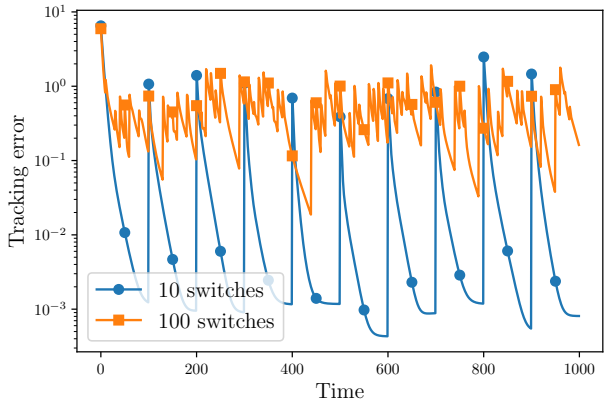


Fig. 3. Tracking error of DOT-ADMM applied to two online problems with different piece-wise constant cost functions.

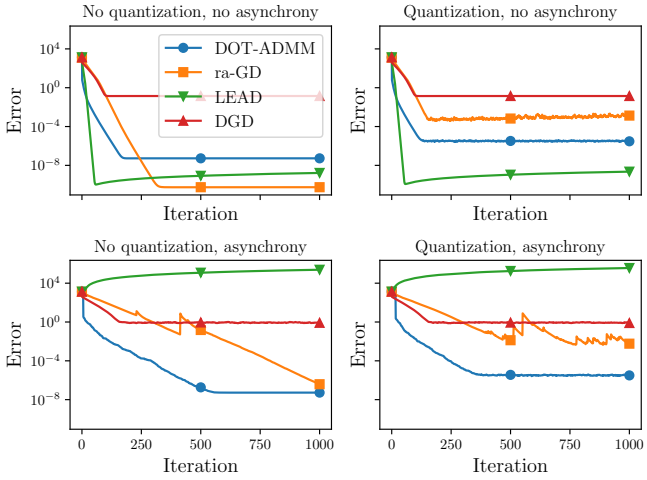


Fig. 2. Comparison on static problems of DOT-ADMM with ra-GD, LEAD, DGD in different scenarios combining quantization/asynchrony.

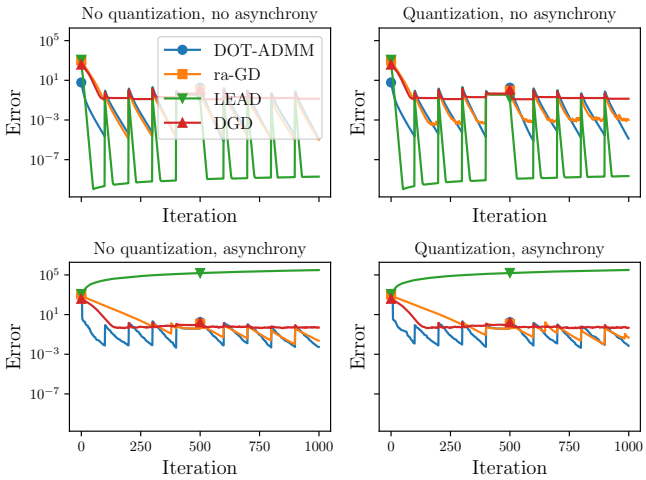


Fig. 4. Comparison on online problems of DOT-ADMM with ra-GD, LEAD, DGD in different scenarios combining quantization/asynchrony.

which translate into random coordinate updates, lead to worse convergence rates. Indeed, the more frequent the updates are, the faster the convergence rate (until achieving that of the synchronous version), and the introduction of slower nodes implies less frequent coordinate updates overall.

D. Online optimization

In this section, we evaluate the performance of DOT-ADMM when applied to two instances of the online logistic regression problem, in which the local cost functions are piece-wise constant. Specifically, the costs change 10 and 100 times, respectively, and are generated so that the maximum distance between consecutive optima is ~ 2.5 (cf. Assumption 2). In Figure 3 we report the tracking error of DOT-ADMM when applied to the two problems. Notice that when the problem changes less frequently, DOT-ADMM has time to converge to smaller errors, up to the bound imposed by the inexact local updates (computed with $\theta = 10^{-4}$). Notice that in the transient the convergence is linear, as predicted by the theory. On the other hand, more frequent changes in the problem yield larger tracking errors overall.

E. Comparison with state-of-the-art algorithms

We conclude by comparing DOT-ADMM with three gradient-based methods, for both static and online problems:

- *ra-GD* [25][4]: gradient tracking algorithm which makes use of the robust ratio consensus to ensure convergence in the presence of asynchrony,
- *LEAD* [45]: gradient tracking algorithm which is designed to be robust to a certain class of unbiased quantizers,
- *DGD* [46]: which does not converge exactly, but has been shown to be robust to additive errors [47] and online scenarios [48], see also [22].

Due to the fact that DOT-ADMM requires a longer time to update the local states (cf. Section V-A), ra-GD, LEAD, and DGD were run for a larger number of iterations to match the computational time of DOT-ADMM. All the step-sizes of these gradient methods were hand-tuned for optimal performance.

In Figure 2 we compare the four algorithms on a static logistic regression problem, and for different scenarios com-

[4]The paper [25] proposes a distributed Newton method, but ra-GD can be derived by replacing the Hessians with identity matrices (cf. [25, Remark IV.1]).

Licensed under CC BY 4.0. DOI: 10.1109/TAC.2024.3441723

bining quantization and asynchrony. In particular, we either use or not the quantizer [45, eq. (14)], and the agents either activate synchronously or asynchronously, using the same set-up as Section V-C. In accordance with the theory, ra-GD is robust to asynchrony, although the convergence is somewhat slow due to a necessarily conservative step-size choice. On the other hand, when quantization is employed the algorithm seems to converge only to a neighborhood of the optimal solution, which is larger than the neighborhood reached by DOT-ADMM (despite the fact that DOT-ADMM also uses inexact updates). As predicted, LEAD shows convergence in the presence of quantization; however, the algorithm is not robust to asynchrony and seems to diverge when the agents are not synchronized. Finally, DGD is robust to both quantization and asynchrony, but its inherent inexactness leads to poor performance.

In Figure 4 we further compare these algorithms for the online logistic regression problem of Section V-D. As we can see the performance of the different algorithms while the costs are not varying largely follows that depicted in Figure 2, with an increase in the error any time the problem changes.

From both Figure 2 and Figure 4 we see that only DOT-ADMM shows robustness to all the three challenges of asynchrony, quantization, and time-varying costs.

VI. Conclusions

This paper proposes DOT-ADMM to solve online learning problems in a multi-agent setting under challenging network constraints, such as asynchronous and inexact agent computations, and unreliable communications. The convergence and robustness of DOT-ADMM have been proven by deriving novel theoretical results in stochastic operator theory for the class of metric-subregular operators, which turns out to be an important class of operators that shows linear convergence to the set of optimal solutions. The broad applicability of this class of operators is supported by the fact that the operator ruling DOT-ADMM applied to the standard linear and logistic regression problems is indeed metric subregular. Future works will focus on studying the optimal design of DOT-ADMM and on the characterization of the linear rate of convergence for specific distributed problems, e.g. online learning and dynamic tracking.

References

[1] D. K. Molzahn, F. Dorfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[2] A. Nedić and J. Liu, "Distributed optimization for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 77–103, 2018.

[3] E. Montijano, G. Oliva, and A. Gasparri, "Distributed estimation and control of node centrality in undirected asymmetric networks," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2304–2311, 2021.

[4] D. Deplano, M. Franceschelli, and A. Giua, "Dynamic min and max consensus and size estimation of anonymous multiagent networks," *IEEE Transactions on Automatic Control*, vol. 68, no. 1, pp. 202–213, 2023.

[5] D. Deplano, M. Franceschelli, and A. Giua, "A nonlinear perron–frobenius approach for stability and consensus of discrete-time multi-agent systems," *Automatica*, vol. 118, p. 109 025, 2020.

[6] M. Santilli, M. Franceschelli, and A. Gasparri, "Dynamic resilient containment control in multirobot systems," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 57–70, 2022.

[7] D. Deplano, M. Franceschelli, and A. Giua, "Novel stability conditions for nonlinear monotone systems and consensus in multi-agent networks," *IEEE Transactions on Automatic Control*, pp. 1–14, 2023.

[8] Y. Shang, "Resilient consensus in multi–agent systems with state constraints," *Automatica*, vol. 122, p. 109 288, 2020.

[9] L. Sheng, W. Gu, G. Cao, and X. Chen, "A distributed detection mechanism and attack–resilient strategy for secure voltage control of ac microgrids," *CSEE Journal of Power and Energy Systems*, pp. 1–10, 2022.

[10] M. Santilli, M. Franceschelli, and A. Gasparri, "Secure rendezvous and static containment in multi-agent systems with adversarial intruders," *Automatica*, vol. 143, p. 110 456, 2022.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "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.

[12] L. Qian, P. Yang, M. Xiao, O. A. Dobre, M. D. Renzo, J. Li, Z. Han, Q. Yi, and J. Zhao, "Distributed learning for wireless communications: Methods, applications and challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 3, pp. 326–342, 2022.

[13] J. Park, S. Samarakoon, A. Elgabli, J. Kim, M. Bennis, S.-L. Kim, and M. Debbah, "Communication-efficient and distributed learning over wireless networks: Principles and applications," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 796–819, 2021.

[14] G. Notarstefano, I. Notarnicola, and A. Camisa, "Distributed optimization for smart cyber-physical networks," *Foundations and Trends® in Systems and Control*, vol. 7, no. 3, pp. 253–383, 2019.

[15] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

[16] X. Li, L. Xie, and N. Li, "A survey on distributed online optimization and online games," *Annual Reviews in Control*, vol. 56, p. 100 904, 2023.

[17] T. Gafni, N. Shlezinger, K. Cohen, Y. C. Eldar, and H. V. Poor, "Federated Learning: A signal processing perspective," *IEEE Signal Processing Magazine*, vol. 39, no. 3, pp. 14–41, 2022.

[18] Z. Peng, T. Wu, Y. Xu, M. Yan, and W. Yin, "Coordinate friendly structures, algorithms and applications," *Annals of Mathematical Sciences and Applications*, vol. 1, no. 1, pp. 57–119, 2016.

[19] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized sgd with changing topology and local updates," in *Proceedings of the 37th International Conference on Machine Learning*, H. D. III and A. Singh, Eds., ser. Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 5381–5393.

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

[21] R. Xin, S. Pu, A. Nedić, and U. A. Khan, "A general framework for decentralized optimization with first-order methods," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 1869–1889, 2020.

[22] K. Yuan, W. Xu, and Q. Ling, "Can primal methods outperform primal-dual methods in decentralized dynamic optimization?" *IEEE Transactions on Signal Processing*, vol. 68, pp. 4466–4480, 2020.

Licensed under CC BY 4.0. DOI: 10.1109/TAC.2024.3441723

[23] G. Carnevale, F. Farina, I. Notarnicola, and G. Notarstefano, "GTAdam: Gradient tracking with adaptive momentum for distributed online optimization," *IEEE Transactions on Control of Network Systems*, pp. 1–12, 2023.
[24] 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.
[25] N. Bof, R. Carli, G. Notarstefano, L. Schenato, and D. Varagnolo, "Multiagent newton-raphson optimization over lossy networks," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2983–2990, 2019.
[26] Y. Tian, Y. Sun, and G. Scutari, "Achieving linear convergence in distributed asynchronous multiagent optimization," *IEEE Transactions on Automatic Control*, vol. 65, no. 12, pp. 5264–5279, 2020.
[27] H. Li, Z. Lin, and Y. Fang, "Variance reduced EXTRA and DIGing and their optimal acceleration for strongly convex decentralized optimization," *Journal of Machine Learning Research*, vol. 23, no. 222, pp. 1–41, 2022.
[28] J. Lei, P. Yi, J. Chen, and Y. Hong, "Distributed variable sample-size stochastic optimization with fixed step-sizes," *IEEE Transactions on Automatic Control*, vol. 67, no. 10, pp. 5630–5637, 2022.
[29] M. Bin, I. Notarnicola, and T. Parisini, "Stability, linear convergence, and robustness of the Wang-Elia algorithm for distributed consensus optimization," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, Cancun, Mexico: IEEE, 2022, pp. 1610–1615.
[30] 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, A2851–A2879, 2016.
[31] E. Wei and A. Ozdaglar, "On the O(1/k) convergence of asynchronous distributed alternating Direction Method of Multipliers," in *2013 IEEE Global Conference on Signal and Information Processing*, 2013, pp. 551–554.
[32] 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 Transactions on Signal Processing*, vol. 64, no. 12, pp. 3118–3130, 2016.
[33] L. Majzoobi, V. Shah-Mansouri, and F. Lahouti, "Analysis of distributed ADMM algorithm for consensus optimisation over lossy networks," *IET Signal Processing*, vol. 12, no. 6, pp. 786–794, 2018.
[34] N. Bastianello, R. Carli, L. Schenato, and M. Todescato, "Asynchronous distributed optimization over lossy networks via relaxed ADMM: Stability and linear convergence," *IEEE Transactions on Automatic Control*, vol. 66, no. 6, pp. 2620–2635, 2021.
[35] L. Majzoobi, F. Lahouti, and V. Shah-Mansouri, "Analysis of distributed ADMM algorithm for consensus optimization in presence of node error," *IEEE Transactions on Signal Processing*, vol. 67, no. 7, pp. 1774–1784, 2019.
[36] Y. Xie and U. V. Shanbhag, "SI-ADMM: A stochastic inexact ADMM framework for stochastic convex programs," *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2355–2370, 2020.
[37] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, 2nd ed. Cham, Switzerland: Springer, 2017.
[38] E. Dall'Anese, A. Simonetto, S. Becker, and L. Madden, "Optimization and Learning With Information Streams: Time-varying algorithms and applications," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 71–83, 2020.
[39] A. Simonetto, E. Dall'Anese, S. Paternain, G. Leus, and G. B. Giannakis, "Time-varying convex optimization: Time-structured algorithms and applications," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 2032–2048, 2020.
[40] P. L. Combettes and J.-C. Pesquet, "Stochastic quasi-fejér block-coordinate fixed point iterations with random sweeping," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 1221–1248, 2015.
[41] N. Bastianello, L. Madden, R. Carli, and E. Dall'Anese, "A stochastic operator framework for optimization and learning with sub-weibull errors," *arXiv preprint arXiv:2105.09884*, 2023.
[42] S. M. Robinson, "Some continuity properties of polyhedral multifunctions," in *Mathematical Programming at Oberwolfach*, Springer, 1981, pp. 206–214.
[43] A. Themelis and P. Patrinos, "Supermann: A superlinearly convergent algorithm for finding fixed points of nonexpansive operators," English, *IEEE Transactions on Automatic Control*, vol. 64, no. 12, pp. 4875–4890, 2019.
[44] N. Bastianello, "Tvopt: A python framework for time-varying optimization," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 227–232.
[45] X. Liu, Y. Li, R. Wang, J. Tang, and M. Yan, "Linear convergent decentralized optimization with compression," *arXiv preprint arXiv:2007.00232*, 2020.
[46] K. Yuan, Q. Ling, and W. Yin, "On the Convergence of Decentralized Gradient Descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.
[47] N. Bastianello and E. Dall'Anese, "Distributed and Inexact Proximal Gradient Method for Online Convex Optimization," in *2021 European Control Conference (ECC)*, Delft, Netherlands, 2021, pp. 2432–2437.
[48] A. Simonetto and G. Leus, "Distributed asynchronous time-varying constrained optimization," in *2014 48th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA: IEEE, 2014, pp. 2142–2146.
[49] F. Bullo, *Contraction Theory for Dynamical Systems*, 1.1. Kindle Direct Publishing, 2023.
[50] S. Sundhar Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of Optimization Theory and Applications*, vol. 147, no. 3, pp. 516–545, 2010.

APPENDIX B: PROOF OF THEOREM 3

[Punctual upper bound] - We make use of the so-called *diagonally-weighted norm* in the sense of [49], where the vector of positive weights is the vector of probabilities $p = [p_1, \dots, p_m]^\top$, which is defined next

$$\|\!|z|\!\|^2 = \sum_{\ell=1}^{m} \frac{1}{p_\ell} z_\ell^2, \tag{19}$$

following the usual notation in our community [40]. Clearly, such a norm satisfies the following, where recall that $\overline{p} = \max_\ell p_\ell$, $\underline{p} = \min_\ell p_\ell$

$$\underline{p}\|\!|z|\!\|^2 \leq \|z\|^2 \leq \overline{p}\|\!|z|\!\|^2. \tag{20}$$

Similarly to the Euclidean distance $d_{\mathsf{T}_k}(z)$ from the set of fixed points of T, we define the distance $d'_{\mathsf{T}_k}(z) = \inf_{y \in \mathrm{fix}(\mathsf{T}_k)} \|\!|z - y|\!\|$, such that

$$\frac{1}{\overline{p}} d^2_{\mathsf{T}_k}(z) \overset{(i)}{\leq} d'^2_{\mathsf{T}_k}(z) \overset{(i)}{\leq} \frac{1}{\underline{p}} d^2_{\mathsf{T}_k}(z) \overset{(ii)}{\leq} \frac{\gamma^2}{\underline{p}} \|(\mathsf{Id} - \mathsf{T}_k)z\|^2, \tag{21}$$

where (i) follow by the definition of projection and eq. (20), whereas (ii) by γ-metric subregularity of T_k.

We also conveniently rewrite the operator $\widetilde{\mathsf{T}}^e_k$ in eq. (10) by

$$z_\ell(k) = \widetilde{\mathsf{T}}^e_{\ell,k}(z(k-1)) = \widetilde{\mathsf{T}}_{\ell,k}(z(k-1)) + \beta_{\ell,k} e_{\ell,k},$$

Licensed under CC BY 4.0. DOI: 10.1109/TAC.2024.3441723

where

$$\widetilde{\mathsf{T}}_{\ell,k}(z(k-1)) = z_\ell(k-1)+\beta_{\ell,k}(\mathsf{T}_{\ell,k}(z(k-1))-z_\ell(k-1)).$$

Letting $e_k \in \mathbb{R}^m$ be the vector stacking all the errors and $z_k^\star \in \mathrm{fix}(\mathsf{T}_k)$, then by eq. (21) and the triangle inequality we can write

$$d'_{\mathsf{T}_k}(z(k))=\left|\!\left|\!\left|\widetilde{\mathsf{T}}_k^e(z(k-1))-z_k^\star\right|\!\right|\!\right| \le \left|\!\left|\!\left|\widetilde{\mathsf{T}}_k(z(k-1))-z_k^\star\right|\!\right|\!\right| + |\!|\!|e_k|\!|\!|$$

and thus

$$\mathbb{E}\left[d'_{\mathsf{T}}(z(k))\right] \le \mathbb{E}\left[\left|\!\left|\!\left|\widetilde{\mathsf{T}}_k(z(k-1))-z_k^\star\right|\!\right|\!\right|\right] + \mathbb{E}\left[|\!|\!|e_k|\!|\!|\right].$$

We are interested in finding an upper-bound to the first term on the right-hand side of the above inequality, whose explicit form is given by $\left|\!\left|\!\left|\widetilde{\mathsf{T}}_k(z(k-1))-z_k^\star\right|\!\right|\!\right|^2 =$

$$\begin{aligned}
&=\sum_{\ell=1}^m \frac{1}{p_\ell}\left[(1-\beta_{\ell,k})z_\ell(k-1)+\beta_{\ell,k}\mathsf{T}_{\ell,k}(z(k-1))-z_{\ell,k}^\star)\right]^2\\
&=\sum_{\ell=1}^m \left[\frac{1-\beta_{\ell,k}}{p_\ell}(z_\ell(k-1)-z_\ell^\star)^2+\frac{\beta_\ell}{p_{ij}}(\mathsf{T}_{\ell,k}(z(k-1))-z_{\ell,k}^\star)^2\right]
\end{aligned}$$

where, since $\beta_{\ell,k} \in \{0,1\}$, we have used the following: $\beta_{\ell,k}^2 = \beta_{\ell,k}$; $(1-\beta_{\ell,k})^2 = (1-\beta_{\ell,k})$; $(1-\beta_{\ell,k})\beta_{\ell,k} = 0$. An upper-bound to the conditional expectation w.r.t. the realizations of all r.v.s at time $k-1$ is given by

$$\begin{aligned}
&\mathbb{E}_{k-1}\left[\left|\!\left|\!\left|\widetilde{\mathsf{T}}_k(z(k-1))-z_k^\star\right|\!\right|\!\right|^2\right] =\\
&=\sum_{\ell=1}^m \left[\frac{1-p_\ell}{p_\ell}(z_\ell(k-1)-z_{\ell,k}^\star)^2+(\mathsf{T}_{\ell,k}(z(k-1))-z_{\ell,k}^\star)^2\right]\\
&=|\!|\!|z(k-1)-z_k^\star|\!|\!|^2-\|z(k-1)-z_k^\star\|^2+\|\mathsf{T}_k(z(k-1))-z_k^\star\|^2\\
&\overset{(i)}{\le}|\!|\!|z(k-1)-z_k^\star|\!|\!|^2-\frac{1-\alpha}{\alpha}\|(\mathsf{Id}-\mathsf{T}_k)z(k-1)\|^2\\
&\overset{(ii)}{\le} d'^2_{\mathsf{T}_k}(z(k-1))-\frac{1-\alpha}{\alpha}\|(\mathsf{Id}-\mathsf{T}_k)z(k-1)\|^2\\
&\overset{(iii)}{\le} d'^2_{\mathsf{T}_k}(z(k-1))-\frac{1-\alpha}{\alpha\gamma^2}\underline{p}d'^2_{\mathsf{T}_k}(z(k-1))\\
&=\left(1-\frac{(1-\alpha)\underline{p}}{\alpha\gamma^2}\right)d'^2_{\mathsf{T}_k}(z(k-1)):=\mu^2 d'^2_{\mathsf{T}_k}(z(k-1))
\end{aligned}$$

where (i) holds by α-averagedness, (ii) follows by selecting $z_k^\star = \mathrm{arginf}_{y\in\mathrm{fix}(\mathsf{T}_k)}|\!|\!|z(k-1)-y|\!|\!|$, and (iii) is a consequence of metric subregularity highlighted in eq. (21), and where $\mu \in (0,1)$ provided that γ is sufficiently large, which we can always assume by overestimating the metric subregularity constant of T_k, in particular by replacing γ^2 with $\lambda = \max\{\gamma^2, (1-\alpha)\underline{p}/\alpha\}$ as in eq. (9). Now, exploiting the (i) concavity of the square root, the (ii) Jensen's inequality and the (iii) law of total expectation, we have

$$\mathbb{E}\left[|\!|\!|\cdot|\!|\!|\right] \overset{(i)}{=} \mathbb{E}\left[\sqrt{|\!|\!|\cdot|\!|\!|^2}\right] \overset{(ii)}{\le} \sqrt{\mathbb{E}\left[|\!|\!|\cdot|\!|\!|^2\right]} \overset{(iii)}{=} \sqrt{\mathbb{E}\left[\mathbb{E}_{k-1}\left[|\!|\!|\cdot|\!|\!|^2\right]\right]}$$

which implies

$$\mathbb{E}\left[\left|\!\left|\!\left|\widetilde{\mathsf{T}}_k(z(k-1))-z^\star\right|\!\right|\!\right|\right] \le \mu\mathbb{E}\left[d_{\mathsf{T}}(z(k-1))\right].$$

Let us now combine the definition of d'_{T_k}, assumption *i)*, and the triangle inequality to derive the following bound

$$\begin{aligned}
d'_{\mathsf{T}_k}(z(k-1)) &= \left|\!\left|\!\left|z(k-1)-\mathrm{proj}_{\mathrm{fix}(\mathsf{T}_k)}(z(k-1))\right|\!\right|\!\right|\\
&\le \left|\!\left|\!\left|z(k-1)-\mathrm{proj}_{\mathrm{fix}(\mathsf{T}_{k-1})}(z(k-1))\right|\!\right|\!\right|\\
&+ \left|\!\left|\!\left|\mathrm{proj}_{\mathrm{fix}(\mathsf{T}_k)}(z(k-1))-\mathrm{proj}_{\mathrm{fix}(\mathsf{T}_{k-1})}(z(k-1))\right|\!\right|\!\right|\\
&\le d'_{\mathsf{T}_{k-1}}(z(k-1))+\frac{1}{\sqrt{\underline{p}}}\varsigma.
\end{aligned}$$

Therefore we now write

$$\begin{aligned}
\mathbb{E}\left[d'_{\mathsf{T}_k}(z(k))\right] &\le \mu\mathbb{E}\left[d'_{\mathsf{T}_k}(z(k-1))\right]+\mathbb{E}[|\!|\!|e_k|\!|\!|]\\
&\le \mu\mathbb{E}\left[d'_{\mathsf{T}_{k-1}}(z(k-1))\right]+\frac{1}{\sqrt{\underline{p}}}(\mu\varsigma+\mathbb{E}[\|e_k\|]).
\end{aligned}$$

Iterating we get

$$\mathbb{E}\left[d'_{\mathsf{T}_k}(z(k))\right] \le \mu^k\mathbb{E}\left[d'_{\mathsf{T}_0}(z(0))\right]+\frac{1}{\sqrt{\underline{p}}}\sum_{h=1}^k \mu^{k-h}(\mu\varsigma+\mathbb{E}[\|e_h\|])$$

and using (20) again yields the thesis. Also, when the problem is static ($\varsigma = 0$) there is not any source of error ($e_k = 0$ for all k), the linear convergence holds also in mean square, indeed,

$$\mathbb{E}\left[d'^2_{\mathsf{T}_k}(z(k))\right] \le \mu^{2k}\mathbb{E}\left[d'^2_{\mathsf{T}_0}(z(0))\right]. \tag{22}$$

[Asymptotic upper bound] - We use the same proof technique as in [41, Corollary 5.3]. Let us define

$$y(k) = \max\left\{0, d_{\mathsf{T}_k}(z(k)) - \sqrt{\frac{\overline{p}}{\underline{p}}}\sum_{h=1}^k \mu^{k-h}(\|e_h\|+\mu\varsigma)\right\}$$

for which, by the previous result on the expected distance and Markov's inequality, we have that for any $\varepsilon > 0$

$$\mathbb{P}[y(k) \ge \varepsilon] \le \frac{\mathbb{E}[y(k)]}{\varepsilon} \le \frac{1}{\varepsilon}\sqrt{\frac{\overline{p}}{\underline{p}}}\mu^k d(z(0)),$$

and, summing over k and using the geometric series,

$$\sum_{k=0}^\infty \mathbb{P}[y(k) \ge \varepsilon] \le \frac{1}{\varepsilon}\sqrt{\frac{\overline{p}}{\underline{p}}}\frac{d(z(0))}{1-\mu} < \infty.$$

But by Borel-Cantelli lemma this means that $\limsup_{k\to\infty} y(k) \le \varepsilon$ almost surely; and since the inequality holds for any $\varepsilon > 0$, the thesis follows.

Appendix B: Proof of Theorem 4

The first claim is a straightforward consequence of Theorem 3 and Lemma 3.1(a) in [50]. For the second claim, notice that the map T is metric subregular at fixed points; indeed, if $z^\star \in \mathrm{fix}(\mathsf{T})$ then $d_{\mathsf{T}}(z^\star) = 0$ and $\|(\mathsf{Id}-\mathsf{T})z^\star\| = 0$. This means that $\mathrm{fix}(\mathsf{T}) \subset \mathcal{X}$ and, in turn, that $\mathcal{X}$ is a neighborhood of $\mathrm{fix}(\mathsf{T})$ with

$$\exists r > 0: \quad \mathcal{X} \supset \{z \in \mathbb{R}^m \mid d_{\mathsf{T}}(z) \le r\}.$$

But by the first claim, we know that $z(k)$ converges almost surely to $\mathrm{fix}(\mathsf{T})$ and, therefore, there exists a finite time $k^\star \in \mathbb{N}$ after which the $z(k)$ evolves inside the neighborhood $\mathcal{X}$ in which locally metric subregularity holds. We can now apply Theorem 3 to prove linear convergence in mean for $k \ge k^\star$, completing the proof.

Licensed under CC BY 4.0. DOI: 10.1109/TAC.2024.3441723

APPENDIX C: PROOF OF PROPOSITION 1

Given an operator $\mathsf{F} : \mathbb{R}^n \to \mathbb{R}^n$, for any $x \in \mathbb{R}^n$ we denote $\hat{x}^{\mathsf{F}} \in \mathrm{fix}(\mathsf{T})$ one of the closest fixed points of F to x, namely

$$\hat{x}^{\mathsf{F}} \in \operatorname*{arginf}_{y \in \mathrm{fix}(\mathsf{T})} \|x - y\| \quad \Rightarrow \quad d_{\mathsf{T}}(x) = \left\|x - \hat{x}^{\mathsf{F}}\right\|.$$

For each component $i \in= 1, \dots, n$, L_i and U_i are affine functions with the same slope but different intercept, yielding

$$\hat{x}_i^{\mathsf{L}} \leq \hat{x}_i^{\mathsf{T}} \leq \hat{x}_i^{\mathsf{U}} \quad \Rightarrow \quad \hat{x}_i^{\mathsf{L}} - x_i \leq \hat{x}_i^{\mathsf{T}} - x_i \leq \hat{x}_i^{\mathsf{U}} - x_i,$$

and therefore $\left|x_i - \hat{x}_i^{\mathsf{T}}\right| \leq \max\{\left|x_i - \hat{x}_i^{\mathsf{U}}\right|, \left|x_i - \hat{x}_i^{\mathsf{L}}\right|\}$. Thus, the following chain of inequalities holds

$$\begin{aligned} d_{\mathsf{T}}(x)^2 = \left\|x - \hat{x}^{\mathsf{T}}\right\| &\leq \textstyle\sum_{i=1}^n \max\{\left|x_i - \hat{x}_i^{\mathsf{U}}\right|^2, \left|x_i - \hat{x}_i^{\mathsf{L}}\right|^2\} \\ &\overset{(b)}{\leq} \textstyle\sum_{i=1}^n \max\{\left\|x - \hat{x}^{\mathsf{U}}\right\|_\infty^2, \left\|x - \hat{x}^{\mathsf{L}}\right\|_\infty^2\} \\ &\overset{(c)}{\leq} n \max\{\left\|x - \hat{x}^{\mathsf{U}}\right\|^2, \left\|x - \hat{x}^{\mathsf{L}}\right\|^2\} \end{aligned}$$

where (b) holds since the infinity norm is the maximum distance among each component, (c) holds since $\|x\|_\infty \leq \|x\|$. We now exploit metric subregularity of U and L due to Proposition 1 to prove metric subregularity of T as follows

$$\begin{aligned} d_{\mathsf{T}}(x) &\leq \sqrt{n} \max\{\left\|x - \hat{x}^{\mathsf{U}}\right\|, \left\|x - \hat{x}^{\mathsf{L}}\right\|\} \\ &\leq \sqrt{n} \max\{\gamma_{\mathsf{L}} \|(\mathsf{Id} - \mathsf{L})x\|, \gamma_{\mathsf{U}} \|(\mathsf{Id} - \mathsf{U})x\|\} \\ &\leq \sqrt{n} \max\{\gamma_{\mathsf{L}}, \gamma_{\mathsf{U}}\} \max\{\|(\mathsf{Id} - \mathsf{L})x\|, \|(\mathsf{Id} - \mathsf{U})x\|\} \\ &\leq \gamma_{\mathsf{T}} \|(\mathsf{Id} - \mathsf{T})x\|, \end{aligned}$$

where the last inequality always holds for a sufficiently large value of γ_{T}. This completes the proof.

Licensed under CC BY 4.0. DOI: 10.1109/TAC.2024.3441723