Distributed Decision-Coupled Constrained Optimization via Proximal-Tracking

Alessandro Falsone [a], Maria Prandini [a]

[a] *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Via Ponzio 34/5, 20133 Milano, Italy*

Abstract

In this paper we deal with decision-coupled problems involving multiple agents over a network. Each agent has its own local objective function and local constraints, and all agents aim at finding the value of a common decision vector that minimizes the sum of all agents cost functions and satisfies all local constraints. To this purpose, we introduce a Proximal-Tracking distributed optimization algorithm that integrates dynamic average consensus within the proximal minimization method. Convergence to an optimal consensus solution is guaranteed for any value of a constant penalty parameter, under a convexity assumption only, without requiring differentiability, Lipschitz continuity, or smoothness of the local objective functions. Numerical simulations show the effectiveness of the proposed scheme.

Key words: Distributed Optimization; Decision-Coupled Optimization; Proximal Algorithm; Gradient-Tracking.

1 Introduction

In the last decade we have been experiencing an increasing pervasiveness of network-connected smart devices in our everyday life as well as in the energy, transportation and manufacturing sectors, where cloud-computing services for remote operation and monitoring have also been introduced. This technological innovation process has encouraged a shift of the focus of the control community from the operation of a single device to the coordination of multiple interacting devices with communication and computational capabilities. New challenges then arise in such a multi-agent system scenario, like privacy concerns if agents are asked to disclose sensitive information. This calls for the introduction of distributed algorithms where each agent contributes to the overall optimal coordination problem solution without sharing its private information with the others.

In this paper we focus on those coordination problems that can be formulated as a (convex) mathematical program in which each agent has its own cost function and its own set of local constraints, and the agents collectively aim at finding the value of a *common decision* vector which minimizes the sum of their local cost functions and satisfies all their local constraints (*decision-coupled problems*). Each agent has access only to its local cost function and constraints, but the agents can cooperate by exchanging their tentative solutions with their neighbors, according to a given communication topology. These problems arise naturally in distributed machine learning [20] and federated learning [1].

The first distributed algorithms dealing with this set-up dates back to [12, 23], where a combination of (sub)gradient iterations and consensus schemes are proposed either in absence of constraints, [23], or with agents having local constraints being all equal to a common constraint set, [12]. Heterogeneous constraints are firstly considered in [24] albeit convergence of the proposed scheme is shown only for an all-to-all (instead of neighbor-to-neighbor) communication strategy. Such a stricter communication assumption is relaxed in [14] at the expense of requiring the agents local objective functions to be smooth (i.e., differentiable with Lipschitz continuous gradient). The differentiability assumption on local objective functions is finally removed in [38]. Different local constraint sets without assuming differentiability are also handled by the approach in [18], where a distributed scheme based on consensus and proximal minimization is proposed, which removes the need to compute (sub)gradients of the objective functions. In [6] an approach dealing with non-convex objective functions is also proposed, but it requires smoothness

⋆ Corresponding author A. Falsone. Tel. +39-02-23993542. Fax +39-02-23993412.

Email addresses:
alessandro.falsone@polimi.it (Alessandro Falsone), maria.prandini@polimi.it (Maria Prandini).

and does not handle different local constraints sets. All the mentioned approaches are applicable to dynamic communication topologies, but require the consensus updates to be balanced (doubly stochasticity assumption on the consensus weights), they either assume the local objective functions to be Lipschitz or the local constraint sets to be bounded, and employ diminishing step-sizes (for (sub)gradient-like approaches) or penalty parameters (for proximal-based methods) to ensure convergence to the optimal solution, which ultimately leads to low convergence rates, see, e.g., [14, 23] for two examples of explicit convergence rates of distributed (sub)gradient methods.

Some research effort has been successfully devoted to relaxing the doubly stochasticity assumption on the consensus weights. More specifically, in [21] the authors propose to combine a distributed (sub)gradient scheme with the so-called push-sum protocol proposed in [7], which requires the matrix to be only stochastic as opposed to doubly stochastic. The extension is particularly useful when dealing with directed communication topologies for which a doubly stochastic matrix is more difficult to construct in a distributed way, [9]. The push-sum scheme has been widely adopted ever since to relax the doubly stochasticity assumption of existing distributed approaches, see, e.g., [5,22]. Note that, in this paper, we focus on undirected communication topologies, for which simple algorithms exist to construct a doubly-stochastic matrix in a distributed way, see e.g. [23, Assumption 6].

As for the time-varying step-size, in [28] the distributed (sub)gradient algorithm in [23] is modified by adding a correction term which guarantees convergence for a constant step-size, at the expense of assuming smoothness of the agents local cost functions, a static communication topology, and doubly stochasticity of the consensus weights. With an additional strong convexity assumption, the authors show linear convergence rate of their method. More recently, (sub)gradient-based distributed approaches have been combined with a technique known as dynamic average consensus (firstly proposed in [37] and then more deeply discussed in [13]) to provide better convergence rates via constant step-size. The work in [31] proposes a Newton-Raphson distributed method which converges with a fixed step-size under strong convexity and smoothness assumptions on the local cost functions. The work in [22] significantly extends the work of [28] guaranteeing convergence with a (sufficiently small) fixed step-size on dynamic communication topologies and without doubly stochasticity (using the push-sum protocol), but requiring strong convexity and smoothness. In [25], the authors show that the method proposed in [22] converges also without assuming strong convexity, under fixed communication topologies and doubly stochastic weights. The authors of [36] further study the method in [22] proving convergence even when the agents have different step-sizes. Cost functions are assumed to be smooth and radially unbounded and a linear convergence rate is guaranteed under an additional strong convexity assumption. The work in [33] further improves the convergence rate of the same scheme under the same assumptions. Finally, the works in [2,35] allow the local cost function of the agents to be the sum of a smooth term and a non-smooth term, which are however required to be, respectively, strongly convex and identical for all the agents. In all mentioned approaches employing a constant step-size, different local constraints are not considered and the step-size parameter has to be carefully chosen to be sufficiently small to ensure convergence. For a very recent overview on distributed optimization approaches we refer the reader to [20].

Other distributed schemes ensuring fast convergence are based on the Alternating Direction Method of Multipliers (ADMM, [4]) and firstly appeared in [16, 30]. Under twice differentiability, strong convexity, and smoothness assumptions for the agents' local objective functions, the authors of [16] developed an ADMM-based algorithm with a linear convergence rate. An accelerated variant of the same method is proposed in [30]. The work in [10] shows that linear convergence can be preserved assuming strong convexity of the sum of the agents local cost functions only, rather than of each one of them. Finally, the work in [17] establishes convergence with sub-linear rate of the same scheme, but without requiring strong convexity and smoothness. It is also worth mentioning that [11] proposes an Augmented Lagrangian scheme, which is similar to ADMM but involves a nested loop strategy, with a linear convergence rate under strong convexity and smoothness assumptions. The cited ADMM-based approaches do not consider the presence of local constraints and require the communication topology to be static because they encode it inside optimization problem. The ADMM-based algorithm in [19] instead considers local constraints, but the agents perform the updates in a sequential (rather than parallel) fashion. All ADMM-based methods use a constant penalty parameter, which is the counterpart of the step-size in gradient-based approaches, but, unlike gradient-based approaches, its value can be set arbitrarily.

In this paper we propose a novel distributed optimization algorithm, called Proximal-Tracking, based on dynamic average consensus and proximal minimization to solve convex *decision-coupled problems*. In contrast with the previously mentioned approaches, Proximal-Tracking has the following appealing features:

1. convergence is guaranteed for *any* value of a single penalty parameter;
2. *different* and possibly *unbounded* local constraints set are allowed;
3. local cost functions are not required to be differentiable, Lipschitz continuous, or smooth, but *only convex*.

None of the approaches reviewed above exhibit all these features jointly. To the best of our knowledge, the only exceptions are [15,29]. Methods in [15,29] are extensions of [28] and, differently from our approach, employ a mix of gradient updates and proximal minimization. Both are able to handle non-smooth objectives and different local constraints, and use a constant step-size for the gradient updates and a constant penalty parameter for the proximal minimization step. Convergence is guaranteed for a sufficiently small step-size, assuming a fixed communication topology and doubly stochasticity of the consensus weights. The most recent version [15] allows each agent to use different step-sizes, but the consensus weights have to be carefully adapted based on the maximum step-size.

As it will be clarified in the sequel, the proposed algorithm has a close connection with the distributed gradient method in [22]. As such, a promising direction of investigation is the extension of the proposed method to dynamic communication topologies, by mimicking [22, Section 3]. This extension appears more difficult for the approaches in [15,29] since they rely on a reformulation of the original decision-coupled problem that changes with the communication topology. This is because all agents have a copy of the common decision vector and these copies are forced to be equal through constraints that encode both the communication topology and the consensus weights, see [32, Section II-A and II-B], similarly to the ADMM-based approaches discussed above. Furthermore, in the considered numerical example, the proposed scheme appears to be slightly faster than [15, 29] in terms of convergence rate.

Lastly, we would like to mention that, thanks to the parallelism with [22], the proposed method appears prone to a relaxation of the doubly-stochasticity assumption to a stochasticity only requirement by leveraging the push-sum protocol in [7] or mimicking recently developed gradient-tracking schemes like [34]. Also this relaxation requires further investigation and is not presented here.

The remainder of the paper unfolds as follows. In Section 2, after introducing the notation, we formalize the problem and provide some background useful for the following derivations. In Section 3 we constructively derive the proposed algorithm and we analyze its convergence. In Section 4 we showcase the performance on a numerical example and, finally, in Section 5 we draw some concluding remarks.

Notation We denote with $\mathbb{N}$ the set of natural numbers, and with $\mathbb{R}$ the set of real numbers. For a function f, we denote by $\partial f(x)$ the subdifferential (i.e., the set of all subgradients) of f at x. If f is differentiable at x, then $\partial f(x) = \{\nabla f(x)\}$ and $\nabla f(x)$ denotes the gradient of f at x. $\mathcal{I}_X(x)$ is the indicator function of the set X, which is equal to zero if $x \in X$ and $+\infty$ if $x \notin X$. The Minkowski sum between sets is denoted by $\oplus$, the Cartesian product is denoted as $\times$, and $\operatorname{relint}(\cdot)$ denotes the relative interior of its argument. The vector in $\mathbb{R}^n$ containing all ones is denoted by $\mathbb{1}_n$. The identity matrix and the zero matrix of order n are denoted by I_n and 0_n (for brevity, subscript will be omitted when clear from the context). The Kronecker product is denoted by $\otimes$. For a matrix S we write $S^\top$ to denote its transpose, $\rho(S)$ to denote its spectral radius, $S \succ 0$ when S is positive definite, and $S \succeq 0$ when S is positive semi-definite. For a vector v, $\|v\|$ is the Euclidean norm of v, and, for any matrix $S \succeq 0$, $\|v\|_S$ is the weighted (semi)norm of v, i.e., $\|v\|_S^2 = v^\top S v$.

2 Preliminaries

In this section we describe the problem set-up, recall the proximal minimization algorithm along its gradient interpretation, then we introduce our distributed computation framework, and review the gradient-tracking scheme proposed in [22].

2.1 Optimization Problem and Assumptions

We consider a system composed of N agents which are willing to collaborate to solve an optimization program formulated over the entire system. The agents shall agree on a common value for the decision vector $x \in \mathbb{R}^n$ which has to be set to minimize the sum of the agents local objective functions $f_i : \mathbb{R}^n \to \mathbb{R}$, while satisfying each agent local constraint set $X_i \subseteq \mathbb{R}^n$. Formally, we address the following mathematical program

$$\begin{aligned} \min_{x} \quad & \sum_{i=1}^{N} f_i(x) \\ \text{subject to:} \quad & x \in \bigcap_{i=1}^{N} X_i. \end{aligned} \tag{$\mathcal{P}$}$$

We impose the following assumption on $\mathcal{P}$.

Assumption 1 (Convexity and well-posedness) *For all $i = 1, \dots, N$, the function f_i is convex and the set X_i is convex and closed. Moreover, the set $\cap_{i=1}^N \operatorname{relint}(X_i)$ is non-empty.* □

Note that we do not require the local constraint sets X_i, $i = 1, \dots, N$, to be neither bounded nor equal to each other. The following assumption guarantees that the minimum of $\mathcal{P}$ exists and is achieved.

Assumption 2 (Existence of optimal solution) *Problem $\mathcal{P}$ admits an optimal solution $x^\star$.* □

For ease of exposition, let us define for each agent i the local extended real-valued function $\varphi_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ as the sum of its objective function $f_i(x)$ with the indicator function $\mathcal{I}_{X_i}(x)$ of its constraint set X_i, i.e., $\varphi_i(x) = f_i(x) + \mathcal{I}_{X_i}(x)$. We can then equivalently rewrite $\mathcal{P}$ as

$$\min_x \ \varphi(x) = \sum_{i=1}^{N} \varphi_i(x). \qquad (\mathcal{P})$$

Assumption 1 implicitly requires each X_i to be non-empty which, together with convexity, makes φ_i a proper convex function, see [26, p. 24]. Under the additional closedness requirement of Assumption 1, each φ_i is also closed, see [26, p. 52]. Assumption 2 implies that $\bigcap_{i=1}^N X_i$ is non-empty and hence also φ is proper, [26, p. 24]. Closedness of φ directly follows from [26, Theorem 9.3].

2.2 Proximal Minimization Algorithm

An iterative method to compute an optimal solution of $\mathcal{P}$ in a centralized way is given by the proximal-minimization algorithm (also known as proximal point algorithm), [3, Chapter 5], which prescribes to update a tentative solution z_k according to the following recursion

$$z_{k+1} = \operatorname*{argmin}_x \ \varphi(x) + \frac{1}{2c}\|x - z_k\|^2, \qquad (1)$$

for any $c > 0$.

Under Assumptions 1 and 2, by [3, Proposition 5.1.3], the sequence $\{z_k\}_{k\geq 0}$ generated by (1) is guaranteed to converge to some optimal solution of $\mathcal{P}$.

Interestingly, [3, Proposition 5.1.1] shows that iteration (1) is equivalent to

$$z_{k+1} = z_k - c\, h_{k+1} \qquad (2a)$$
$$h_{k+1} \in \partial\varphi(z_{k+1}), \qquad (2b)$$

which provides us with a gradient interpretation of the proximal iteration in (1), where the old estimate z_k is updated with a subgradient of φ computed at z_{k+1}. Clearly, (2) cannot be implemented as-is because we would need to know z_{k+1} to compute h_{k+1}.

We shall stress that, differently from the standard gradient method, convergence is guaranteed for a constant step-size c even when φ is not smooth, and we will see that this difference has a direct counterpart in the distributed framework considered in this paper.

2.3 Distributed Computation Framework

At each iteration k, we assume that the N agents can communicate with each other according to a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, N\}$ is the set of nodes, each node representing one agent, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, representing the communication links. The presence of edge (i, j) in $\mathcal{E}$ models the fact that agent i receives information from agent j. We assume that the communication graph is static (i.e., fixed across iterations) and, consequently, $\mathcal{E}$ does not depend on the iteration index k. We denote by $\mathcal{N}_i = \{j \in \mathcal{V} : \ (i, j) \in \mathcal{E}\}$ the set of neighbors of agent i, assuming that $(i, i) \in \mathcal{E}$ for all $i = 1, \dots, N$. We then impose the following connectivity property on $\mathcal{G}$.

Assumption 3 (Connectivity) *The graph $\mathcal{G}$ is undirected and connected, i.e., $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$ and for every pair of vertices in $\mathcal{V}$ there exists a path of edges in $\mathcal{E}$ that connects them.* □

For each edge $(i, j) \in \mathcal{E}$, let us associate a weight w_{ij} measuring how much agent i values the information received by agent j. If there is no communication link between agent i and j (i.e., $(i, j) \notin \mathcal{E}$), then we set $w_{ij} = 0$. We impose the following assumption on the network weights.

Assumption 4 (Balanced information exchange) *For all $i, j = 1, \dots, N$, $w_{ij} \in [0, 1)$ and $w_{ij} = w_{ji}$. Furthermore*

- $\sum_{i=1}^N w_{ij} = 1$ *for all* $j = 1, \dots, N$,
- $\sum_{j=1}^N w_{ij} = 1$ *for all* $i = 1, \dots, N$,

and $w_{ij} > 0$ if and only if $(i, j) \in \mathcal{E}$. □

Let $\mathcal{W} \in \mathbb{R}^{N\times N}$ be the matrix whose (i, j)-th entry is w_{ij}, often referred to as the *consensus matrix*. Assumption 4 translates into requiring $\mathcal{W}$ to be symmetric and doubly stochastic, i.e., $\mathcal{W} = \mathcal{W}^\top$ and $\mathcal{W}\mathbb{1}_N = \mathcal{W}^\top\mathbb{1}_N = \mathbb{1}_N$. We should point out that Assumptions 3 and 4 are common in the consensus-based distributed optimization literature, see, e.g., [23,24], and can be relaxed leveraging the push-sum protocol proposed in [7].

Finally, we impose the following additional assumption on the consensus matrix.

Assumption 5 *$\mathcal{W}$ is a positive semi-definite matrix.* □

Remark 1 *Note that, this assumption is not too restrictive as it can be enforced starting from any matrix $\mathcal{W}'$ satisfying Assumption 4 and letting the agents construct (in a distributed way) the matrix $\mathcal{W} = \frac{1}{2}(I + \mathcal{W}')$. It is worth mentioning that some other approaches like [15, 28, 29] directly use $\frac{1}{2}(I + \mathcal{W}')$ inside the algorithm instead of assuming the consensus matrix be positive (semi)definite. We opted for this assumption to keep the notation light. Lastly, we also refer the reader to [8, Section 3.3.2] for an alternative way of satisfying Assumption 5.*

2.4 Gradient-Tracking

Under the additional assumptions that the problem is unconstrained (i.e., $X_i = \mathbb{R}^n$ for all $i = 1, \dots, N$) and the local cost functions f_i are smooth (i.e., differentiable with Lipschitz continuous gradient) one can solve $\mathcal{P}$ in a distributed way by means of the gradient-tracking scheme introduced in [22, Algorithm 1] and known as DIGing.

In DIGing, a generic agent i stores a local estimate $x_{i,k}$ of the common decision vector x and a local estimate $g_{i,k}$ of the network average gradient $\frac{1}{N}\sum_{i=1}^N \nabla f_i(x_{i,k})$ and updates them according to the following steps

$$x_{i,k+1} = \sum_{j\in\mathcal{N}_i} w_{ij} x_{j,k} - c\, g_{i,k} \tag{3a}$$

$$g_{i,k+1} = \sum_{j\in\mathcal{N}_i} w_{ij} g_{j,k} + \nabla f_i(x_{i,k+1}) - \nabla f_i(x_{i,k}), \tag{3b}$$

with $g_{i,0} = \nabla f_i(x_{i,0})$, for all $i = 1, \dots, N$. Note that under the unconstrained and smoothness assumptions $\varphi_i = f_i$ for all $i = 1, \dots, N$.

In (3b) agent i updates its own estimate $g_{i,k}$ of the global quantity $\frac{1}{N}\sum_{i=1}^N \nabla f_i(x_{i,k})$ according to a dynamic average consensus mechanism, [13,37], while in (3a) agent i performs a gradient update along $g_{i,k}$ starting from the average between its own tentative solution $x_{i,k}$ and that of its neighbors.

In (3), $g_{i,k}$ acts as a *distributed tracker* of the (time-varying) signal $\frac{1}{N}\sum_{i=1}^N \nabla f_i(x_{i,k})$ and steers the agents to take a common update direction in (3a), while the averaging part of (3a) steers the agents towards consensus on a common decision vector.

By carefully selecting the step-size coefficient $c > 0$, DIGing is guaranteed to converge at a $O(1/k)$ rate for smooth local objective functions, [25], and at a linear rate if we further assume the local objective functions to be strongly convex, [22, 25].

3 Proximal-Tracking Distributed Algorithm

In this section we propose our novel distributed optimization algorithm for decision-coupled problems. We show how we derived it starting from the DIGing and the proximal-minimization algorithm and we analyze its convergence properties.

3.1 Algorithm Derivation and Interpretation

Let us first consider the DIGing algorithm in (3) and show how to modify it to fit the gradient interpretation in (2) of the proximal algorithm in (1). To this end we first move from gradients to subgradients by replacing $\nabla f_i(x_{i,k})$ in (3b) with a vector $v_{i,k} \in \partial\varphi_i(x_{i,k})$ like in (2b), where we also used φ_i in place of f_i to allow also for local constraints X_i. Then we change update (3a) using $g_{i,k+1}$ in place of $g_{i,k}$ to mimic (2a). The distributed counterpart of (2) then reads as

$$x_{i,k+1} = \sum_{j\in\mathcal{N}_i} w_{ij} x_{j,k} - c\, g_{i,k+1} \tag{4a}$$

$$g_{i,k+1} = \sum_{j\in\mathcal{N}_i} w_{ij} g_{j,k} + v_{i,k+1} - v_{i,k} \tag{4b}$$

$$v_{i,k+1} \in \partial\varphi_i(x_{i,k+1}), \tag{4c}$$

with $g_{i,0} = v_{i,0}$ and $v_{i,0} \in \partial\varphi_i(x_{i,0})$, for all $i = 1, \dots, N$. Similarly to the centralized counterpart we discussed in Section 2.2, the steps in (4) are not implementable as-is. In the following, we show how to manipulate (4) to obtain the proposed Proximal-Tracking distributed algorithm.

To ease the notation, let us define $\xi_{i,k} = \sum_{j\in\mathcal{N}_i} w_{ij} x_{j,k}$ and $\gamma_{i,k} = \sum_{j\in\mathcal{N}_i} w_{ij} g_{j,k}$. By using (4b) in (4a) we obtain the following identity

$$0 = v_{i,k+1} + (\gamma_{i,k} - v_{i,k}) + \tfrac{1}{c}(x_{i,k+1} - \xi_{i,k}), \tag{5}$$

which, together with $v_{i,k+1} \in \partial\varphi_i(x_{i,k+1})$ yields

$$0 \in \partial\varphi_i(x_{i,k+1}) \oplus \{(\gamma_{i,k} - v_{i,k})\} \oplus \{\tfrac{1}{c}(x_{i,k+1} - \xi_{i,k})\}. \tag{6}$$

By interpreting $(\gamma_{i,k} - v_{i,k})$ as the gradient of $(\gamma_{i,k} - v_{i,k})^\top x$ and $\frac{1}{c}(x_{i,k+1} - \xi_{i,k})$ as the gradient of $\frac{1}{2c}\|x - \xi_{i,k}\|^2$ both evaluated at $x_{i,k+1}$, condition (6) is equivalent to

$$\begin{aligned} 0 \in \partial\varphi_i(x_{i,k+1}) &\oplus \partial\left((\gamma_{i,k} - v_{i,k})^\top x\right)(x_{i,k+1}) \\ &\oplus \partial\left(\tfrac{1}{2c}\|x - \xi_{i,k}\|^2\right)(x_{i,k+1}). \end{aligned} \tag{7}$$

Under Assumption 1, by [26, Theorem 23.8], condition (7) is equivalent to

$$0 \in \partial\left(\varphi_i(x) + (\gamma_{i,k} - v_{i,k})^\top x + \tfrac{1}{2c}\|x - \xi_{i,k}\|^2\right)(x_{i,k+1}),$$

which is an optimality condition for $x_{i,k+1}$ and can be expressed as

$$x_{i,k+1} \in \operatorname*{argmin}_{x_i} \varphi_i(x_i) + (\gamma_{i,k} - v_{i,k})^\top x_i + \tfrac{1}{2c}\|x_i - \xi_{i,k}\|^2,$$

or, equivalently,

$$x_{i,k+1} \in \operatorname*{argmin}_{x_i\in X_i} f_i(x_i) + (\gamma_{i,k} - v_{i,k})^\top x_i + \tfrac{1}{2c}\|x_i - \xi_{i,k}\|^2.$$

Since the previous optimization problem depends on

Algorithm 1 Proximal-Tracking

1: **Initialization**
2: $x_{i,0} \in \mathbb{R}^n$, $v_{i,0} \in \mathbb{R}^n$, $g_{i,0} = v_{i,0}$
3: **For each iteration** k **do**
4: $\xi_{i,k} = \sum_{j \in \mathcal{N}_i} w_{ij}\, x_{j,k}$
5: $\gamma_{i,k} = \sum_{j \in \mathcal{N}_i} w_{ij}\, g_{j,k}$
6: $x_{i,k+1} = \underset{x_i \in X_i}{\operatorname{argmin}} \Big\{ f_i(x_i) + (\gamma_{i,k} - v_{i,k})^\top x_i + \frac{1}{2c}\|x_i - \xi_{i,k}\|^2 \Big\}$
7: $v_{i,k+1} = \frac{1}{c}(\xi_{i,k} - x_{i,k+1}) + (v_{i,k} - \gamma_{i,k})$
8: $g_{i,k+1} = \gamma_{i,k} + v_{i,k+1} - v_{i,k}$
9: $k \leftarrow k + 1$

quantities at iteration k only, $x_{i,k+1}$ can now be computed. By means of (5) we can compute $v_{i,k+1}$ as

$$v_{i,k+1} = \tfrac{1}{c}(\xi_{i,k} - x_{i,k+1}) + (v_{i,k} - \gamma_{i,k})$$

and finally $g_{i,k+1}$ is computed using (4b).

The resulting Proximal-Tracking distributed algorithm is summarized in Algorithm 1 from the perspective of agent i.

First of all we shall stress that all steps in Algorithm 1 are fully distributed, as they use quantities either collected by agent i from its neighbors (cf. Steps 4 and 5) or locally available to agent i (cf. Steps 6-8).

Differently from the gradient-tracking scheme in (3), in Algorithm 1 each agent stores and updates tree quantities: a tentative solution $x_{i,k}$ for $\mathcal{P}$, a subgradient $v_{i,k}$ of φ_i, which encodes both the local cost function f_i and the local constraints X_i, and a local estimate (tracker) $g_{i,k}$ of the global quantity $\frac{1}{N}\sum_{i=1}^N v_{i,k}$.

At the beginning of each iteration, agent i constructs the averages $\xi_{i,k}$ and $\gamma_{i,k}$ of its current tentative solution $x_{i,k}$ and tracker variable $v_{i,k}$ with the corresponding quantities of its neighbors (cf. Steps 4 and 5). Then, it computes its new tentative solution $x_{i,k+1}$ by minimizing, subject to its local constraints X_i, an objective function composed by three terms: its own local objective f_i, a quadratic term that penalizes the distance between the new solution and the neighbor average $\xi_{i,k}$ (similarly to the work in [18]), and a linear correction term which counteract the "pull" of $f_i(x_i)$ with the term $-v_{i,k}^\top x_i$ (containing the subgradient $v_{i,k}$ at the past iteration) and pushes towards the direction given by $\gamma_{i,k}$, which is an estimate of the network average subgradient $\frac{1}{N}\sum_{i=1}^N v_{i,k}$ (cf. Step 6). Once $x_{i,k+1}$ is obtained, agent i computes the value of the subgradient of φ_i at $x_{i,k+1}$ (cf. Step 7) and then uses this subgradient to update the tracker variable $g_{i,k+1}$ (cf. Step 8).

Note that even if each agent stores and updates three quantities, it only shares with the neighbors two of them: the tentative solution $x_{i,k}$ and the tracker $g_{i,k}$. The local gradient $v_{i,k}$ remains instead a private information, like the local objective function f_i and the local constraints X_i. In terms of communication load, Proximal-Tracking is therefore equivalent to the gradient-tracking scheme in (3).

Consistently with other approaches leveraging a dynamic average consensus scheme, the correct initialization $g_{i,0} = v_{i,0}$ of the tracker variable $g_{i,k}$ is crucial for Proximal-Tracking to work, see [6, 22, 25, 31, 33, 36]. As for the initialization of $x_{i,k}$ and $v_{i,k}$, the user can select any $x_{i,0} \in \mathbb{R}^n$ and any $v_{i,0} \in \mathbb{R}^n$. If well-defined, a sensible value for the initialization of the tentative solution is $x_{i,0} \in \operatorname{argmin}_{x_i \in X_i} f_i(x_i)$, while $v_{i,k}$ can be initialized as $v_{i,0} = 0$ to have $v_{i,0} \in \partial f_i(x_{i,0})$.

Finally, the parameter $c > 0$ in Step 6 is constant and is similar to the step-size of gradient-tracking schemes like (3), with the crucial difference that its value in Proximal-Tracking can be arbitrary. This fact is actually the distributed counterpart of the available step-size/penalty parameter choices between the centralized gradient method, where the step-size has to be chosen based on the Lipschitz constant of the gradient of the objective function, and the centralized proximal minimization algorithm, where the penalty parameter is freely tunable. Moreover, leveraging the proximal perspective, we are also able to handle non-smooth functions and the presence of different local constraints sets per agent, similarly to [15, 29].

We conclude this section with the main theoretical result, which establishes the convergence of the proposed Proximal-Tracking to an optimal solution of $\mathcal{P}$.

Theorem 1 (Optimality) *Under Assumptions 1-4, all sequences $\{x_{i,k}\}_{k\geq 0}$, for all $i = 1, \ldots, N$, generated by Proximal-Tracking converge to the same optimal solution $x^\star$ of $\mathcal{P}$ and, for each $i = 1, \ldots, N$, the sequence $\{v_{i,k}\}_{k\geq 0}$ converges to one element of $\partial\varphi_i(x^\star)$.* □

When implementing Algorithm 1, some stopping criterion has to be adopted. Since deriving a stopping criterion based on a desired accuracy level would entail studying the convergence rate of the proposed algorithm, and the derivation of an explicit convergence rate is left as a future work, we suggest to stop Algorithm 1 after a (user-chosen) maximum number of iterations.

3.2 Algorithm Analysis

In this section we guide the reader through the proof of Theorem 1.

3.2.1 Aggregate Reformulation of Proximal-Tracking

To ease the analysis and the notation, let us reformulate in a compact form the steps collectively performed by all agents running Proximal-Tracking in parallel.

To this end we define the following bold symbols, which are network-wide vectors obtained by stacking the corresponding (non-bold) quantities of all agents:

$$\begin{aligned} \boldsymbol{x}_k &= [x_{1,k}^\top \cdots x_{N,k}^\top]^\top, \qquad \boldsymbol{\xi}_k = [\xi_{1,k}^\top \cdots \xi_{N,k}^\top]^\top, \\ \boldsymbol{g}_k &= [g_{1,k}^\top \cdots g_{N,k}^\top]^\top, \qquad \boldsymbol{\gamma}_k = [\gamma_{1,k}^\top \cdots \gamma_{N,k}^\top]^\top, \\ \boldsymbol{v}_k &= [v_{1,k}^\top \cdots v_{N,k}^\top]^\top. \end{aligned}$$

Leveraging the equivalence between Algorithm 1 and (4) together with the network-wide notation, the sequences generated by Proximal-Tracking running over the whole multi-agent network satisfy the following identities for all $k \geq 0$

$$\boldsymbol{x}_{k+1} = W\boldsymbol{x}_k - c\,\boldsymbol{g}_{k+1} \tag{8a}$$
$$\boldsymbol{g}_{k+1} = W\boldsymbol{g}_k + \boldsymbol{v}_{k+1} - \boldsymbol{v}_k, \tag{8b}$$
$$\boldsymbol{v}_{k+1} \in \partial\boldsymbol{\varphi}(\boldsymbol{x}_{k+1}), \tag{8c}$$

where $\boldsymbol{\varphi}(\boldsymbol{x}) = \sum_{i=1}^N \varphi_i(x_i)$ with $\boldsymbol{x} = [x_1^\top \cdots x_N^\top]^\top$, $W = \mathcal{W} \otimes I_n$, and we used $\boldsymbol{\xi}_k = W\boldsymbol{x}_k$ and $\boldsymbol{\gamma}_k = W\boldsymbol{g}_k$, which represent the network-wide formulations of the two average terms in (4a) and (4b), respectively. Note that $\boldsymbol{\varphi} : \mathbb{R}^{Nn} \to \mathbb{R} \cup \{+\infty\}$ while $\varphi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and, hence, $\boldsymbol{\varphi}$ and φ are, in general, different objects, but if we consider a vector $\boldsymbol{x} = \mathbb{1}_N \otimes x$, then $\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{\varphi}(\mathbb{1}_N \otimes x) = \sum_{i=1}^N \varphi_i(x) = \varphi(x)$. Note also that $\partial\boldsymbol{\varphi}(\boldsymbol{x}_k) = \partial\varphi_1(x_{1,k}) \times \cdots \times \partial\varphi_N(x_{N,k})$. Finally, also $\boldsymbol{\varphi}$ is a proper closed convex function, see [26, p. 24 (proper) and Theorem 9.3 (closed)].

In (8), identities (8a) and (8b) are linear conditions on the sequences generated by Algorithm 1, whereas condition (8c) is nonlinear. To prove Theorem 1 we will study the two sets of conditions separately: we will use the linear identities to establish relations between two consecutive iterations of Algorithm 1, and then leverage the nonlinear condition to turn such relations into an inequality involving a Lyapunov function.

3.2.2 Properties of the Consensus Matrix

Before starting, let us recall some properties and introduce some identities involving the consensus matrix. Since we will work with the aggregate reformulation in (8), we introduce the matrices $W_\infty = (\frac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top) \otimes I_n$, and $\tilde{W} = W - W_\infty$, and we state the results directly for these extended matrices.

Lemma 1 (Properties of the Consensus Matrix) *Under Assumption 4 we have the following properties for matrices W, W_∞, and $\tilde{W}$:*

$$W_\infty \boldsymbol{z} = \bar{\boldsymbol{z}}, \tag{9a}$$
$$W_\infty W = W W_\infty = W_\infty, \tag{9b}$$
$$W(\mathbb{1}_N \otimes y) = \mathbb{1}_N \otimes y, \tag{9c}$$
$$W_\infty(\mathbb{1}_N \otimes y) = \mathbb{1}_N \otimes y, \tag{9d}$$
$$W_\infty(\boldsymbol{z} - \bar{\boldsymbol{z}}) = 0, \tag{9e}$$

for all $y \in \mathbb{R}^n$, $\boldsymbol{z} = [z_1^\top \cdots z_N^\top]^\top$, $z_i \in \mathbb{R}^n$ $i = 1, \ldots, N$, $\bar{z} = \frac{1}{N}\sum_{i=1}^N z_i$, and $\bar{\boldsymbol{z}} = \mathbb{1}_N \otimes \bar{z}$. Adding also Assumption 3 yields

$$\rho(\tilde{W}) < 1, \tag{9f}$$
$$(I - \tilde{W})^{-1}(I - W) = I - W_\infty. \tag{9g}$$

PROOF. *See the Appendix.* □

3.2.3 Averages Properties

We start by stating some important properties regarding the network averages of the sequences in (8). Define

$$\bar{x}_k = \frac{1}{N}\sum_{i=1}^N x_{i,k}, \quad \bar{g}_k = \frac{1}{N}\sum_{i=1}^N g_{i,k}, \quad \bar{v}_k = \frac{1}{N}\sum_{i=1}^N v_{i,k}.$$

For convenience, we also introduce the following vectors, which contains N copies of the respective average quantities

$$\begin{aligned} \bar{\boldsymbol{x}}_k &= \mathbb{1}_N \otimes \bar{x}_k \quad & \bar{\boldsymbol{g}}_k &= \mathbb{1}_N \otimes \bar{g}_k \quad & \bar{\boldsymbol{v}}_k &= \mathbb{1}_N \otimes \bar{v}_k \\ &= W_\infty \boldsymbol{x}_k, & &= W_\infty \boldsymbol{g}_k, & &= W_\infty \boldsymbol{v}_k, \end{aligned}$$

where the second equality of each term is due to (9a).

We are now ready to state the following results.

Lemma 2 (Average Primal Update) *Under Assumption 4, for all $k \geq 0$, we have*

$$\bar{\boldsymbol{x}}_{k+1} = \bar{\boldsymbol{x}}_k - c\,\bar{\boldsymbol{g}}_{k+1}. \tag{10}$$

PROOF. *The desired result is obtained by left-multiplying* (8a) *by W_∞, using the identity $W_\infty W = W_\infty$ in* (9b) *and the definition of $\bar{\boldsymbol{x}}_k$ and $\bar{\boldsymbol{g}}_k$.* □

Lemma 3 (Tracking Property) *Under Assumption 4, for all $k \geq 0$, we have*

$$\bar{\boldsymbol{g}}_k = \bar{\boldsymbol{v}}_k. \tag{11}$$

PROOF. *The result can be proven by induction. At $k = 0$ we have $\boldsymbol{g}_0 = \boldsymbol{v}_0$ (cf. Step 2 in Algorithm 1) and, hence, $\bar{\boldsymbol{g}}_0 = W_\infty \boldsymbol{g}_0 = W_\infty \boldsymbol{v}_0 = \bar{\boldsymbol{v}}_0$. Assume now that (11) holds for some $k > 0$. Then, we can show that it holds for $k+1$ (thus concluding the proof by induction) as follows*

$$\begin{aligned}\bar{\boldsymbol{g}}_{k+1} &= W_\infty \boldsymbol{g}_{k+1} \\ &\overset{(a)}{=} W_\infty (W \boldsymbol{g}_k + \boldsymbol{v}_{k+1} - \boldsymbol{v}_k) \\ &\overset{(9b)}{=} W_\infty \boldsymbol{g}_k + W_\infty \boldsymbol{v}_{k+1} - W_\infty \boldsymbol{v}_k \\ &\overset{(b)}{=} \bar{\boldsymbol{g}}_k + \bar{\boldsymbol{v}}_{k+1} - \bar{\boldsymbol{v}}_k \\ &\overset{(c)}{=} \bar{\boldsymbol{v}}_{k+1},\end{aligned}$$

where (a) is obtained by left-multiplying (8b) by W_∞, (b) using the definition of $\bar{\boldsymbol{g}}_k$ and $\bar{\boldsymbol{v}}_k$, and (c) using the induction step. □

Combining Lemmas 2 and 3 we see how the proposed Proximal-Tracking is mimicking the centralized proximal step in (2), the only difference being that $\bar{\boldsymbol{x}}_{k+1}$ is updated using the average $\bar{\boldsymbol{v}}_{k+1}$ of the subgradients $\boldsymbol{v}_{k+1} \in \partial \boldsymbol{\varphi}(\boldsymbol{x}_{k+1})$ (cf. (8c)) instead of a common subgradient from $\partial \varphi(\bar{x}_{k+1})$.

3.2.4 Convergence Analysis: Optimality Relation

Under Assumption 2 we know that $\mathcal{P}$ admits at least one optimal solution $x^\star$. To ease the notation, let us introduce the corresponding stacked vector $\boldsymbol{x}^\star = \mathbb{1}_N \otimes x^\star$, and the optimality error $\boldsymbol{e}_k^\star = \bar{\boldsymbol{x}}_k - \boldsymbol{x}^\star$.

We then start our convergence analysis from the dynamics of the optimality error

$$\begin{aligned}\boldsymbol{e}_{k+1}^\star &= \bar{\boldsymbol{x}}_{k+1} - \boldsymbol{x}^\star \\ &\overset{(10)}{=} \bar{\boldsymbol{x}}_k - \boldsymbol{x}^\star - c\,\bar{\boldsymbol{g}}_{k+1} \\ &= \boldsymbol{e}_k^\star - c\,\bar{\boldsymbol{g}}_{k+1}.\end{aligned}$$

If we now bring the $-c\,\boldsymbol{g}_{k+1}$ term to the left-hand side and take the square, we obtain

$$\|\boldsymbol{e}_{k+1}^\star\|^2 + 2\boldsymbol{e}_{k+1}^{\star\top} c\,\bar{\boldsymbol{g}}_{k+1} + \|c\,\bar{\boldsymbol{g}}_{k+1}\|^2 = \|\boldsymbol{e}_k^\star\|^2, \quad (12)$$

which constitutes our first building block in the convergence analysis of Proximal-Tracking.

If we substitute $c\,\bar{\boldsymbol{g}}_{k+1} = \bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}}_{k+1}$ (cf. (10)), then relation (12) is similar to the three-term inequality in [3, Proposition 5.1.2] used to prove convergence of the centralized proximal minimization algorithm. However, while in the centralized case the term $2\boldsymbol{e}_{k+1}^{\star\top} c\,\bar{\boldsymbol{g}}_{k+1}$ can be proven to be non-negative, this is not the case for the distributed algorithm.

3.2.5 Convergence Analysis: Error Relations

Since (12) involves average quantities only, the next step is to study the so-called consensus error, i.e., the distance of the agents local estimates from the corresponding network averages. To this end, let $\boldsymbol{e}_k^{\boldsymbol{x}} = \boldsymbol{x}_k - \bar{\boldsymbol{x}}_k$ and $\boldsymbol{e}_k^{\boldsymbol{g}} = c(\boldsymbol{g}_k - \bar{\boldsymbol{g}}_k)$, and compute

$$\begin{aligned}\boldsymbol{e}_{k+1}^{\boldsymbol{x}} &= \boldsymbol{x}_{k+1} - \bar{\boldsymbol{x}}_{k+1} \\ &\overset{(a)}{=} W\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k - c(\boldsymbol{g}_{k+1} - \bar{\boldsymbol{g}}_{k+1}) \\ &\overset{(b)}{=} W\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k - \boldsymbol{e}_{k+1}^{\boldsymbol{g}} \\ &\overset{(c)}{=} W(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k) - \boldsymbol{e}_{k+1}^{\boldsymbol{g}} \\ &= W\boldsymbol{e}_k^{\boldsymbol{x}} - \boldsymbol{e}_{k+1}^{\boldsymbol{g}} \\ &\overset{(d)}{=} \tilde{W}\boldsymbol{e}_k^{\boldsymbol{x}} - \boldsymbol{e}_{k+1}^{\boldsymbol{g}}, \end{aligned} \quad (13)$$

where in (a) we used both (8a) and (10), in (b) the definition of $\boldsymbol{e}_{k+1}^{\boldsymbol{g}}$, in (c) we used $\bar{\boldsymbol{x}}_k = W\bar{\boldsymbol{x}}_k$ by (9c), and in (d) we subtracted $W_\infty \boldsymbol{e}_k^{\boldsymbol{x}}$, which is zero by (9e), together with the definition of $\tilde{W} = W - W_\infty$.

The next step is to analyze relation (8b). First, let us introduce what we will show to be the limiting value of the sequence $\{\boldsymbol{v}_k\}_{k\geq 0}$. Consider an optimal solution $x^\star$ of $\mathcal{P}$. Let $v_i^\star \in \partial \varphi_i(x^\star)$, $i = 1, \ldots, N$, and define the vector $\boldsymbol{v}^\star = [v_1^{\star\top} \cdots v_N^{\star\top}]^\top$. Under Assumption 1, by [26, Theorem 23.8],

$$\partial \varphi_1(x^\star) \oplus \cdots \oplus \partial \varphi_N(x^\star) = \partial \varphi(x^\star).$$

Since $x^\star \in \operatorname{argmin}_x \varphi(x)$, then $0 \in \partial \varphi(x^\star)$, meaning that we can choose the $v_i^\star$'s such that $\sum_{i=1}^N v_i^\star = 0$, or, compactly (cf. (9a)), such that

$$W_\infty \boldsymbol{v}^\star = 0. \quad (14)$$

Consider now relation (8b). If we bring the terms at $k+1$ on the same side and we subtract $\boldsymbol{v}^\star$ on both sides, we have

$$\begin{aligned}\boldsymbol{v}_{k+1} - \boldsymbol{v}^\star - \boldsymbol{g}_{k+1} &\overset{(8b)}{=} \boldsymbol{v}_k - \boldsymbol{v}^\star - W\boldsymbol{g}_k \\ &\overset{(a)}{=} \boldsymbol{v}_k - \boldsymbol{v}^\star - \boldsymbol{g}_k + (I - W)\boldsymbol{g}_k,\end{aligned}$$

where in (a) we added and subtracted the quantity $\boldsymbol{g}_k$. To make the consensus error $\boldsymbol{e}_k^{\boldsymbol{g}}$ appear as input, we left-multiply the previous relation by $c(I - \tilde{W})^{-1}$, thus obtaining

$$\begin{aligned}&c(I - \tilde{W})^{-1}(\boldsymbol{v}_{k+1} - \boldsymbol{v}^\star - \boldsymbol{g}_{k+1}) \\ &\quad = c(I - \tilde{W})^{-1}(\boldsymbol{v}_k - \boldsymbol{v}^\star - \boldsymbol{g}_k) + c(I - \tilde{W})^{-1}(I - W)\boldsymbol{g}_k \\ &\quad \overset{(9g)}{=} c(I - \tilde{W})^{-1}(\boldsymbol{v}_k - \boldsymbol{v}^\star - \boldsymbol{g}_k) + c(I - W_\infty)\boldsymbol{g}_k\end{aligned}$$

$$\overset{(a)}{=} c(I-\tilde{W})^{-1}(\boldsymbol{v}_k-\boldsymbol{v}^\star-\boldsymbol{g}_k)+\boldsymbol{e}_k^{\boldsymbol{g}}, \tag{15}$$

where (a) is due to (9a) together with the definition of $\boldsymbol{e}_k^{\boldsymbol{g}}$.

Recalling (13) and using the definition $\boldsymbol{u}_k = c(I-\tilde{W})^{-1}(\boldsymbol{v}_k-\boldsymbol{v}^\star-\boldsymbol{g}_k)$ in (15), we obtain the following relations involving the consensus errors $\boldsymbol{e}_k^{\boldsymbol{x}}$ and $\boldsymbol{e}_k^{\boldsymbol{g}}$ plus the additional term $\boldsymbol{u}_k$

$$\boldsymbol{e}_{k+1}^{\boldsymbol{x}}+\boldsymbol{e}_{k+1}^{\boldsymbol{g}}=\tilde{W}\boldsymbol{e}_k^{\boldsymbol{x}} \tag{16a}$$
$$\boldsymbol{u}_{k+1}=\boldsymbol{u}_k+\boldsymbol{e}_k^{\boldsymbol{g}}, \tag{16b}$$

which constrain the dynamics of the "out-of-average" components of the sequences generated by Algorithm 1 and constitute the second building block for our convergence analysis.

3.2.6 Convergence Analysis: Subgradient Condition

In this section we show how the subgradient condition (8c) can be leveraged to establish a link between optimality and error relations, thus setting the stage for the proof of Theorem 1.

By definition of subgradient (see [26, p. 214]), $\boldsymbol{v}_{k+1} \in \partial\boldsymbol{\varphi}(\boldsymbol{x}_{k+1})$ if and only if

$$\boldsymbol{\varphi}(\boldsymbol{x}) \geq \boldsymbol{\varphi}(\boldsymbol{x}_{k+1}) + \boldsymbol{v}_{k+1}^\top[\boldsymbol{x}-\boldsymbol{x}_{k+1}] \tag{17}$$

for all $\boldsymbol{x} \in \mathbb{R}^{Nn}$. Note that, for all $k \geq 0$, $x_{i,k+1} \in X_i$, which is non-empty under Assumption 1, hence $\varphi_i(x_{i,k+1}) < +\infty$ and $\boldsymbol{\varphi}(\boldsymbol{x}_{k+1}) < +\infty$ for all $k \geq 0$. Similarly, $\boldsymbol{v}^\star \in \partial\boldsymbol{\varphi}(\boldsymbol{x}^\star)$ if and only if

$$\boldsymbol{\varphi}(\boldsymbol{x}) \geq \boldsymbol{\varphi}(\boldsymbol{x}^\star) + \boldsymbol{v}^{\star\top}[\boldsymbol{x}-\boldsymbol{x}^\star] \tag{18}$$

for all $\boldsymbol{x} \in \mathbb{R}^{Nn}$, where $\boldsymbol{\varphi}(\boldsymbol{x}^\star) = \varphi(x^\star) < +\infty$ under Assumption 2. Setting $\boldsymbol{x} = \boldsymbol{x}^\star$ in (17) and $\boldsymbol{x} = \boldsymbol{x}_{k+1}$ in (18), multiplying by $c > 0$, and summing the two inequalities yields

$$c[\boldsymbol{v}_{k+1}-\boldsymbol{v}^\star]^\top[\boldsymbol{x}_{k+1}-\boldsymbol{x}^\star] \geq 0. \tag{19}$$

Noticing that

$$\boldsymbol{x}_{k+1}-\boldsymbol{x}^\star = \boldsymbol{x}_{k+1} \mp \bar{\boldsymbol{x}}_{k+1} - \boldsymbol{x}^\star = \boldsymbol{e}_{k+1}^{\boldsymbol{x}} + \boldsymbol{e}_{k+1}^\star$$

and that $\boldsymbol{e}_{k+1}^\star = W_\infty \boldsymbol{e}_{k+1}^\star$ by (9d), yields

$$c[\boldsymbol{v}_{k+1}-\boldsymbol{v}^\star]^\top W_\infty \boldsymbol{e}_{k+1}^\star + c[\boldsymbol{v}_{k+1}-\boldsymbol{v}^\star]^\top \boldsymbol{e}_{k+1}^{\boldsymbol{x}} \geq 0,$$

which can be further simplified in

$$c\,\bar{\boldsymbol{g}}_{k+1}^\top \boldsymbol{e}_{k+1}^\star + c[\boldsymbol{v}_{k+1}-\boldsymbol{v}^\star]^\top \boldsymbol{e}_{k+1}^{\boldsymbol{x}} \geq 0,$$

recalling that $W_\infty \boldsymbol{v}^\star \overset{(14)}{=} 0$ and $W_\infty \boldsymbol{v}_{k+1} \overset{(9a)}{=} \bar{\boldsymbol{v}}_{k+1} \overset{(11)}{=} \bar{\boldsymbol{g}}_{k+1}$. By definition of $\boldsymbol{u}_{k+1}$, $c[\boldsymbol{v}_{k+1}-\boldsymbol{v}^\star] = (I-\tilde{W})\boldsymbol{u}_{k+1} + c\,\boldsymbol{g}_{k+1}$, hence

$$c\,\bar{\boldsymbol{g}}_{k+1}^\top \boldsymbol{e}_{k+1}^\star + c\,\boldsymbol{g}_{k+1}^\top \boldsymbol{e}_{k+1}^{\boldsymbol{x}} + \boldsymbol{e}_{k+1}^{\boldsymbol{x}\top}(I-\tilde{W})\boldsymbol{u}_{k+1} \geq 0.$$

The inequality can be finally rewritten as

$$c\,\bar{\boldsymbol{g}}_{k+1}^\top \boldsymbol{e}_{k+1}^\star + \boldsymbol{e}_{k+1}^{\boldsymbol{g}\top} \boldsymbol{e}_{k+1}^{\boldsymbol{x}} + \boldsymbol{e}_{k+1}^{\boldsymbol{x}\top}(I-\tilde{W})\boldsymbol{u}_{k+1} \geq 0 \tag{20}$$

noticing that

$$\begin{aligned} c\,\boldsymbol{g}_{k+1}^\top \boldsymbol{e}_{k+1}^{\boldsymbol{x}} &\overset{(a)}{=} c\,\boldsymbol{g}_{k+1}^\top (I-W_\infty)\boldsymbol{e}_{k+1}^{\boldsymbol{x}} + c\,\boldsymbol{g}_{k+1}^\top W_\infty \boldsymbol{e}_{k+1}^{\boldsymbol{x}} \\ &\overset{(9e)}{=} c\,\boldsymbol{g}_{k+1}^\top (I-W_\infty)\boldsymbol{e}_{k+1}^{\boldsymbol{x}} + 0 \\ &\overset{(9a)}{=} c\,[\boldsymbol{g}_{k+1}-\bar{\boldsymbol{g}}_{k+1}]^\top \boldsymbol{e}_{k+1}^{\boldsymbol{x}} \\ &\overset{(b)}{=} \boldsymbol{e}_{k+1}^{\boldsymbol{g}\top} \boldsymbol{e}_{k+1}^{\boldsymbol{x}}, \end{aligned}$$

where in (a) we added and subtracted W_∞ within the inner product and in (b) we used the definition of $\boldsymbol{e}_{k+1}^{\boldsymbol{g}}$.

Relation (20) links the inner product $c\,\bar{\boldsymbol{g}}_{k+1}^\top \boldsymbol{e}_{k+1}^\star$ in (12) with the quantities appearing in the error relations (16), and constitutes the last building block for the convergence analysis of Proximal-Tracking.

3.3 Proof of Theorem 1

Consider the error relations (16) in the following (equivalent) matrix form

$$\underbrace{\begin{bmatrix} I & I & 0 \\ 0 & 0 & I \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} \boldsymbol{e}_{k+1}^{\boldsymbol{x}} \\ \boldsymbol{e}_{k+1}^{\boldsymbol{g}} \\ \boldsymbol{u}_{k+1} \end{bmatrix}}_{\boldsymbol{\zeta}_{k+1}} = \underbrace{\begin{bmatrix} \tilde{W} & 0 & 0 \\ 0 & I & I \end{bmatrix}}_{N} \underbrace{\begin{bmatrix} \boldsymbol{e}_k^{\boldsymbol{x}} \\ \boldsymbol{e}_k^{\boldsymbol{g}} \\ \boldsymbol{u}_k \end{bmatrix}}_{\boldsymbol{\zeta}_k}$$

from which we can build the following relation

$$\|\boldsymbol{\zeta}_{k+1}\|^2_{M^\top P M} = \|\boldsymbol{\zeta}_k\|^2_{N^\top P N}, \tag{21}$$

with

$$P = P^\top = \begin{bmatrix} 2I & I \\ I & I \end{bmatrix} \succ 0.$$

By (20) we have

$$-\boldsymbol{e}_{k+1}^{\boldsymbol{g}\top} \boldsymbol{e}_{k+1}^{\boldsymbol{x}} - \boldsymbol{e}_{k+1}^{\boldsymbol{x}\top}(I-\tilde{W})\boldsymbol{u}_{k+1} \leq c\,\bar{\boldsymbol{g}}_{k+1}^\top \boldsymbol{e}_{k+1}^\star,$$

which can be used in the optimality relation (12) to obtain

$$\|\boldsymbol{e}^\star_{k+1}\|^2 - 2\boldsymbol{e}^{\boldsymbol{g}\top}_{k+1}\boldsymbol{e}^{\boldsymbol{x}}_{k+1} - 2\boldsymbol{e}^{\boldsymbol{x}\top}_{k+1}(I-\tilde{W})\boldsymbol{u}_{k+1} \leq \|\boldsymbol{e}^\star_k\|^2 - \|c\,\bar{\boldsymbol{g}}_{k+1}\|^2,$$

or, compactly,

$$\|\boldsymbol{e}^\star_{k+1}\|^2 - \boldsymbol{\zeta}^\top_{k+1}C\boldsymbol{\zeta}_{k+1} \leq \|\boldsymbol{e}^\star_k\|^2 - \|c\,\bar{\boldsymbol{g}}_{k+1}\|^2, \quad (22)$$

with

$$C = \begin{bmatrix} 0 & I & I-\tilde{W} \\ I & 0 & 0 \\ I-\tilde{W} & 0 & 0 \end{bmatrix}$$

recalling that $\tilde{W}$ is symmetric under Assumption 4.

Summing (21) and (22) we have

$$\|\boldsymbol{e}^\star_{k+1}\|^2 + \|\boldsymbol{\zeta}_{k+1}\|^2_{M^\top PM-C} \leq \|\boldsymbol{e}^\star_k\|^2 + \|\boldsymbol{\zeta}_k\|^2_{N^\top PN} - \|c\,\bar{\boldsymbol{g}}_{k+1}\|^2, \quad (23)$$

and to show convergence we need to check if

$$M^\top PM - C \succ 0, \quad (24a)$$
$$M^\top PM - C - N^\top PN = Q \succeq 0. \quad (24b)$$

Let us start with condition (24a). By simple computations we get that

$$M^\top PM - C = \begin{bmatrix} 2I & I & \tilde{W} \\ I & 2I & I \\ \tilde{W} & I & I \end{bmatrix} \quad (25)$$

which, by the Schur complement lemma, is positive definite if and only if

$$\begin{cases} \begin{bmatrix} 2I & I \\ I & I \end{bmatrix} \succ 0 \\ 2I - \begin{bmatrix} I & \tilde{W} \end{bmatrix} \begin{bmatrix} 2I & I \\ I & I \end{bmatrix}^{-1} \begin{bmatrix} I \\ \tilde{W} \end{bmatrix} \succ 0. \end{cases}$$

The first condition is trivially satisfied, while the second condition is equivalent to

$$\begin{aligned} 2I - \begin{bmatrix} I & \tilde{W} \end{bmatrix} \begin{bmatrix} 2I & I \\ I & I \end{bmatrix}^{-1} \begin{bmatrix} I \\ \tilde{W} \end{bmatrix} &= 2I - \begin{bmatrix} I & \tilde{W} \end{bmatrix} \begin{bmatrix} I & -I \\ -I & 2I \end{bmatrix} \begin{bmatrix} I \\ \tilde{W} \end{bmatrix} \\ &= I + 2\tilde{W} - 2\tilde{W}^2. \end{aligned}$$

Since, under Assumption 5, $\tilde{W} \succeq 0$ and by (9f) $\rho(\tilde{W}) < 1$, then $\tilde{W} \succeq \tilde{W}^2$, and hence $I + 2\tilde{W} - 2\tilde{W}^2 \succeq I \succ 0$. Thus condition (24a) is satisfied.

Next, consider (24b). After simple computations we obtain the matrix

$$Q = \begin{bmatrix} 2(I-\tilde{W}^2) & I-\tilde{W} & 0 \\ I-\tilde{W} & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

which is positive semi-definite if and only if the reduced matrix

$$R = \begin{bmatrix} 2(I-\tilde{W}^2) & I-\tilde{W} \\ I-\tilde{W} & I \end{bmatrix}$$

is positive (semi-)definite. Using again the Schur complement lemma, $R \succ 0$ if and only if

$$\begin{cases} I \succ 0 \\ 2(I-\tilde{W}^2) - (I-\tilde{W})^2 \succ 0. \end{cases}$$

Also in this case, the first condition is trivially satisfied, while the second condition can be simplified as

$$\begin{aligned} 2(I-\tilde{W}^2) - (I-\tilde{W})^2 &= 2I - 2\tilde{W}^2 - I + 2\tilde{W} - \tilde{W}^2 \\ &= I + 2\tilde{W} - 3\tilde{W}^2. \end{aligned}$$

Since, under Assumption 5, $\tilde{W} \succeq 0$ and by (9f) $\rho(\tilde{W}) < 1$, then $I \succ \tilde{W} \succeq \tilde{W}^2$ and hence $I + 2\tilde{W} - 3\tilde{W}^2 \succeq I - \tilde{W}^2 \succ 0$, meaning that also condition (24b) is satisfied.

Combining (23) with (24) we obtain

$$\|\boldsymbol{e}^\star_{k+1}\|^2 + \|\boldsymbol{\zeta}_{k+1}\|^2_{P'} \leq \|\boldsymbol{e}^\star_k\|^2 + \|\boldsymbol{\zeta}_k\|^2_{P'} - \|c\,\bar{\boldsymbol{g}}_{k+1}\|^2 - \|\boldsymbol{\zeta}_k\|^2_Q,$$

with $P' = M^\top PM - C \succ 0$ and $Q \succeq 0$. If we define $\boldsymbol{e}_k = [\boldsymbol{e}^{\boldsymbol{x}\top}_k\ \boldsymbol{e}^{\boldsymbol{g}\top}_k]^\top$ we can also rewrite the previous inequality as

$$\|\boldsymbol{e}^\star_{k+1}\|^2 + \|\boldsymbol{\zeta}_{k+1}\|^2_{P'} \leq \|\boldsymbol{e}^\star_k\|^2 + \|\boldsymbol{\zeta}_k\|^2_{P'} - \|c\,\bar{\boldsymbol{g}}_{k+1}\|^2 - \|\boldsymbol{e}_k\|^2_R, \quad (26)$$

with $R \succ 0$, which tells us that $\|\boldsymbol{e}^\star_k\|^2 + \|\boldsymbol{\zeta}_k\|^2_{P'}$ is a Lyapunov function for the discrete-time dynamical system represented by Algorithm 1. From (26) we get that the sequence

$$\{\|\boldsymbol{e}^\star_k\|^2 + \|\boldsymbol{\zeta}_k\|^2_{P'}\}_{k\geq 0}$$

is non-increasing and, since it is also positive (recall that $P' \succ 0$), then it is convergent and, hence, bounded. Moreover, summing (26) over k from 0 to ∞ and rear-

ranging the terms, we obtain

$$\sum_{k=0}^{\infty} \|c\,\bar{\boldsymbol{g}}_{k+1}\|^2 + \|\boldsymbol{e}_k\|_R^2 \leq \|\boldsymbol{e}_0^\star\|^2 + \|\boldsymbol{\zeta}_0\|_{P'}^2 < \infty,$$

which, recalling that $R \succ 0$ and $c > 0$, implies

$$\lim_{k\to\infty} \|\bar{\boldsymbol{g}}_k\| = 0 \quad \overset{(11)}{\Longrightarrow} \quad \lim_{k\to\infty} \|\bar{\boldsymbol{v}}_k\| = 0, \tag{27a}$$

$$\lim_{k\to\infty} \|\boldsymbol{e}_k^{\boldsymbol{g}}\| = 0 \quad \overset{(27a)}{\Longrightarrow} \quad \lim_{k\to\infty} \|\boldsymbol{g}_k\| = 0, \tag{27b}$$

$$\lim_{k\to\infty} \|\boldsymbol{e}_k^{\boldsymbol{x}}\| = 0. \tag{27c}$$

Since $\{\|\boldsymbol{e}_k^\star\|^2 + \|\boldsymbol{\zeta}_k\|_{P'}^2\}_{k\geq 0}$ is bounded and both $\|\boldsymbol{e}_k^\star\|^2$ and $\|\boldsymbol{\zeta}_k\|_{P'}^2$ are non-negative, then also $\{\boldsymbol{e}_k^\star\}_{k\geq 0}$ and $\{\|\boldsymbol{\zeta}_k\|_{P'}\}_{k\geq 0}$ are bounded. Moreover, by definition of $\boldsymbol{\zeta}_k$ together with $P' \succ 0$, we have that $\{\boldsymbol{e}_k^{\boldsymbol{x}}\}_{k\geq 0}$, $\{\boldsymbol{e}_k^{\boldsymbol{g}}\}_{k\geq 0}$, and $\{\boldsymbol{u}_k\}_{k\geq 0}$ are all bounded. By boundedness of $\{\boldsymbol{u}_k\}_{k\geq 0}$ together with (27b) and (27c), we have that $\lim_{k\to\infty} \|\boldsymbol{e}_k^\star\|^2 + \|\boldsymbol{\zeta}_k\|_{P'}^2 = \lim_{k\to\infty} \|\boldsymbol{e}_k^\star\|^2 + \|\boldsymbol{u}_k\|^2$ (recall that P' equals $M^\top P M - C$ defined in (25)). Since $\boldsymbol{u}_k = c(I - \tilde{W})^{-1}[\boldsymbol{v}_k - \boldsymbol{v}^\star - \boldsymbol{g}_k]$, using (27b), we obtain

$$\begin{aligned}\lim_{k\to\infty} \|\boldsymbol{e}_k^\star\|^2 + \|\boldsymbol{\zeta}_k\|_{P'}^2 &= \lim_{k\to\infty} \|\boldsymbol{e}_k^\star\|^2 + \|\boldsymbol{u}_k\|^2 \\ &= \lim_{k\to\infty} \|\boldsymbol{e}_k^\star\|^2 + \|c(I - \tilde{W})^{-1}[\boldsymbol{v}_k - \boldsymbol{v}^\star]\|^2,\end{aligned}$$

meaning that also the sequence

$$\{\|\boldsymbol{e}_k^\star\|^2 + \|c(I - \tilde{W})^{-1}[\boldsymbol{v}_k - \boldsymbol{v}^\star]\|^2\}_{k\geq 0}$$

is convergent. Moreover, from boundedness of $\{\boldsymbol{u}_k\}_{k\geq 0}$ and $(I - \tilde{W})^{-1}$ being non-singular we have that also $\{c[\boldsymbol{v}_k - \boldsymbol{v}^\star]\}_{k\geq 0}$ is bounded. By definition of $\boldsymbol{e}_k^\star = \bar{\boldsymbol{x}}_k - \boldsymbol{x}^\star$ and since $\boldsymbol{x}^\star$ is fixed, we have that also the sequence $\{\bar{\boldsymbol{x}}_k\}_{k\geq 0}$ is bounded. Similarly, since $\boldsymbol{v}^\star$ is fixed, then also the sequence $\{\boldsymbol{v}_k\}_{k\geq 0}$ is bounded. Finally, by boundedness of $\{\bar{\boldsymbol{x}}_k\}_{k\geq 0}$ and (27c), we infer that the sequence $\{\boldsymbol{x}_k\}_{k\geq 0}$ is also bounded.

Up to now, we showed that the agents estimates $x_{i,k}$ and $g_{i,k}$ reach consensus on $\bar{x}_k$ and $\bar{g}_k$ respectively (cf. (27c) and (27b)), and that Algorithm 1 is stable, meaning that the generated sequences remain all bounded. Next, we shall prove that Proximal-Tracking converges to an optimal solution of $\mathcal{P}$.

Similarly to Section 3.2.6, by definition of subgradient (see [26, p. 214]), $\boldsymbol{v}_k \in \partial\boldsymbol{\varphi}(\boldsymbol{x}_k)$ if and only if

$$\boldsymbol{\varphi}(\boldsymbol{x}) \geq \boldsymbol{\varphi}(\boldsymbol{x}_k) + \boldsymbol{v}_k^\top[\boldsymbol{x} - \boldsymbol{x}_k] \tag{28}$$

for all $\boldsymbol{x} \in \mathbb{R}^{Nn}$ with $\boldsymbol{\varphi}(\boldsymbol{x}_k) < +\infty$. Setting $\boldsymbol{x} = \boldsymbol{x}^\star$ we obtain

$$\begin{aligned}\boldsymbol{\varphi}(\boldsymbol{x}^\star) &\geq \boldsymbol{\varphi}(\boldsymbol{x}_k) + \boldsymbol{v}_k^\top[\boldsymbol{x}^\star - \boldsymbol{x}_k] \\ &\overset{(a)}{=} \boldsymbol{\varphi}(\boldsymbol{x}_k) - \boldsymbol{v}_k^\top \boldsymbol{e}_k^{\boldsymbol{x}} - \boldsymbol{v}_k^\top \boldsymbol{e}_k^\star \\ &\overset{(b)}{=} \boldsymbol{\varphi}(\boldsymbol{x}_k) - \boldsymbol{v}_k^\top \boldsymbol{e}_k^{\boldsymbol{x}} - \bar{\boldsymbol{v}}_k^\top \boldsymbol{e}_k^\star\end{aligned} \tag{29}$$

where we used $\boldsymbol{x}_k - \boldsymbol{x}^\star = \boldsymbol{x}_k \mp \bar{\boldsymbol{x}}_k - \boldsymbol{x}^\star = \boldsymbol{e}_k^{\boldsymbol{x}} + \boldsymbol{e}_k^\star$ in (a) and $\boldsymbol{e}_k^\star \overset{(9d)}{=} W_\infty \boldsymbol{e}_k^\star$ together with $W_\infty \boldsymbol{v}_k \overset{(9a)}{=} \bar{\boldsymbol{v}}_k$ in (b). Using (27a) together with boundedness of $\{\boldsymbol{e}_k^\star\}_{k\geq 0}$ and (27c) together with boundedness of $\{\boldsymbol{v}_k\}_{k\geq 0}$ yields,

$$\lim_{k\to\infty} \bar{\boldsymbol{v}}_k^\top \boldsymbol{e}_k^\star = 0 \qquad \text{and} \qquad \lim_{k\to\infty} \boldsymbol{v}_k^\top \boldsymbol{e}_k^{\boldsymbol{x}} = 0,$$

which can be combined with the limit supremum on both sides of (29) to get

$$\begin{aligned}\limsup_{k\to\infty} \boldsymbol{\varphi}(\boldsymbol{x}_k) &= \limsup_{k\to\infty} \boldsymbol{\varphi}(\boldsymbol{x}_k) - \boldsymbol{v}_k^\top \boldsymbol{e}_k^{\boldsymbol{x}} - \bar{\boldsymbol{v}}_k^\top \boldsymbol{e}_k^\star \\ &\overset{(29)}{\leq} \boldsymbol{\varphi}(\boldsymbol{x}^\star) \\ &= \boldsymbol{\varphi}(\mathbb{1}_N \otimes x^\star) \\ &= f(x^\star).\end{aligned} \tag{30}$$

Since the sequence $\{(\boldsymbol{x}_k, \boldsymbol{v}_k)\}_{k\geq 0}$ is bounded, it admits a convergent subsequence $\{(\boldsymbol{x}_k, \boldsymbol{v}_k)\}_{k\in\mathcal{K}}$ with $\mathcal{K} \subseteq \mathbb{N}$. Let $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{v}})$ be its limit point. Since all limit points have to satisfy $\lim_{k\to\infty} \|\boldsymbol{e}_k^{\boldsymbol{x}}\| = 0$, then $\tilde{\boldsymbol{x}}$ is such that $\tilde{\boldsymbol{x}} = \mathbb{1}_N \otimes \tilde{x}$ and hence $\boldsymbol{\varphi}(\tilde{\boldsymbol{x}}) = \varphi(\tilde{x})$. Under Assumptions 1 and 2, $\boldsymbol{\varphi}$ is a closed proper convex function and, by [26, p. 52], it is lower semi-continuous, i.e.,

$$\boldsymbol{\varphi}(\boldsymbol{y}) = \liminf_{\boldsymbol{x}\to\boldsymbol{y}} \boldsymbol{\varphi}(\boldsymbol{x}).$$

We thus immediately have

$$\varphi(\tilde{x}) = \boldsymbol{\varphi}(\tilde{\boldsymbol{x}}) \overset{(a)}{=} \liminf_{\mathcal{K}\ni k\to\infty} \boldsymbol{\varphi}(\boldsymbol{x}_k) \overset{(30)}{\leq} f(x^\star),$$

where (a) is due to lower semi-continuity of $\boldsymbol{\varphi}$. This means that the limit point $\tilde{x}$ is an optimal solution for $\mathcal{P}$.

Now taking the limit of (28) across $\mathcal{K}$ we have

$$\begin{aligned}\boldsymbol{\varphi}(\boldsymbol{x}) &\overset{(28)}{\geq} \lim_{\mathcal{K}\ni k\to\infty} \boldsymbol{\varphi}(\boldsymbol{x}_k) + \lim_{\mathcal{K}\ni k\to\infty} \boldsymbol{v}_k^\top[\boldsymbol{x} - \boldsymbol{x}_k] \\ &= \lim_{\mathcal{K}\ni k\to\infty} \boldsymbol{\varphi}(\boldsymbol{x}_k) + \tilde{\boldsymbol{v}}^\top[\boldsymbol{x} - \tilde{\boldsymbol{x}}] \\ &\overset{(a)}{\geq} \boldsymbol{\varphi}(\tilde{\boldsymbol{x}}) + \tilde{\boldsymbol{v}}^\top[\boldsymbol{x} - \tilde{\boldsymbol{x}}],\end{aligned}$$

where (a) is due to $\boldsymbol{\varphi}$ being lower semi-continuous, which implies $\tilde{\boldsymbol{v}} \in \partial\boldsymbol{\varphi}(\tilde{\boldsymbol{x}})$. Moreover, since every

limit point of the sequence $\{\boldsymbol{v}_k\}_{k\geq 0}$ has to satisfy $\lim_{k\to\infty}\|\bar{\boldsymbol{v}}_k\| = 0$ (cf. (27a)), then $W_\infty\tilde{\boldsymbol{v}} = 0$. Since the sequence $\{\|\bar{\boldsymbol{x}}_k - \boldsymbol{x}^\star\|^2 + \|c(I-\tilde{W})^{-1}[\boldsymbol{v}_k - \boldsymbol{v}^\star]\|^2\}_{k\geq 0}$ (recall $\boldsymbol{e}_k^\star = \bar{\boldsymbol{x}}_k - \boldsymbol{x}^\star$) is convergent for any $\boldsymbol{x}^\star = \mathbb{1}_N \otimes x^\star$ and any $\boldsymbol{v}^\star$ such that $W_\infty \boldsymbol{v}^\star = 0$ (cf. (14)), then we can select $\boldsymbol{x}^\star = \mathbb{1}\otimes\tilde{x}$ and $\boldsymbol{v}^\star = \tilde{\boldsymbol{v}}$ to conclude that

$$\lim_{\mathcal{K}\ni k\to\infty} \|\bar{\boldsymbol{x}}_k - \boldsymbol{x}^\star\|^2 + \|c(I-\tilde{W})^{-1}[\boldsymbol{v}_k - \boldsymbol{v}^\star]\|^2 = 0,$$

but since $\{\|\bar{\boldsymbol{x}}_k - \boldsymbol{x}^\star\|^2 + \|c(I-\tilde{W})^{-1}[\boldsymbol{v}_k - \boldsymbol{v}^\star]\|^2\}_{k\geq 0}$ is convergent, all its limit points are the same, meaning that

$$\lim_{k\to\infty} \|\bar{\boldsymbol{x}}_k - \boldsymbol{x}^\star\|^2 + \|c(I-\tilde{W})^{-1}[\boldsymbol{v}_k - \boldsymbol{v}^\star]\|^2 = 0.$$

Since the two terms are both non-negative, $(I-\tilde{W})^{-1}$ is non-singular, and $c > 0$, we have that

$$\lim_{k\to\infty} \boldsymbol{x}_k \overset{(27c)}{=} \lim_{k\to\infty} \bar{\boldsymbol{x}}_k = \boldsymbol{x}^\star \quad \text{and} \quad \lim_{k\to\infty} \boldsymbol{v}_k = \boldsymbol{v}^\star,$$

which concludes the proof. □

4 Numerical example

To showcase the performance of the proposed algorithm we test it on a random linear program with $N = 50$ agents and the following structure

$$\begin{aligned} \min_{x} \quad & \sum_{i=1}^{N} p_i^\top x \\ \text{subject to:} \quad & x \in \bigcap_{i=1}^{N} \{x : A_i x \leq b_i\}, \end{aligned} \tag{31}$$

where $x \in \mathbb{R}^{10}$ and, for all $i = 1,\dots,N$, each component of $p_i \in \mathbb{R}^{10}$ and $A_i \in \mathbb{R}^{50\times 10}$ is independently extracted at random from a Gaussian distribution with zero mean and unitary variance, and each component of $b_i \in \mathbb{R}^{50}$ is independently extracted at random from a uniform distribution over the interval $[0, 10]$. Since all quantities in (31) are generated at random, it might happen that the resulting linear program does not satisfy Assumption 2. If this is the case (we can check it using a centralized solver), we simply discard that instance of the problem and generate another one.

Clearly, (31) fits the structure of $\mathcal{P}$ and we can therefore apply Proximal-Tracking to compute an optimal solution in a distributed way. To this end, we also generate at random a graph $\mathcal{G}$ satisfying Assumption 3, a matrix $\mathcal{W}'$, compliant with $\mathcal{G}$, satisfying Assumptions 4, and then set $\mathcal{W} = \frac{1}{2}(I + \mathcal{W}')$ to satisfy Assumption 5. For comparison purposes we also run the P-EXTRA algorithm from [29] (which is also equivalent to the algorithm in [15] when considering the non-differentiable term only) using the same matrices $\mathcal{W}'$ and $\mathcal{W}$.

To assess the impact of the tuning parameter c, we evaluated the performance of Proximal-Tracking and P-EXTRA for the following values of the penalty parameter $c \in \{10^{-1}, 10^{-1.5}, 10^{-2}, 10^{-2.5}, 10^{-3}, 10^{-3.5}\}$. For each c, we run both Algorithm 1 and P-EXTRA for 10^4 iterations, and we report in Figure 1 (left and right, respectively), on a logarithmic scale, the value of the relative optimality gap (upper plots)

$$\frac{|p^\top \bar{x}_k - p^\top x^\star|}{|p^\top x^\star|}$$

with respect to an optimal solution $x^\star$ computed with a centralized solver and the relative constraint violation (lower plots)

$$\frac{\|\max\{A\bar{x}_k - b, 0\}\|_\infty}{\|b\|}$$

related to the network average solution $\bar{x}_k$, where $p = \sum_{i=1}^N p_i$, $A = [A_1^\top \cdots A_N^\top]^\top$, and $b = [b_1^\top \cdots b_N^\top]^\top$. We report the evolution of $\bar{x}_k$ only, as agents local estimates $x_{i,k}$ behave similarly to $\bar{x}_k$. As can be observed from the left figure, the proposed algorithm converges to an optimal solution of $\mathcal{P}$ in all cases. It is also interesting to note how the performance of the algorithm is almost the same for a fairly wide range of values for the penalty parameter c, with noticeable slowdowns for the extreme cases $c = 10$ and $c = 10^{-4}$ only. Furthermore, it is interesting to mention that despite the value of c affects the transient behavior, for the considered problem the convergence is eventually exponential in all cases, with a rate (cf. the slopes of the lines in Figure 1) not affected by c. Similar comments applies also to the P-EXTRA algorithm. However, from Figure 1, it is easy to see how the proposed Proximal-Tracking is better than P-EXTRA in terms of convergence rate, as testified by the steeper slopes in the left plots with respect to those in the right plots, for all values of c.

5 Conclusion

In this paper we propose a novel distributed optimization algorithm for decision-coupled problems, which allows the agents to have different local constraints sets while requiring only convexity of their local objective functions and local constraints sets. The algorithm is proven to converge to an optimal solution for all values of a constant penalty parameter, and numerical simulations show that a linear convergence rate is achieved even without the strong convexity assumption. Our future research efforts will be devoted to formally estimate the convergence rate of the proposed algorithm, to re-

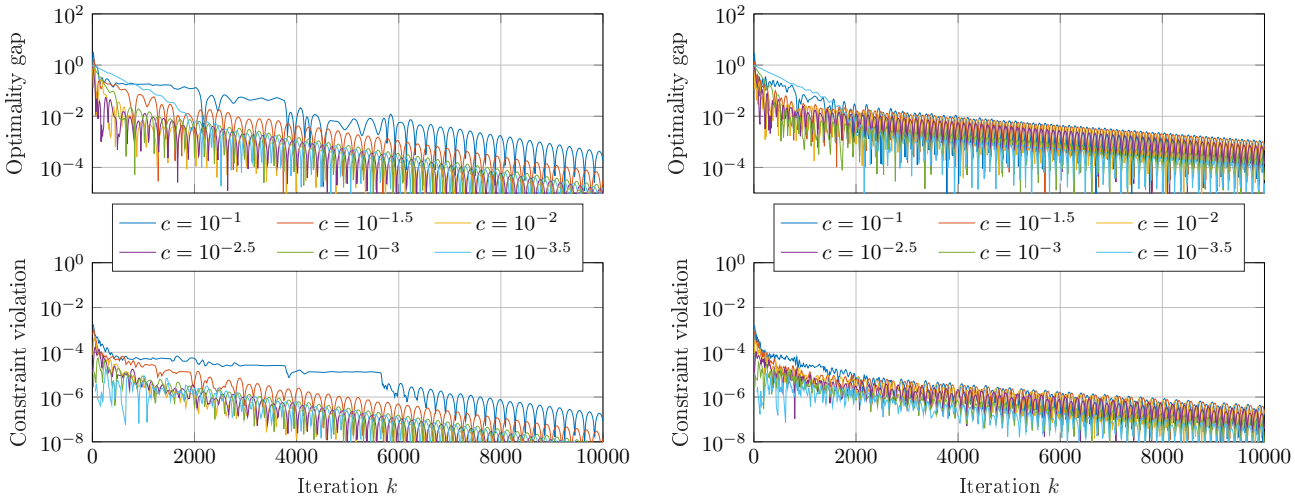

Fig. 1. Relative optimality gap (upper plots) and relative constraint violation (lower plots) of $\bar{x}_k$ across iterations of Proximal-Tracking (left plots) and P-EXTRA (right plots), for different values of the penalty parameter c.

lax the doubly stochasticity assumption with the aid of the push-sum protocol, and to modify the algorithm in order to cope with dynamic communication topologies. Another interesting research direction would be to study the effect of an inexact minimization step, using as a starting point the work in [27].

References

[1] Mohammed Aledhari, Rehma Razzak, Reza M. Parizi, and Fahad Saeed. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access*, 8:140699–140725, 2020.

[2] Sulaiman A. Alghunaim, Ernest Ryu, Kun Yuan, and Ali H. Sayed. Decentralized proximal gradient algorithms with linear convergence rates. *IEEE Transactions on Automatic Control*, 2020. In press.

[3] Dimitri P. Bertsekas and Athena Scientific. *Convex optimization algorithms*. Athena Scientific Belmont, 2015.

[4] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989.

[5] Nicoletta Bof, Ruggero Carli, Giuseppe Notarstefano, Luca Schenato, and Damiano Varagnolo. Multiagent newton–raphson optimization over lossy networks. *IEEE Transactions on Automatic Control*, 64(7):2983–2990, 2018.

[6] Paolo Di Lorenzo and Gesualdo Scutari. NEXT: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, 2016.

[7] Alejandro D. Domínguez-García and Christoforos N. Hadjicostis. Distributed strategies for average consensus in directed graphs. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 2124–2129. IEEE, 2011.

[8] Alessandro Falsone, Ivano Notarnicola, Giuseppe Notarstefano, and Maria Prandini. Tracking-ADMM for distributed constraint-coupled optimization. *Automatica*, 117:108962, 2020.

[9] Bahman Gharesifard and Jorge Cortés. Distributed strategies for generating weight-balanced and doubly stochastic digraphs. *European Journal of Control*, 18(6):539–557, 2012.

[10] Franck Iutzeler, Pascal Bianchi, Philippe Ciblat, and Walid Hachem. Explicit convergence rate of a distributed alternating direction method of multipliers. *IEEE Transactions on Automatic Control*, 61(4):892–904, 2015.

[11] Dušan Jakovetić, José M. F. Moura, and João M. F. Xavier. Linear convergence rate of a class of distributed augmented Lagrangian algorithms. *IEEE Transactions on Automatic Control*, 60(4):922–936, 2015.

[12] Bjorn Johansson, Tamás Keviczky, Mikael Johansson, and Karl Henrik Johansson. Subgradient methods and consensus algorithms for solving convex optimization problems. In *IEEE Conference on Decision and Control (CDC)*, pages 4185–4190, 2008.

[13] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez. Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems Magazine*, 39(3):40–72, 2019.

[14] Soomin Lee and Angelia Nedic. Distributed random projection algorithm for convex optimization. *IEEE Journal of Selected Topics in Signal Processing*, 7(2):221–229, 2013.

[15] Zhi Li, Wei Shi, and Ming Yan. A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *IEEE Transactions on Signal Processing*, 67(17):4494–4506, 2019.

[16] Qing Ling and Alejandro Ribeiro. Decentralized linearized alternating direction method of multipliers. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5447–5451. IEEE, 2014.

[17] Ali Makhdoumi and Asuman Ozdaglar. Convergence rate of distributed admm over networks. *IEEE Transactions on Automatic Control*, 62(10):5082–5095, 2017.

[18] Kostas Margellos, Alessandro Falsone, Simone Garatti, and Maria Prandini. Distributed constrained optimization and consensus in uncertain networks via proximal minimization. *IEEE Transactions on Automatic Control*, 63(5):1372–1387, 2018.

[19] João F. C. Mota, João M. F. Xavier, Pedro M. Q. Aguiar, and Markus Püschel. D-ADMM: A communication-efficient

distributed algorithm for separable optimization. *IEEE Transactions on Signal Processing*, 61(10):2718–2723, 2013.

[20] Angelia Nedic. Distributed gradient methods for convex machine learning problems in networks: Distributed optimization. *IEEE Signal Processing Magazine*, 37(3):92–101, 2020.

[21] Angelia Nedić and Alex Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2015.

[22] Angelia Nedić, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM J. on Optimization*, 27(4):2597–2633, 2017.

[23] Angelia Nedić and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[24] Angelia Nedić, Asuman Ozdaglar, and Pablo A. Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4), 2010.

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

[26] Ralph Tyrell Rockafellar. *Convex analysis*. Princeton University Press, 1970.

[27] Saverio Salzo and Silvia Villa. Inexact and accelerated proximal point algorithms. *Journal of Convex Analysis*, 19(4):1167–1192, 2012.

[28] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

[29] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. A proximal gradient algorithm for decentralized composite optimization. *IEEE Transactions on Signal Processing*, 63(22):6013–6023, 2015.

[30] Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 62(7):1750–1761, 2014.

[31] Damiano Varagnolo, Filippo Zanella, Angelo Cenedese, Gianluigi Pillonetto, and Luca Schenato. Newton-Raphson consensus for distributed convex optimization. *IEEE Transactions on Automatic Control*, 61(4):994–1009, 2016.

[32] Tianyu Wu, Kun Yuan, Qing Ling, Wotao Yin, and Ali H. Sayed. Decentralized consensus optimization with asynchrony and delays. *IEEE Transactions on Signal and Information Processing over Networks*, 4(2):293–307, 2018.

[33] Chenguang Xi, Ran Xin, and Usman A. S. Khan. ADD-OPT: Accelerated distributed directed optimization. *IEEE Transactions on Automatic Control*, 63(5):1329–1339, 2018.

[34] Ran Xin, Shi Pu, Angelia Nedić, and Usman A. Khan. A general framework for decentralized optimization with first-order methods. *Proceedings of the IEEE*, 108(11):1869–1889, 2020.

[35] Jinming Xu, Ye Tian, Ying Sun, and Gesualdo Scutari. A unified algorithmic framework for distributed composite optimization. In *59th IEEE Conference on Decision and Control (CDC)*, pages 2309–2316, 2020.

[36] Jinming Xu, Shanying Zhu, Yeng Chai Soh, and Lihua Xie. Convergence of asynchronous distributed gradient methods over stochastic networks. *IEEE Transactions on Automatic Control*, 63(2):434–448, 2018.

[37] Minghui Zhu and Sonia Martínez. Discrete-time dynamic average consensus. *Automatica*, 46(2):322–329, 2010.

[38] Minghui Zhu and Sonia Martínez. On distributed convex optimization under inequality and equality constraints. *IEEE Transactions on Automatic Control*, 57(1):151–164, 2012.

Appendix

Proof of Lemma 1

Recall that $W = \mathcal{W} \otimes I_n$, $W_\infty = (\frac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top) \otimes I_n$, and $\tilde{W} = W - W_\infty$.

Proof of (9a)

Consider a vector $\boldsymbol{z} = [z_1^\top \ \cdots \ z_N^\top]^\top$, with $z_i \in \mathbb{R}^n$ for all $i = 1, \ldots, N$, define $\bar{z} = \frac{1}{N}\sum_{i=1}^N z_i$, and $\bar{\boldsymbol{z}} = \mathbb{1}_N \otimes \bar{z}$. Then,

$$W_\infty \boldsymbol{z} = ((\tfrac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top) \otimes I_n)\boldsymbol{z} = \tfrac{1}{N}\begin{bmatrix} I_n & \cdots & I_n \\ \vdots & \ddots & \vdots \\ I_n & \cdots & I_n \end{bmatrix}\begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix}$$
$$= \begin{bmatrix} \frac{1}{N}\sum_{i=1}^N z_i \\ \vdots \\ \frac{1}{N}\sum_{i=1}^N z_i \end{bmatrix} = \begin{bmatrix} \bar{z} \\ \vdots \\ \bar{z} \end{bmatrix} = \mathbb{1}_N \otimes \bar{z} = \bar{\boldsymbol{z}}$$

Proof of (9b)

$$\begin{aligned} W_\infty W &= ((\tfrac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top) \otimes I_n)(\mathcal{W} \otimes I_n) \\ &\overset{(a)}{=} ((\tfrac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top\mathcal{W}) \otimes I_n I_n) \\ &\overset{(b)}{=} ((\tfrac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top) \otimes I_n) = W_\infty \end{aligned}$$

and

$$\begin{aligned} W W_\infty &= (\mathcal{W} \otimes I_n)((\tfrac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top) \otimes I_n) \\ &\overset{(a)}{=} ((\tfrac{1}{N}\mathcal{W}\mathbb{1}_N\mathbb{1}_N^\top) \otimes I_n I_n) \\ &\overset{(b)}{=} ((\tfrac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top) \otimes I_n) = W_\infty \end{aligned}$$

where in both derivations (a) is due to the mixed-product property of the Kronecker product and (b) is due to $\mathbb{1}_N^\top\mathcal{W} = \mathbb{1}_N^\top$ and $\mathcal{W}\mathbb{1}_N = \mathbb{1}_N$ by the doubly stochasticity of $\mathcal{W}$ under Assumption 4.

Proof of (9c)

Let $y \in \mathbb{R}^n$, then

$$\begin{aligned} W(\mathbb{1}_N \otimes y) &= (\mathcal{W} \otimes I_n)(\mathbb{1}_N \otimes y) \\ &\overset{(a)}{=} (\mathcal{W}\mathbb{1}_N \otimes I_n y) \\ &\overset{(b)}{=} (\mathbb{1}_N \otimes y), \end{aligned}$$

where (a) is due to the mixed-product property of the Kronecker product and (b) is due to $\mathcal{W}\mathbb{1}_N = \mathbb{1}_N$ under the doubly stochasticity requirement of Assumption 4.

Proof of (9d)

Similarly to the proof of (9c), let $y \in \mathbb{R}^n$, then

$$\begin{aligned} W_\infty(\mathbb{1}_N \otimes y) &= ((\tfrac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top) \otimes I_n)(\mathbb{1}_N \otimes y) \\ &\overset{(a)}{=} ((\tfrac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top\mathbb{1}_N) \otimes I_n y) \\ &\overset{(b)}{=} (\mathbb{1}_N \otimes y), \end{aligned}$$

where (a) is due to the mixed-product property of the Kronecker product and (b) is due to $\mathbb{1}_N^\top\mathbb{1}_N = N$.

Proof of (9e)

It suffices to see that

$$W_\infty \boldsymbol{z} \overset{(9a)}{=} \bar{\boldsymbol{z}} = (\mathbb{1}_N \otimes \bar{z}) \overset{(9d)}{=} W_\infty(\mathbb{1}_N \otimes \bar{z}) = W_\infty \bar{\boldsymbol{z}}$$

to conclude that $W_\infty(\boldsymbol{z} - \bar{\boldsymbol{z}}) = 0$.

Proof of (9f)

We have that

$$\begin{aligned} &\rho(\tilde{W}) \leq \|\tilde{W}\| = \|(\mathcal{W} \otimes I_n) - (\tfrac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top \otimes I_n)\| \\ &\overset{(a)}{=} \|(\mathcal{W} - \tfrac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top) \otimes I_n)\| \overset{(b)}{=} \|\mathcal{W} - \tfrac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top\|\|I_n\| \overset{(c)}{<} 1, \end{aligned}$$

where $\|\tilde{W}\|$ is the spectral norm of $\tilde{W}$, (a) and (b) are due to the linearity and spectral property respectively of the Kronecker product and (c) follows from $\|\mathcal{W} - \frac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top\| < 1$ as a consequence of the Perron-Frobenius theorem, under Assumptions 3 and 4.

Proof of (9g)

First let us note that $(I - \tilde{W})$ is invertible as a consequence of all eigenvalues of $\tilde{W}$ lying in the open unit circle due to $\rho(\tilde{W}) < 1$ from (9f). Moreover,

$$\begin{aligned} W_\infty W_\infty &= (\tfrac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top \otimes I_n)(\tfrac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top \otimes I_n) \\ &\overset{(a)}{=} (\tfrac{1}{N^2}\mathbb{1}_N\mathbb{1}_N^\top\mathbb{1}_N\mathbb{1}_N^\top \otimes I_n) \\ &\overset{(b)}{=} (\tfrac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top \otimes I_n) = W_\infty, \end{aligned}$$

where (a) is due to the mixed-product property of the Kronecker product and (b) is due to $\mathbb{1}_N^\top\mathbb{1}_N = N$. The previous relation can be used to show that

$$\begin{aligned} \tilde{W}W_\infty = WW_\infty - W_\infty W_\infty = WW_\infty - W_\infty \\ \overset{(9b)}{=} W_\infty - W_\infty = 0. \quad \text{(A.1)} \end{aligned}$$

Then, it suffices to compute

$$\begin{aligned} (I - \tilde{W})(I - W_\infty) &= I - \tilde{W} - W_\infty + \tilde{W}W_\infty \\ &= I - W \end{aligned}$$

where the second equality is due to $\tilde{W}W_\infty = 0$ by (A.1) and $W = \tilde{W} + W_\infty$. The desired result is obtained left-multiplying the previous relation by $(I - \tilde{W})^{-1}$, which exists since $\rho(\tilde{W}) < 1$. $\square$