Automatica 157 (2023) 111269

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Augmented Lagrangian Tracking for distributed optimization with equality and inequality coupling constraints ☆

Alessandro Falsone *, Maria Prandini

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

A R T I C L E   I N F O

Article history:
Received 9 September 2022
Received in revised form 20 June 2023
Accepted 24 July 2023
Available online 5 September 2023

Keywords:
Distributed optimization
Constraint-coupled optimization
Proximal algorithm
Augmented Lagrangian
ADMM

A B S T R A C T

In this paper we propose a novel Augmented Lagrangian Tracking distributed optimization algorithm for solving multi-agent optimization problems where each agent has its own decision variables, cost function and constraint set, and the goal is to minimize the sum of the agents' cost functions subject to local constraints plus some additional coupling constraint involving the decision variables of all the agents. In contrast to alternative approaches available in the literature, the proposed algorithm jointly features a constant penalty parameter, the ability to cope with unbounded local constraint sets, and the ability to handle both affine equality and nonlinear inequality coupling constraints, while requiring convexity only. The effectiveness of the approach is shown first on an artificial example with complexity features that make other state-of-the-art algorithms not applicable and then on a realistic example involving the optimization of the charging schedule of a fleet of electric vehicles.

1. Introduction

In the last decade, distributed optimization has proven to be a successful technique to optimize the behavior of large-scale multi-agent systems. Key to its success is the ability to split an optimization problem involving the whole system into smaller ones to be solved locally by the agents while exchanging information on their tentative solutions with their neighbors, so as to jointly converge to an optimal solution.

In this paper we consider those optimization problems in which each agent has its own decision variables, cost function and constraint set, and the goal is to minimize the sum of the agents' cost functions subject to some coupling constraint involving the decision variables of all the agents (*Constraint-Coupled Problem* – CCP). The presence of the coupling element makes the problem solution challenging, especially in a distributed framework where no central authority is available to coordinate the agents and each agent has no knowledge of the local information of the others. CCPs typically model situations in which each agent has its own control action subject to its own actuation constraint, but, in order to perform these actions, consumes some resources, which are in a finite amount and are shared by all agents. Even though CCPs arise naturally in practical applications, most of the early literature on distributed optimization focuses on optimization problems where the agents are coupled because they have to agree on a common decision vector (*Decision-Coupled Problems* – DCPs) and only recently the interest shifted from DCPs to CCPs.

The earliest distributed solutions to DPCs are algorithms based on a combination of standard (sub)gradient methods for the optimization of the agents local cost functions subject to the local constraints and consensus schemes to drive the agents towards a common optimal decision, see, e.g., Nedić and Ozdaglar (2009), Nedić, Ozdaglar, and Parrilo (2010). Other works are based on consensus schemes mixed with primal–dual methods, see, e.g., Zhu and Martínez (2012), or mixed with proximal minimization, see Margellos, Falsone, Garatti, and Prandini (2018), for the optimization of the agents local cost functions.

Note that distributed algorithms developed for DCPs can, in principle, be applied to CCPs by interpreting the collection of the decision variables of all the agents as the common decision vector to agree upon. However, this entails that each agent would have to store, update, and communicate the tentative solutions of all other agents and have access to the whole coupling constraint rather than the portion affected by its decision variables only, thus increasing communication and computational burden for the entire network and ultimately hampering the applicability of such strategies.

Approaches directly aiming at CCPs typically leverage Lagrangian duality to deal with the coupling constraint. Indeed, if the coupling constraint can be expressed as a sum of agents contributions, then the Lagrangian has a separable structure in the primal decision variables and the dual problem has the same

☆ The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Luca Schenato under the direction of Editor Christos G. Cassandras.

* Corresponding author.
E-mail addresses: alessandro.falsone@polimi.it (A. Falsone), maria.prandini@polimi.it (M. Prandini).

https://doi.org/10.1016/j.automatica.2023.111269

structure of a DCP with a common vector of Lagrange multipliers as decision vector. Works based on primal–dual algorithms, like, e.g., Chang, Nedić, and Scaglione (2014), seek to find the saddle point of said Lagrangian. Methods leveraging dual decomposition exploit the fact that the dual of a CCP is a DCP to build upon the techniques developed for DCPs, for example combining consensus schemes together with either dual subgradient methods, see, e.g., Falsone, Margellos, Garatti, and Prandini (2017) and Notarnicola and Notarstefano (2020), or dual proximal minimization approaches, see Falsone and Prandini (2020). However, they typically require an additional procedure to recover the optimal primal solution of the CCP.

All the approaches mentioned so far require a vanishing step-size for the gradient update or an increasing penalty for the proximal operator, which ultimately leads to a slow convergence rate of the corresponding algorithm. The effort of the community thus shifted to designing distributed algorithms with a faster convergence rate, primarily by employing a fixed step-size, but not only, see, e.g., Romao, Margellos, Notarstefano, and Papachristodoulou (2021).

A first attempt in this direction for DCPs is presented in Mota, Xavier, Aguiar, and Püschel (2013), where a consensus scheme is used together with the Alternating Direction Method Multipliers (ADMM, see Bertsekas & Tsitsiklis, 1989) with a constant penalty parameter, in place of the (sub)gradient method. Linear convergence rate for distributed algorithms solving DCPs has been recently achieved by the so-called gradient-tracking schemes (see, e.g., Nedić, Olshevsky, & Shi, 2017; Qu & Li, 2018), where the gradient method for handling the local optimization is used together with the so-called dynamic average consensus (firstly proposed in Zhu & Martínez, 2010 and further elaborated in Kia et al., 2019) in place of the original consensus scheme. In these methods, the faster convergence is achieved thanks to a more reactive consensus mechanism together with a constant step-size, but at the expense of strict requirements on the agents local cost functions like strong convexity and smoothness/Lipschitz continuity of the gradient and the inability to cope with local constraints. In the recent work Falsone and Prandini (2022), the dynamic average consensus scheme is used within a proximal minimization framework to achieve convergence with a constant penalty parameter and with local constraints under just convexity assumptions on the cost and constraints. Even though a rate is not provided in Falsone and Prandini (2022), numerical experiments show that convergence is faster then algorithms with a time varying penalty.

Similarly to DCPs, latest distributed algorithms for solving CCPs are aiming at improving the convergence rate via a constant step-size. A primal–dual algorithm with constant step-size is proposed in Liang, Wang, and Yin (2020), but requires smoothness of the local cost functions. In the recent paper Su, Wang, and Sun (2021) another primal–dual algorithm has been proposed, which does not require any smoothness assumption, but requires compactness of the local constraints sets and deals with equality coupling constraints only. An approach based on ADMM and consensus is derived in Chang (2016), but deals with affine coupling constraints only, and a strategy combining ADMM and dynamic average consensus with faster convergence is presented in Falsone, Notarnicola, Notarstefano, and Prandini (2020), dealing with equality constraints only. A proximal-minimization based algorithm has been also very recently proposed in Wang and Hu (2022), but it deals with equality constraints only and such coupling constraints must also comply with the communication topology of the agents.

The contribution of this paper is to propose a novel Augmented Lagrangian Tracking distributed algorithm for solving CCPs in a distributed way. Similarly to other approaches designed for CCPs (see, e.g., Falsone & Prandini, 2020), we leverage duality theory and build upon a recently proposed algorithm for DCPs to develop the proposed algorithm for CCPs. However, in contrast to all approaches for CCPs in the literature, the devised algorithm *jointly* exhibits the following features:

- a constant penalty parameter is adopted, which is only required to be positive;
- coupling constraints can be a mix of affine equalities and nonlinear inequalities;
- cost functions and coupling constraint nonlinear functions are *not* required to be Lipschitz continuous, smooth, or strongly convex, but only convex;
- local constraints set are not required to be bounded.

The rest of the paper is organized as follows. In Section 2 we present the set-up of constraint-coupled optimization problems. In Section 3 we introduce the distributed computation framework, gradually derive the proposed distributed algorithm, state its convergence properties, and also highlight some interesting connections between the proposed algorithm and other ones in the literature. In Section 4 we apply our algorithm to an artificial example with complexity features that make state-of-the-art distributed algorithms not applicable, and then we extensively analyze its performance on a realistic example regarding the optimal charging schedule for a fleet of electric vehicles. In Section 5 we draw some conclusions. Finally, proofs of the main results are reported in the Appendix to streamline the presentation.

Notation. We denote with $\mathbb{N}$ the set of non-negative integers, with $\mathbb{R}$ the set of real numbers, and with $\mathbb{R}_+$ the set of non-negative reals. For an extended real-valued function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, $\mathrm{dom}(f) = \{x : f(x) < +\infty\}$ is the domain of f, we denote by $\partial f(x) \subset \mathbb{R}^n$ 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 $\nabla f(x) \in \mathbb{R}^n$ is 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 $\mathrm{relint}(\cdot)$ denotes the relative interior of its argument. The vector in $\mathbb{R}^n$ containing all ones is denoted by $\mathbb{1}_n$ (for brevity, subscript will be omitted when clear from the context). For a matrix S we write $S^\top$ to denote its transpose. For a vector v, $\|v\|$ is the Euclidean norm of v and $[v]_s$ is its sth component. Given a collection of vectors $\{v_1, \dots, v_m\}$, we use $(v_1, \dots, v_m)$ as a shorthand for $[v_1^\top \cdots v_m^\top]^\top$, and we denote by $\min\{v_1, \dots, v_m\}$ or $\max\{v_1, \dots, v_m\}$ their component-wise minimum and maximum, respectively.

2. Constraint-coupled optimization

We consider a multi-agent system composed of N agents that are willing to cooperate to solve a decision making problem involving the whole system. Specifically, all agents shall set their local decision variables $x_i \in \mathbb{R}^{n_i}$, $i = 1, \dots, N$, so as to find an optimal solution to the following constrained optimization program

$$\inf_{x_1,\dots,x_N} \quad \sum_{i=1}^N f_i(x_i) \tag{$\mathcal{P}$}$$

$$\text{subject to:} \quad \sum_{i=1}^N A_i x_i = b, \quad \sum_{i=1}^N h_i(x_i) \le 0$$

$$x_i \in X_i \qquad i = 1, \dots, N,$$

where $A_i \in \mathbb{R}^{p \times n_i}$ and $b \in \mathbb{R}^p$, with $p \in \mathbb{N}$, characterize the equality coupling constraint, $h_i : \mathbb{R}^{n_i} \to \mathbb{R}^q \cup \{+\infty\}$, with $q \in \mathbb{N}$, characterize the inequality coupling constraints, $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$ denotes the local cost function, and $X_i \subset \mathbb{R}^{n_i}$ is the local constraint set of agent i, $i = 1, \dots, N$. We impose the following assumption:

Assumption 1 (*Convexity*)**.** For all $i = 1, \dots, N$, the functions f_i and h_i and the set X_i are convex and closed, and $\cap_{s=1}^{q} \operatorname{relint}(\operatorname{dom}([h_i]_s)) \neq \emptyset$. □

Note that, differently from other approaches in the literature, we do not require the local constraint sets X_i to be compact, functions f_i and g_i to be smooth, differentiable, or strictly/strongly convex. Moreover, if, for some s, $[h_i]_s$ is a polyhedral function, then $\operatorname{relint}(\operatorname{dom}([h_i]_s))$ can be substituted with $\operatorname{dom}([h_i]_s)$.

To deal with problems in the form of $\mathcal{P}$, a common practice is resorting to duality theory to handle the coupling constraint. Let $\boldsymbol{x} = (x_1, \dots, x_N)$, consider a vector $\lambda \in \mathbb{R}^p$ and a vector $\mu \in \mathbb{R}^q_+$ of Lagrange multipliers and let

$$L(\boldsymbol{x}, \lambda, \mu) = \sum_{i=1}^{N} f_i(x_i) + \lambda^\top \sum_{i=1}^{N}(A_i x_i - b) + \mu^\top \sum_{i=1}^{N} h_i(x_i)$$
$$= \sum_{i=1}^{N} f_i(x_i) + \lambda^\top (A_i x_i - b_i) + \mu^\top h_i(x_i), \tag{1}$$

with $b_1, \dots, b_N$ such that $\sum_{i=1}^{N} b_i = b$, be the Lagrangian function obtained by dualizing the coupling constraints $\sum_{i=1}^{N} A_i x_i = b$ and $\sum_{i=1}^{N} h_i(x_i) \leq 0$. The dual of $\mathcal{P}$ is then given by

$$\max_{\substack{\lambda \in \mathbb{R}^p \\ \mu \in \mathbb{R}^q_+}} \inf_{\boldsymbol{x} \in X} L(\boldsymbol{x}, \lambda, \mu) = \max_{\substack{\lambda \in \mathbb{R}^p \\ \mu \in \mathbb{R}^q_+}} \sum_{i=1}^{N} \varphi_i(\lambda, \mu), \tag{$\mathcal{D}$}$$

where $X = X_1 \times \cdots \times X_N$, and the ith contribution φ_i of the dual objective function is defined as

$$\varphi_i(\lambda, \mu) = \inf_{x_i \in X_i} f_i(x_i) + \lambda^\top (A_i x_i - b_i) + \mu^\top h_i(x_i). \tag{2}$$

Note that $\varphi_i(\lambda, \mu)$ may assume the value $-\infty$ for some $(\lambda, \mu) \in \mathbb{R}^p \times \mathbb{R}^q_+$. The next assumption ensures that $\mathcal{P}$ and $\mathcal{D}$ are well-posed and that the domains of the local dual functions overlap sufficiently and not just on the boundary.

Assumption 2 (*Well-posedness*)**.** The optimal value $f^\star$ of $\mathcal{P}$ is finite, $\mathcal{D}$ admits an optimal solution $(\lambda^\star, \mu^\star)$, and strong duality holds. Moreover, $(\mathbb{R}^p \times \mathbb{R}^q_+) \cap (\cap_{i=1}^{N} \operatorname{relint}(\operatorname{dom}(-\varphi_i))) \neq \emptyset$. □

Note that compactness of X_i is a sufficient (but not necessary!) condition for the second part of Assumption 2 to be verified. Moreover, similarly to the discussion after Assumption 1, if, for some i, φ_i is polyhedral, then $\operatorname{relint}(\operatorname{dom}(-\varphi_i))$ can be replaced with $\operatorname{dom}(-\varphi_i)$ in the second part of Assumption 2. Therefore, in the particular case of a linear program, the second part of Assumption 2 translates to all φ_i's having at least a point in common, which is trivially true if an optimal dual solution $(\lambda^\star, \mu^\star)$ exists.

3. Augmented Lagrangian Tracking algorithm

In optimization, the application of an algorithm to solve the dual of some primal problem often gives rise to a novel algorithm to solve the primal problem itself. This is indeed the case for the augmented Lagrangian method presented in Bertsekas (2015, Chapter 5), which results from the application of a proximal minimization algorithm to the dual of a constrained optimization problem. Here, we are able to extend this strategy to a distributed framework involving multiple agents that are coupled by some constraints.

We start by introducing the distributed computational framework of interest in Section 3.1, we then exploit the Proximal-Tracking algorithm recently devised in Falsone and Prandini (2022) for DCPs and apply it to the dual $\mathcal{D}$ of $\mathcal{P}$ to then develop a novel algorithm for solving directly CCPs in the form of $\mathcal{P}$, in a distributed way.

Since a direct application of the Proximal-Tracking algorithm to solve $\mathcal{D}$ in a distributed way does not lead to a readily implementable procedure (see Section 3.2), our first contribution is to introduce an equivalent algorithm which can instead be implemented in practice (Section 3.3). Proving such an equivalence, automatically shows convergence of the proposed algorithm to the dual optimal solution. In contrast with the case when the Proximal-Tracking algorithm is directly applied to $\mathcal{D}$, the proposed algorithm generates tentative primal solutions. Our second contribution is then to show that, in the limit, the generated primal iterates are feasible and achieve the optimal cost.

Finally, in Section 3.4 we point out some interesting connections between the proposed distributed resolution scheme for $\mathcal{P}$, augmented Lagrangian methods and the Tracking-ADMM approach proposed in Falsone et al. (2020).

3.1. Distributed computation framework

In the considered distributed set-up, the cost function f_i, constraint set X_i, and contributions A_i and h_i of agent i to the coupling constraints have to be regarded as private information, not to be disclosed to other agents, whereas the value of b is known to every agent.

To cooperatively solve $\mathcal{P}$, the agents must repeatedly exchange information through the communication network. In this work, communications among the agents are modeled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the vertex set $\mathcal{V} = \{1, \dots, N\}$ represents the agents, and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the communication links between the agents. If $(i, j) \in \mathcal{E}$, then, at each iteration k, agent i receives information from agent j. We denote by $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ the set of *neighbors* of agent i. We impose the following assumption of $\mathcal{G}$, which is common in distributed optimization and ensures that information can flow from any agent to any other agent.

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. □

3.2. Proximal-Tracking

We start by noticing that $\mathcal{D}$ is a DCP and it could be solved in a distributed way by applying the Proximal-Tracking algorithm proposed in Falsone and Prandini (2022).

Accordingly, at each iteration, agent i should perform the following steps[1]

$$\begin{bmatrix} \ell_i^k \\ m_i^k \end{bmatrix} = \sum_{j \in \mathcal{N}_i} w_{ij} \begin{bmatrix} \lambda_j^k \\ \mu_j^k \end{bmatrix} \tag{3a}$$

$$\begin{bmatrix} \delta_i^k \\ \gamma_i^k \end{bmatrix} = \sum_{j \in \mathcal{N}_i} w_{ij} \begin{bmatrix} d_j^k \\ g_j^k \end{bmatrix} \tag{3b}$$

[1] To ease the comparison, we packed together with square brackets those sequences that maps into a single sequence of Falsone and Prandini (2022, Algorithm 1). This way (3a)–(3e) have a one-to-one correspondence with those steps that Falsone and Prandini (2022, Algorithm 1) performs at each iteration.

$$\begin{bmatrix}\lambda_i^{k+1}\\ \mu_i^{k+1}\end{bmatrix} = \arg\min_{\substack{\lambda_i\in\mathbb{R}^p\\ \mu_i\in\mathbb{R}^q_+}} -\varphi_i(\lambda_i,\mu_i) + \begin{bmatrix}\delta_i^k - u_i^k\\ \gamma_i^k - v_i^k\end{bmatrix}^\top \begin{bmatrix}\lambda_i\\ \mu_i\end{bmatrix} + \frac{1}{2c}\left\|\begin{bmatrix}\lambda_i-\ell_i^k\\ \mu_i - m_i^k\end{bmatrix}\right\|^2 \tag{3c}$$

$$\begin{bmatrix}u_i^{k+1}\\ v_i^{k+1}\end{bmatrix} = \frac{1}{c}\begin{bmatrix}\ell_i^k - \lambda_i^{k+1}\\ m_i^k - \mu_i^{k+1}\end{bmatrix} + \begin{bmatrix}u_i^k - \delta_i^k\\ v_i^k - \gamma_i^k\end{bmatrix} \tag{3d}$$

$$\begin{bmatrix}d_i^{k+1}\\ g_i^{k+1}\end{bmatrix} = \begin{bmatrix}\delta_i^k\\ \gamma_i^k\end{bmatrix} + \begin{bmatrix}u_i^{k+1}\\ v_i^{k+1}\end{bmatrix} - \begin{bmatrix}u_i^k\\ v_i^k\end{bmatrix} \tag{3e}$$

where (λ_i^k, μ_i^k) are local estimates of the optimal solution of $\mathcal{D}$, $c > 0$ is a penalty coefficient, and w_{ij} is a link weight, modeling how much agent i values the information received by agent j. For those $(i,j) \notin \mathcal{E}$, $w_{ij} = 0$, meaning that agent i does not receive any information from agent j.

Let $W \in \mathbb{R}^{N\times N}$ be the matrix whose (i,j)-th entry is w_{ij}, also known as the *consensus matrix*. In order for the Proximal-Tracking algorithm in (3) to work, we need to impose the following assumption on W.

Assumption 4 (*Consensus Weights*)**.** Matrix W is symmetric ($W = W^\top$), doubly stochastic ($W\mathbb{1} = W^\top\mathbb{1} = \mathbb{1}$), and positive semi-definite. □

Following the interpretation provided in Falsone and Prandini (2022), the sequence (u_i^k, v_i^k) represents a subgradient of the local objective function $-\varphi_i(\lambda_i,\mu_i)$ of agent i, while (d_i^k, g_i^k) acts as a (locally available to agent i) estimate of the global subgradient $\frac{1}{N}\sum_{i=1}^N (u_i^k, v_i^k)$.

In (3a), agent i constructs a weighted average (ℓ_i^k, m_i^k) of its own estimate (λ_i^k, μ_i^k) of the dual optimal solution and the estimates of its neighboring agents. Similarly, in (3b) agent i constructs a weighted average (δ_i^k, γ_i^k) of its own estimate (d_i^k, g_i^k) of the network average subgradient and the estimates of its neighboring agents. Then, in (3c), it updates its local estimate by solving a local minimization problem where the local cost function $-\varphi_i(\cdot)$ is augmented by a quadratic term that penalizes the distance of the new estimate from the average (ℓ_i^k, m_i^k) computed in (3a) and by a linear term, which steers the minimization away from the direction given by the local subgradient and towards the direction of the global subgradient, estimated by (d_i^k, g_i^k). It then computes the new subgradient in (3d) and it finally updates the global subgradient estimate (d_i^k, g_i^k) in (3e) using a dynamic consensus mechanism, see Kia et al. (2019).

According to Falsone and Prandini (2022), the iterative scheme in (3) must be initialized as $(\lambda_i^0, \mu_i^0) \in \mathbb{R}^p \times \mathbb{R}^q$, $(u_i^0, v_i^0) \in \mathbb{R}^p \times \mathbb{R}^q$ and $(d_i^0, g_i^0) = (u_i^0, v_i^0)$.

Unfortunately, step (3c) involves minimizing a function $-\varphi_i(\cdot)$, defined only implicitly via the minimization in (2), which (despite being convex) is not easy in practice. Moreover, no primal sequence $\{x_i^k\}_{k\geq 0}$ is generated by (3). In the next section, we introduce a different formulation of (3) that can be actually implemented by means of available solvers and show that it is indeed equivalent to (3). Notably, the proposed algorithm directly provides also primal iterates that are shown to be, in the limit, an optimal solution to the primal problem $\mathcal{P}$. Note that this is not always the case for dual algorithms, since they typically require additional recovery procedures/sequences to ensure primal optimality (see, e.g., Falsone et al., 2017; Falsone & Prandini, 2020).

3.3. Proposed algorithm

In the proposed Algorithm 1, in Steps 6–9 agent i constructs a weighted average of its own estimate and those from the neighbors of the dual tentative solutions (λ_i^k, μ_i^k) (cf. Steps 6–7) and the quantities (d_i^k, g_i^k) (cf. Steps 8–9), as in (3a) and (3b) of (3). It then performs a local minimization step involving its local cost function f_i, plus a linear and a quadratic terms related to the equality coupling constraints, and another quadratic term involving the inequality coupling constraints (cf. Step 10). This local minimization step provides an estimate of the optimal local solution x_i^{k+1} to the primal problem $\mathcal{P}$, which is then used for updating the quantities (d_i^k, g_i^k) (cf. Steps 12–13) according to a dynamic average consensus scheme (see Kia et al., 2019), meaning that (d_i^k, g_i^k) serves as a local estimate of the global quantity $-\frac{1}{N}\sum_{i=1}^N (A_i x_i^k - b_i, h_i(x_i^k) + \sigma_i^k)$, which is the opposite of the violation of the coupling constraint. The local estimate (d_i^k, g_i^k) of such a violation is then used in the local multiplier updates (cf. Step 14–15). The non-negative quantity σ_i^k acts as a slack variable and takes into account the fact that the h_i's contributes to an inequality constraint when updating the sequence g_i^k.

Algorithm 1 Augmented Lagrangian Tracking (ALT)

1: **Initialization**
2: $x_i^0 \in \mathbb{R}^{n_i}$, $\sigma_i^0 \in \mathbb{R}^q_+$
3: $\lambda_i^0 \in \mathbb{R}^p$, $d_i^0 = -(A_i x_i^0 - b_i)$
4: $\mu_i^0 \in \mathbb{R}^q$, $g_i^0 = -(h_i(x_i^0) + \sigma_i^0)$
5: **Repeat until convergence**
6: $\ell_i^k = \sum_{j\in\mathcal{N}_i} w_{ij}\,\lambda_j^k$
7: $m_i^k = \sum_{j\in\mathcal{N}_i} w_{ij}\,\mu_j^k$
8: $\delta_i^k = \sum_{j\in\mathcal{N}_i} w_{ij}\,d_j^k$
9: $\gamma_i^k = \sum_{j\in\mathcal{N}_i} w_{ij}\,g_j^k$
10: $x_i^{k+1} \in \arg\min_{x_i\in X_i} \Big\{ f_i(x_i) + {\ell_i^k}^\top A_i x_i + \frac{c}{2}\|A_i x_i - A_i x_i^k - \delta_i^k\|^2 + \frac{1}{2c}\|\max\{m_i^k + c\,(h_i(x_i) - h_i(x_i^k) - \sigma_i^k - \gamma_i^k), 0\}\|^2 \Big\}$
11: $\sigma_i^{k+1} = \max\{\gamma_i^k - h_i(x_i^{k+1}) + h_i(x_i^k) + \sigma_i^k - \frac{1}{c} m_i^k, 0\}$
12: $d_i^{k+1} = \delta_i^k - A_i x_i^{k+1} + A_i x_i^k$
13: $g_i^{k+1} = \gamma_i^k - (h_i(x_i^{k+1}) + \sigma_i^{k+1}) + (h_i(x_i^k) + \sigma_i^k)$
14: $\lambda_i^{k+1} = \ell_i^k - c\,d_i^{k+1}$
15: $\mu_i^{k+1} = m_i^k - c\,g_i^{k+1}$
16: $k \leftarrow k+1$

Clearly, in absence of equality constraints (i.e., $p = 0$), one can skip Steps 3, 6, 8, 12, and 14, and can remove all terms involving ℓ_i^k, A_i, and δ_i^k from the cost function of Step 10. Similarly, in absence of inequality constraints (i.e., $q = 0$) one can skip Steps 4, 7, 9, 11, 13, and 15, and can remove the $\frac{1}{2c}\|\max\{\cdot, 0\}\|$ term from the cost function of Step 10.

The following result formalizes the non-trivial relationship between Algorithm 1 and the iterative scheme in (3).

Theorem 1 (*Equivalence*)**.** *Under Assumptions 1 and 2, for any $c > 0$, Algorithm 1 and the iterative scheme in (3) are equivalent: they generate the same sequences $\{(\lambda_i^k, \mu_i^k)\}_{k\geq 0}$ and $\{(d_i^k, g_i^k)\}_{k\geq 0}$, and sequence $\{(u_i^k, v_i^k)\}_{k\geq 0}$ by (3) is equal to sequence $\{-(A_i x_i^k - b_i, h_i(x_i^k) + \sigma_i^k)\}_{k\geq 0}$ by Algorithm 1.* □

Owing to the equivalence granted by Theorem 1 and the convergence guarantees of the Proximal-Tracking algorithm provided in Falsone and Prandini (2022), we have the following corollary.

Corollary 1 (*Dual Optimality*). *Under Assumptions 1–4, for any $c > 0$, the sequence $\{(\lambda_i^k, \mu_i^k)\}_{k\geq 0}$ generated by Algorithm 1 converges to the same optimal solution $(\lambda^\star, \mu^\star)$ of $\mathcal{D}$ and, for each i, the sequence $\{-(A_i x_i^k - b_i, h_i(x_i^k) + \sigma_i^k)\}_{k\geq 0}$ converges to one element of $\partial(-\varphi_i + \mathcal{I}_{\mathbb{R}^p \times \mathbb{R}_+^q})(\lambda^\star, \mu^\star)$.* □

Unfortunately, Corollary 1 provides guarantees that the agents are converging to the same optimal solution of $\mathcal{D}$, but we are actually interested in solving $\mathcal{P}$. The following result guarantees that we are able to retrieve an optimal solution of $\mathcal{P}$.

Theorem 2 (*Primal Optimality*). *Under Assumptions 1–4, for any $c > 0$, the sequence $\{(x_1^k, \ldots, x_N^k)\}_{k\geq 0}$ generated by Algorithm 1 satisfies*

- $\lim_{k\to\infty} \sum_{i=1}^N f_i(x_i^k) = f^\star$,
- $\lim_{k\to\infty} \sum_{i=1}^N A_i x_i^k = b$,
- $\limsup_{k\to\infty} \sum_{i=1}^N h_i(x_i^k) \leq 0$,

meaning that $(x_1^k, \ldots, x_N^k)$ is, in the limit, feasible for $\mathcal{P}$ and achieves its optimal cost. □

While the proof of Corollary 1 is immediate as a consequence of Theorem 1, the results stated by Theorems 1 and 2 are non-trivial and their proofs are deferred to the Appendix.

Note that an optimal solution $x^\star = (x_1^\star, \ldots, x_N^\star)$ to $\mathcal{P}$ need not exist and, consequently, $(x_1^k, \ldots, x_N^k)$ need not converge. Nonetheless, Algorithm 1 still works, as $(x_1^k, \ldots, x_N^k)$ can only escape along feasible directions and its cost approaches the optimal one. To the best of our knowledge, this is the most general result among those available in the literature, which typically requires the existence of a primal optimal solution. Furthermore, with a mild additional assumption on $\mathcal{P}$, we can show that $\{(x_1^k, \ldots, x_N^k)\}_{k\geq 0}$ does not diverge and all its limit points are optimal primal solutions.

Corollary 2 (*Primal Limit Points*). *Consider Assumptions 1–4 and suppose that $c > 0$. If $\mathcal{P}$ admits a non-empty and bounded set of optimal solutions, then the sequence $\{(x_1^k, \ldots, x_N^k)\}_{k\geq 0}$ generated by Algorithm 1 is bounded and all its limit points are optimal primal solutions. Furthermore, if the optimal solution $x^\star$ is unique, then $\{(x_1^k, \ldots, x_N^k)\}_{k\geq 0}$ converges to $x^\star$.* □

Note that the additional assumption requiring the optimal solution set of $\mathcal{P}$ to be bounded is equivalent to assume that the overall primal objective function is coercive (i.e., grows unbounded) along those directions in which the feasible set of $\mathcal{P}$ is unbounded. Compactness of the X_i's sets is a sufficient but not necessary condition for this to hold.

3.4. Connections with existing algorithms

The first connection to highlight is the parallelism between the centralized algorithms and their distributed counterpart, which actually guided the developments in the previous subsection. In Falsone and Prandini (2022) it has been shown that Proximal-Tracking is the distributed counterpart of the proximal minimization algorithm. According to Bertsekas (2015), when a (centralized) proximal minimization algorithm is applied to a dual of a constrained optimization problem, the resulting dual proximal minimization algorithm gives rise to an augmented Lagrangian method, see Bertsekas (2015, Section 5.2.1). The proposed Augmented Lagrangian Tracking can thus be seen as the distributed counterpart of the augmented Lagrangian method, where each agent i minimizes an augmented Lagrangian in which the information regarding the contribution of the other agents on the coupling constraints has been replaced by local estimates via a dynamic average consensus (a.k.a. tracking) mechanism, hence the name of the proposed algorithm. The difference in sign of the dual updates with respect to the centralized augmented Lagrangian method is due to (d_i^k, g_i^k) being a local estimate of the opposite of the violation of the coupling constraint.

Another interesting connection is given by the relationship between Augmented Lagrangian Tracking and Tracking-ADMM in Falsone et al. (2020). Indeed, if we assume to have no inequality constraint ($q = 0$), then Augmented Lagrangian Tracking collapses (apart from the sign of d_i^k and δ_i^k) into Tracking-ADMM, which is thus a special case of the proposed approach. This connection forces us to observe that, while in the centralized case the augmented Lagrangian method and ADMM are different (despite similar) algorithms, their distributed counterparts studied in this paper and in Falsone et al. (2020) respectively, are the same. The development of this connection is certainly worth further investigations and constitutes an interesting direction of future research.

4. Numerical simulations

We first showcase the more general applicability of the proposed Augmented Lagrangian Tracking algorithm by considering an artificial example with various complexity features. We then consider a realistic example, which does not have all complexity features, but allows us to make a comparative analysis of Augmented Lagrangian Tracking versus a competing algorithm.

4.1. Artificial example

We consider a set of N agents, each one handling a non-negative decision variable $z_i \in \mathbb{R}_+$ and a cost function given by the maximum between two parabolas. The agents have to coordinate to minimize the sum of their cost functions subject to a budget constraint on the cumulative Euclidean norm of their decision variables and an equality constraint on their cumulative 1-norm. Formally, the problem can be posed as

$$\begin{aligned} \min_{z_1,\ldots,z_N} \quad & \sum_{i=1}^N \max\{(z_i - \nu_{i,1})^2, (z_i - \nu_{i,2})^2\} \qquad (4)\\ \text{subject to:} \quad & \|(z_1, \ldots, z_N)\|_1 = \|(s_1, \ldots, s_N)\|_1 \\ & \|(z_1, \ldots, z_N)\| \leq \|(r_1, \ldots, r_N)\| \\ & z_i \geq 0 \qquad i = 1, \ldots, N, \end{aligned}$$

which, owing to the fact that

$$\|(z_1, \ldots, z_N)\| \leq \|(r_1, \ldots, r_N)\| \iff \|(z_1, \ldots, z_N)\|^2 \leq \|(r_1, \ldots, r_N)\|^2,$$

and

$$z_i \geq 0 \quad \forall i \implies \|(z_1, \ldots, z_N)\|_1 = \sum_{i=1}^N z_i,$$

can be cast as an instance of $\mathcal{P}$ setting $x_i = z_i$, $f_i(x_i) = \max\{(x_i - \nu_{i,1})^2, (x_i - \nu_{i,2})^2\}$, $X_i = \mathbb{R}_+$, $A_i = 1$, $b_i = |s_i|$, $b = \sum_{i=1}^N b_i$, and $h_i(x_i) = x_i^2 - r_i^2$.

As it is clear by the previous equivalences, the resulting optimization problem has non-smooth non Lipschitz continuous local objective functions, nonlinear non Lipschitz continuous inequality coupling constraint function, unbounded local constraint sets, and has both equality and inequality coupling constraints. The reader should also note that if $\nu_{i,1}, \nu_{i,2} > 0$, then the unconstrained minimum is achieved at $x_i = \tilde{\nu}_i = \frac{1}{2}(\nu_{i,1}+\nu_{i,2}) > 0$ for all $i = 1, \ldots, N$, so that it is sufficient to set $s_i < \tilde{\nu}_i$ to ensure that the equality coupling constraints is not trivially satisfied. Then starting from the optimal solution, say $(x'_1, \ldots, x'_N)$, with the equality coupling constraint only, we can set $(r_1, \ldots, r_N) < (x'_1, \ldots, x'_N)$ to ensure

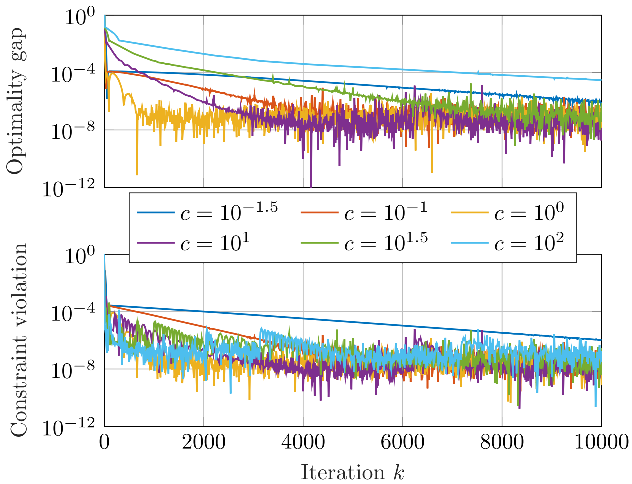

Fig. 1. Relative optimality gap (upper plot) and relative coupling constraint violation (lower plot) of $(x_1^k, \dots, x_N^k)$ across iterations of Augmented Lagrangian Tracking applied to (4), for different values of the penalty parameter c.

that also the inequality coupling constraint is active. Moreover, due to non-smoothness of the cost functions, if $N \gg 1$ and the coupling constrains perturb the solution only slightly, than the constrained minimizer $(x_1^\star, \dots, x_N^\star)$ may (depending on the values of $\nu_{i,1}$ and $\nu_{i,2}$) satisfy $x_i^\star = \tilde{\nu}_i$ for some $i \in \{1, \dots, N\}$, so that the cost function is non-differentiable precisely at the optimal solution.

In our tests we set $N = 10$, $\nu_{i,1}$ and $\nu_{i,2}$ are independently extracted at random from a uniform probability distribution over the intervals $[0.5, 1.5]$ and $[2.5, 3.5]$, respectively, for all $i = 1, \dots, N$, and $s_i = r_i = 0.95\,\tilde{\nu}_i$. Given the realizations of $\nu_{i,1}$ and $\nu_{i,2}$, it turns out that $x_i^\star = \tilde{\nu}_i$ for four agents.

In order to satisfy Assumption 3 we generate a communication network as follows. For each possible agent pair, the corresponding edge is included in the graph based on the outcome of the extraction from a Bernoulli probability distribution with success probability 0.15. If the resulting graph is not connected, then it is discarded and the procedure is restarted. To satisfy also Assumption 4, a tentative consensus matrix W is constructed using the procedure in Sinkhorn and Knopp (1967). If the resulting matrix is not positive semi-definite, then the graph and the consensus matrix are both discarded and the procedure is repeated.

We run Algorithm 1 for 10^4 iterations, for different values of the penalty parameter $c \in \{10^{-1.5}, 10^{-1}, 1, 10, 10^{1.5}, 10^2\}$ so as to cover a wide range of penalty parameters (three and a half orders of magnitude). In Fig. 1 we report, on a semi-logarithmic chart, the behavior across iterations of the relative optimality gap

$$\frac{|\sum_{i=1}^N f_i(x_i^k) - f^\star|}{|f^\star|},$$

where $f^\star$ is the optimal cost computed by a centralized solver, and the normalized maximum violation of the coupling constraints

$$\frac{\max\{|\sum_{i=1}^N A_i x_i^k - b_i|, \sum_{i=1}^N h_i(x_i^k)\}}{\max\{\|(s_1, \dots, s_N)\|_1, \|(r_1, \dots, r_N)\|^2\}}.$$

As can be observed from the picture, the proposed algorithm converges to an optimal solution of $\mathcal{P}$ for all values of c.

4.2. Realistic example

We now consider the plug-in electric vehicles optimal charging schedule problem described in Vujanic, Esfahani, Goulart, Mariéthoz, and Morari (2016).

The goal is to find a minimum-cost overnight charging strategy for a fleet of N electric vehicles. The charging profile of each vehicle must cope with upper and lower limits for the energy stored in the battery and must satisfy a desired target state of charge for the next morning. Moreover, all vehicles use the same point of connection to draw energy from the grid, which impose an additional constraint on the maximum amount of energy that the fleet can exchange with the grid. For simplicity we consider the "only charging" case, in which vehicles only draw energy without injecting any. Finally, differently from the problem considered in Vujanic et al. (2016), at each time-slot, we allow each vehicle to optimize the charging rate, instead of deciding whether to charge or not the internal battery at the nominal rate.

The resulting optimization program is given by

$$\begin{aligned} \min_{\pi_1,\dots,\pi_N} \quad & \sum_{i=1}^N \rho_i^\top \pi_i \qquad (5) \\ \text{subject to:} \quad & \sum_{i=1}^N \pi_i \le \bar{\pi} \\ & \pi_i \in \Pi_i \qquad i = 1, \dots, N, \end{aligned}$$

where, for each vehicle i, vector π_i is its charging power profile, Π_i encodes its local constraints such as power limits, battery storage limits, and desired final state of charge, ρ_i its cost of buying one unit of power for an entire time-slot, and $\bar{\pi}$ is the maximum power that the grid can provide at each time-slot. For a precise formulation of Π_i, we refer the reader to Vujanic et al. (2016).

It is easy to see how (5) fits the structure of $\mathcal{P}$. Indeed it is sufficient to take $x_i = \pi_i$, $f_i(x_i) = \rho_i^\top \pi_i$, $X_i = \Pi_i$, and $h_i(x_i) = \pi_i - \frac{\bar{\pi}}{N}$. Since we do not have any equality coupling constraint, we set $p = 0$ and we do not define $A_1, \dots, A_N$, and b.

In our simulation we considered a fleet of $N = 50$ vehicles. Each vehicle i has $n_i = 24$ decision variables representing the charging rate over the 24 20-minute time-slots making up the 8 hour charging horizon. The local constraint set Π_i is defined by 95 inequalities and the number of inequality coupling constraints, and thus the size of the Lagrange multiplier vector μ, is $q = 24$.

As for Assumptions 3 and 4, we adopt the same procedure of the previous example for building the communication graph and the consensus matrix. We then run Algorithm 1 for 10^4 iterations, with the penalty parameter c taking values in $\{10^{-3.5}, 10^{-4}, 10^{-4.5}, 10^{-5}, 10^{-5.5}, 10^{-6}\}$, which are those values (spanning two and a half orders of magnitude) that give the best performance among those that we explored.

Note that, since there are no equality coupling constraints in (5), we can skip Steps 3, 6, 8, 12, and 14, and in Step 10 we can remove the terms involving ℓ_i^k, A_i, and δ_i^k from the cost function.

In Fig. 2 we report, on a semi-logarithmic chart, the behavior across iterations of the relative optimality gap (upper plot), between the value of the cost function achieved by the primal tentative solution x_i^k and the optimal cost $f^\star$ computed by a centralized solver (as in the previous example), and the relative maximum constraint violation (lower plot)

$$\frac{\|\max\{\sum_{i=1}^N h_i(x_i^k), 0\}\|_\infty}{\|\bar{\pi}\|_\infty}$$

of the inequality coupling constraints (lower plot). As can be observed from the picture, the proposed algorithm converges to an optimal solution of $\mathcal{P}$ for all values of c. Furthermore, it is interesting to mention that, differently from the previous example, despite the value of c affects the transient behavior, for this problem convergence is eventually exponential in all cases, with a rate (cf. the slopes of the lines in Fig. 2) not affected by c.

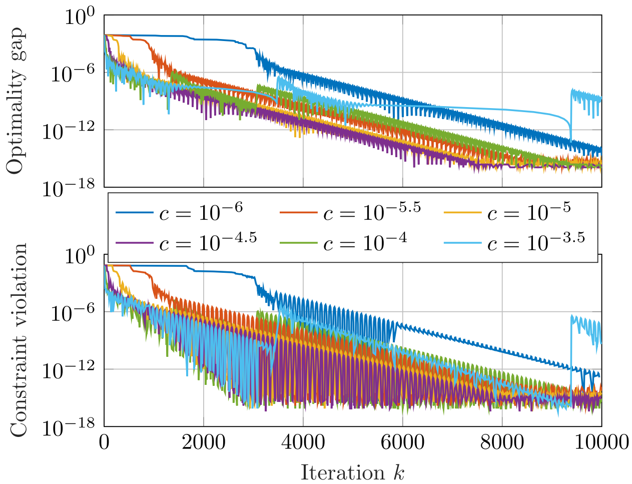

Fig. 2. Relative optimality gap (upper plot) and relative coupling constraint violation (lower plot) of $(x_1^k, \dots, x_N^k)$ across iterations of Augmented Lagrangian Tracking applied to (5), for different values of the penalty parameter c.

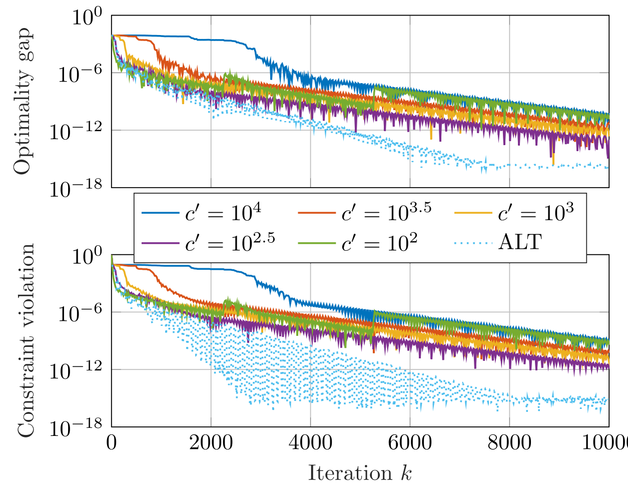

Fig. 3. Relative optimality gap (upper plot) and relative coupling constraint violation (lower plot) of $(x_1^k, \dots, x_N^k)$ across iterations of Chang (2016, Algorithm 1) applied to (5), for different values of the corresponding penalty parameter c' (solid lines). For comparison purposes, we also report the behavior of our Augmented Lagrangian Tracking with $c = 10^{-4.5}$ (ALT, dotted line).

It is worth noticing that the results in Fig. 2 are similar to those reported in Falsone et al. (2020), where the Tracking-ADMM algorithm is applied to a reformulation of the same problem aimed at converting the linear inequality coupling constraints in (5) into linear equality coupling constraints that Tracking-ADMM can handle. The fact that the rate of convergence is similar is not surprising given the intimate relation between the two algorithms described in Section 3.4.

For comparison purposes we solve the same problem using Chang (2016, Algorithm 1), which can be applied since the inequality coupling constraints are affine. In Fig. 3 we report the relative optimality gap (upper plot) and the relative violation of the joint constraint (lower plot) associated to the sequences generated by Chang (2016, Algorithm 1) for different values of its penalty parameter, here denoted as c'. For a fair comparison we show the runs of Chang (2016, Algorithm 1) associated to those values of the penalty parameter that achieve the best performance: $c' \in \{10^2, 10^{2.5}, 10^3, 10^{3.5}, 10^4\}$. By comparing Figs. 2 and 3, we can see how the proposed algorithm outperforms the one in Chang (2016) in terms of convergence rate both for optimality and feasibility. This is testified by the slopes of the curves in Fig. 2, which are steeper than those in Fig. 3, irrespective of the value of the penalty coefficient. To ease the comparison, we also report in Fig. 3 the best run of Augmented Lagrangian Tracking ($c = 10^{-4.5}$).

Since Augmented Lagrangian Tracking requires the resolution of an optimization problem at each iteration (cf. Step 10), it is worth assessing its robustness in case an approximate minimization is carried out. To this purpose, we run Algorithm 1 for 10^4 iterations on the same problem, for different values of the solver (CPLEX 12.10) tolerance $\varepsilon \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$. As can be seen from Fig. 4, the solver tolerance does have an impact on the overall performance. However, the accuracy of the solution retrieved by Augmented Lagrangian Tracking never exceeds the tolerance used by the solver, which shows that Augmented Lagrangian Tracking is robust against numerical errors in the minimization of Step 10. The fact that Augmented Lagrangian Tracking behaves the same for any tolerance $\varepsilon \leq 10^{-5}$ is probably due to (5) being a linear program. Indeed, for linear programs, at least one optimal solution is on a vertex of the feasible set and, if the cost of any other (non-optimal) vertex exceeds the optimal cost by an amount greater than 10^{-5}, when $\varepsilon \leq 10^{-5}$ a suboptimal vertex cannot be selected as optimal by the solver in place of an optimal one.

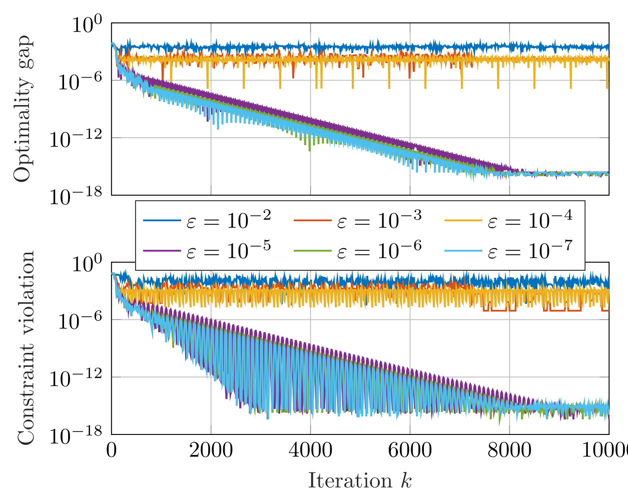

Fig. 4. Relative optimality gap (upper plot) and relative coupling constraint violation (lower plot) of $(x_1^k, \dots, x_N^k)$ across iterations of Augmented Lagrangian Tracking applied to (5), for different solver tolerances ε.

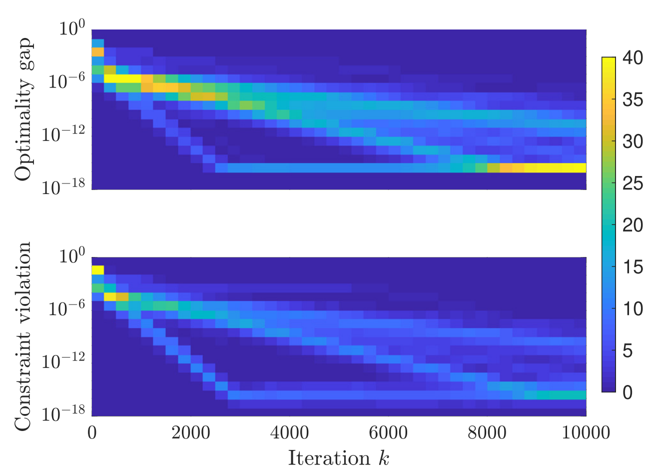

Fig. 5. Histogram of the relative optimality gap (upper plot) and relative coupling constraint violation (lower plot) of $(x_1^k, \dots, x_N^k)$ across iterations of Augmented Lagrangian Tracking with $c = 10^{-4.5}$ applied to 100 instances of (5).

Finally, we also report a Monte Carlo analysis of the performance of Augmented Lagrangian Tracking on 100 instances of (5), with $c = 10^{-4.5}$. Results are shown in Fig. 5 by means of 2D histograms. The iteration vs. accuracy plane is discretized into cells of 250 iterations width and 1 order of magnitude height. A color is assigned to each cell based on how many runs fall within that cell. Since a run may have an accuracy within the cell accuracy range for a number of iterations smaller than 250, each run is weighted proportionally to how many iterations out of 250 the run stays within the cell. This way if one run falls within a cell for one iteration only, its weight is set to 1/250, whereas if it stays within the cell for all the 250 iterations, its weight is set to 1. From the histograms we can see that the instance used for the previous analysis is actually an average instance, as there are other instances for which Augmented Lagrangian Tracking converges as fast as $3 \cdot 10^3$ iterations and others for which its convergence is slower. Note that, on most instances, we achieve an acceptable accuracy ($\leq 10^{-6}$) within $5 \cdot 10^3$ iterations.

5. Conclusions

In this paper we proposed a novel distributed optimization algorithm to solve almost-separable multi-agent optimization problems coupled by affine equality and nonlinear inequality constraints. In contrast with the approaches available in the literature, the proposed algorithm works under very mild assumptions and proved to be effective both on an artificial example with complexity features that make other state-of-the-art algorithms not applicable, and on a realistic application involving the charging schedule of a fleet of electric vehicles. As a future research direction we plan to provide a convergence rate for the proposed algorithm and investigate more deeply the connections with other centralized/distributed algorithms in the literature.

Appendix. Proofs

In this appendix we provide the proofs of the theoretical results presented in Section 3.3. We start by introducing an auxiliary result, which is an optimality condition for the sum of two extended-real convex functions that is used for proving Theorem 1. Then, in the proof of Theorem 1, we build upon the optimality conditions of (3c) and show how they can be manipulated, together with (3d) and (3e), to solve (3c) explicitly with respect to λ_i^{k+1} and μ_i^{k+1} and thus turn (3) into Algorithm 1. Then in the proof of Theorem 2 we show how the convergence of Algorithm 1 to the dual optimal solution implies first feasibility of the primal iterates and then, owing to Step 10, also cost-optimality. We conclude with the proof of Corollary 2, which shows that boundedness of the optimal solutions set together with boundedness of the cost and constraint violation sequences implies boundedness of the $\{(x_1^k, \dots, x_N^k)\}_{k\geq 0}$ sequence and hence optimality of all its limit points.

Auxiliary result

Let J be a function, recall that $\mathrm{dom}(J)$ denotes the domain of J and $\mathrm{relint}(\cdot)$ the relative interior of its argument. The following result is an extension of Bertsekas and Tsitsiklis (1989, Lemma 4.1, p. 257).

Lemma 1. *Let $J_1, J_2 : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be two extended real-valued proper convex functions (cf. Rockafellar, 1970, p. 24) such that $\mathrm{relint}(\mathrm{dom}(J_1)) \cap \mathrm{relint}(\mathrm{dom}(J_2)) \neq \emptyset$. Then,*

$$z^\star \in \arg\min_z J_1(z) + J_2(z)$$

if and only if there exists $\eta_2 \in \partial J_2(z^\star)$ such that

$$z^\star \in \arg\min_z J_1(z) + \eta_2^\top z.$$

Proof. By Rockafellar (1970, Theorem 23.2), $z^\star \in \arg\min_z J_1(z) + J_2(z)$ if and only if

$$0 \in \partial(J_1 + J_2)(z^\star) = \partial J_1(z^\star) \oplus \partial J_2(z^\star),$$

where the equality is due to Rockafellar (1970, Theorem 23.8). By definition of Minkowski sum, the previous inclusion holds if and only if there exist $\eta_1 \in \partial J_1(z^\star)$ and $\eta_2 \in \partial J_2(z^\star)$ such that $0 = \eta_1 + \eta_2$. This is true if and only if $-\eta_2 \in \partial J_1(z^\star)$, which, by Rockafellar (1970, Theorem 23.5), holds if and only if

$$z^\star \in \arg\min_z J_1(z) - (-\eta_2)^\top z = \arg\min_z J_1(z) + \eta_2^\top z,$$

thus concluding the proof. □

Proof of Theorem 1 (Equivalence)

Under Assumption 1 all f_i and h_i are closed proper convex functions. We start by noticing that, under Assumption 1, $-\varphi_i$ is a closed proper convex function, according to Rockafellar (1970, Theorem 9.4). Under Assumption 2, $-\sum_{i=1}^N \varphi_i$ is also closed proper convex by Rockafellar (1970, Theorem 9.3). This makes $\mathcal{D}$ fit the structure of Falsone and Prandini (2022, $\mathcal{P}$, see discussion after Assumption 2).

Given the minimization in (3c), for $(\lambda_i^{k+1}, \mu_i^{k+1})$ to be the optimal solution, by Rockafellar (1970, Theorem 23.2), it must hold that

$$0 \in \partial\Big(-\varphi_i(\lambda_i, \mu_i) + \mathcal{I}_{\mathbb{R}_+^q}(\mu_i) + (\delta_i^k - u_i^k)^\top \lambda_i + \tfrac{1}{2c}\|\lambda_i - \ell_i^k\|^2 + (\gamma_i^k - v_i^k)^\top \mu_i + \tfrac{1}{2c}\|\mu_i - m_i^k\|^2 \Big)(\lambda_i^{k+1}, \mu_i^{k+1}), \quad (6)$$

where the objective function has been augmented with the indicator function $\mathcal{I}_{\mathbb{R}_+^q}(\mu_i)$ to account for the non-negativity constraints on μ_i. Under Assumption 2, $(\mathbb{R}^p \times \mathbb{R}_+^q) \cap \mathrm{relint}(\mathrm{dom}(-\varphi_i)) \neq \emptyset$ and, by Rockafellar (1970, Theorem 23.8), (6) is equivalent to

$$\begin{aligned}
0 \;&\in\; \partial(-\varphi_i)(\lambda_i^{k+1}, \mu_i^{k+1}) \oplus \partial\mathcal{I}_{\mathbb{R}^p \times \mathbb{R}_+^q}(\lambda_i^{k+1}, \mu_i^{k+1}) \\
&\quad \oplus \partial\Big((\delta_i^k - u_i^k)^\top \lambda_i + \tfrac{1}{2c}\|\lambda_i - \ell_i^k\|^2 \\
&\qquad + (\gamma_i^k - v_i^k)^\top \mu_i + \tfrac{1}{2c}\|\mu_i - m_i^k\|^2\Big)(\lambda_i^{k+1}, \mu_i^{k+1}) \\
&\overset{(a)}{=} \partial(-\varphi_i)(\lambda_i^{k+1}, \mu_i^{k+1}) \oplus \partial\mathcal{I}_{\mathbb{R}^p \times \mathbb{R}_+^q}(\lambda_i^{k+1}, \mu_i^{k+1}) \\
&\quad \oplus \left\{\begin{bmatrix} (\delta_i^k - u_i^k) + \frac{1}{c}(\lambda_i^{k+1} - \ell_i^k) \\ (\gamma_i^k - v_i^k) + \frac{1}{c}(\mu_i^{k+1} - m_i^k) \end{bmatrix}\right\}, \\
&\overset{(3d)}{=} \partial(-\varphi_i)(\lambda_i^{k+1}, \mu_i^{k+1}) \oplus \partial\mathcal{I}_{\mathbb{R}^p \times \mathbb{R}_+^q}(\lambda_i^{k+1}, \mu_i^{k+1}) \\
&\quad \oplus \left\{\begin{bmatrix} -u_i^{k+1} \\ -v_i^{k+1} \end{bmatrix}\right\},
\end{aligned} \quad (7)$$

where (a) is due to the linear and quadratic terms being differentiable. According to Rockafellar (1970, p. 226),

$$\partial\mathcal{I}_{\mathbb{R}^p \times \mathbb{R}_+^q}(\lambda, \mu) = \big\{(0_p, -\sigma) :\; \sigma \in \mathbb{R}_+^q \wedge \sigma^\top \mu = 0\big\} \quad (8)$$

while, according to Danskin's theorem (see Bertsekas, 1999, Proposition B.25),

$$\partial(-\varphi_i)(\lambda, \mu) = \Big\{-(A_i x_i' - b_i, h_i(x_i')) : x_i' \in \arg\min_{x_i \in X_i} f_i(x_i) + \lambda^\top A_i x_i + \mu^\top h_i(x_i)\Big\}. \quad (9)$$

Therefore, to satisfy (7) given (8) and (9) together, there must exist $x_i^{k+1} \in \mathbb{R}^{n_i}$ and $\sigma_i^{k+1} \in \mathbb{R}_+^q$ such that

$$x_i^{k+1} \in \arg\min_{x_i \in X_i} f_i(x_i) + {\lambda_i^{k+1}}^\top A_i x_i + {\mu_i^{k+1}}^\top h_i(x_i) \quad (10a)$$

$$u_i^{k+1} = -(A_i x_i^{k+1} - b_i) \tag{10b}$$
$$v_i^{k+1} = -h_i(x_i^{k+1}) - \sigma_i^{k+1} \tag{10c}$$
$$\sigma_i^{k+1\top} \mu_i^{k+1} = 0. \tag{10d}$$

Relations (10b) and (10c) can be used in (3e) to get

$$d_i^{k+1} = \delta_i^k - A_i x_i^{k+1} + A_i x_i^k \tag{11a}$$
$$g_i^{k+1} = \gamma_i^k - h_i(x_i^{k+1}) - \sigma_i^{k+1} + h_i(x_i^k) + \sigma_i^k, \tag{11b}$$

while (3d) can be made explicit in λ_i^{k+1} and μ_i^{k+1} as

$$\lambda_i^{k+1} = \ell_i^k + c\,(A_i x_i^{k+1} - A_i x_i^k - \delta_i^k) \tag{12a}$$
$$\stackrel{(11a)}{=} \ell_i^k - c\, d_i^{k+1} \tag{12b}$$
$$\mu_i^{k+1} = m_i^k + c\,\sigma_i^{k+1} + c\,(h_i(x_i^{k+1}) - h_i(x_i^k) - \sigma_i^k - \gamma_i^k) \tag{12c}$$
$$\stackrel{(11b)}{=} m_i^k - c\, g_i^{k+1}. \tag{12d}$$

Up to now, we have shown that (10)–(12) are equivalent to (3c)–(3e), but, unfortunately, x_i^{k+1} and σ_i^{k+1} are defined only implicitly. Indeed, substituting (12a) and (12c) in (10a) reveals that the cost function of (10a) depends on x_i^{k+1}, which is the outcome of the optimization in (10a). Similarly, substituting (12c) in (10d) reveals that σ_i^{k+1} must satisfy a quadratic equation. In the remaining part of the proof we will show that x_i^{k+1} and σ_i^{k+1} can actually be computed explicitly using information available at iteration k only.

The quantity σ_i^{k+1} is related to the non-negativity constraints that μ_i^{k+1} must comply with. Therefore, given (12c), it is intuitive to set

$$\sigma_i^{k+1} = \max\left\{\gamma_i^k - h_i(x_i^{k+1}) + h_i(x_i^k) + \sigma_i^k - \tfrac{1}{c} m_i^k, 0\right\} \tag{13}$$

so that

$$\begin{aligned}\mu_i^{k+1} &\stackrel{(12c)}{=} c\,\sigma_i^{k+1} + m_i^k + c\,(h_i(x_i^{k+1}) - h_i(x_i^k) - \sigma_i^k - \gamma_i^k)\\ &\stackrel{(13)}{=} c\max\left\{\gamma_i^k - h_i(x_i^{k+1}) + h_i(x_i^k) + \sigma_i^k - \tfrac{1}{c} m_i^k, 0\right\}\\ &\quad + m_i^k + c\,(h_i(x_i^{k+1}) - h_i(x_i^k) - \sigma_i^k - \gamma_i^k)\\ &= \max\left\{m_i^k + c\,(h_i(x_i^{k+1}) - h_i(x_i^k) - \sigma_i^k - \gamma_i^k), 0\right\}\end{aligned} \tag{14}$$

and (10d) is trivially satisfied.

Now, the only quantity left to be determined is x_i^{k+1}. Given (10a), by Lemma 1 setting $z = x_i$, $J_1(x_i) = f_i(x_i) + \mathcal{I}_{X_i}(x_i)$, and $J_2(x_i) = \lambda_i^{k+1\top} A_i x_i + \mu_i^{k+1\top} h_i(x_i)$, there must exist $\eta_i^{k+1} \in \partial J_2(x_i^{k+1})$ such that

$$x_i^{k+1} \in \arg\min_{x_i \in X_i} f_i(x_i) + \eta_i^{k+1\top} x_i. \tag{15}$$

Moreover,

$$\begin{aligned}\partial J_2 &\stackrel{(a)}{=} \partial(\lambda_i^{k+1\top} A_i x_i) \oplus \partial(\mu_i^{k+1\top} h_i(x_i))\\ &\stackrel{(b)}{=} \{A_i^\top \lambda_i^{k+1}\} \oplus \partial(\mu_i^{k+1\top} h_i(x_i))\\ &\stackrel{(a)}{=} \{A_i^\top \lambda_i^{k+1}\} \oplus \bigoplus_{s=1}^{q} \partial([\mu_i^{k+1}]_s [h_i(x_i)]_s)\\ &\stackrel{(c)}{=} \{A_i^\top \lambda_i^{k+1}\} \oplus \bigoplus_{s=1}^{q} [\mu_i^{k+1}]_s \partial([h_i(x_i)]_s),\end{aligned}$$

where (a) are both due to Rockafellar (1970, Theorem 23.8) under Assumption 1, (b) is given by differentiability of the function inside the first term, and (c) trivially follows from the definition of subgradient together with each $[\mu_i^{k+1}]_s$ being a non-negative scalar. By definition of Minkowski sum, there must exist $\hbar_{i,s}^{k+1} \in \partial([h_i(x_i)]_s)(x_i^{k+1})$, for all $s = 1, \dots, q$, such that

$$\eta_i^{k+1} = A_i^\top \lambda_i^{k+1} + H_i^{k+1} \mu_i^{k+1}, \tag{16}$$

with $H_i^{k+1} = [\hbar_{i,1}^{k+1} \cdots \hbar_{i,q}^{k+1}]$. Let us now consider the function

$$\begin{aligned}\tilde{J}_2(x_i) &= \ell_i^{k\top} A_i x_i + \tfrac{c}{2}\|A_i x_i - A_i x_i^k - \delta_i^k\|^2\\ &\quad + \tfrac{1}{2c}\|\max\{m_i^k + c\,(h_i(x_i) - h_i(x_i^k) - \sigma_i^k - \gamma_i^k), 0\}\|^2,\end{aligned}$$

and show that $\eta_i^{k+1} \in \partial\tilde{J}_2(x_i^{k+1})$. First, let us rewrite $\tilde{J}_2(x_i)$ as $\tilde{J}_2(x_i) = J_3(x_i) + J_4(J_5(x_i))$ with $J_3 : \mathbb{R}^{n_i} \to \mathbb{R}$, $J_4 : \mathbb{R}^q \to \mathbb{R}$, and $J_5 : \mathbb{R}^{n_i} \to \mathbb{R}^q$ defined as

$$J_3(x_i) = \ell_i^{k\top} A_i x_i + \tfrac{c}{2}\|A_i x_i - A_i x_i^k - \delta_i^k\|^2$$
$$J_4(y) = \tfrac{1}{2c}\|\max\{y, 0\}\|^2$$
$$J_5(x_i) = m_i^k + c\,(h_i(x_i) - h_i(x_i^k) - \sigma_i^k - \gamma_i^k).$$

By convexity and differentiability of $J_3(x_i)$, we have

$$\begin{aligned}J_3(x_i) - J_3(x_i^{k+1}) &\geq \nabla J_3(x_i^{k+1})^\top (x_i - x_i^{k+1})\\ &= (A_i^\top \ell_i^k + c\,A_i^\top (A_i x_i^{k+1} - A_i x_i^k - \delta_i^k))^\top (x_i - x_i^{k+1})\\ &\stackrel{(12a)}{=} (A_i^\top \lambda_i^{k+1})^\top (x_i - x_i^{k+1}).\end{aligned} \tag{17}$$

Similarly, by convexity and differentiability of $J_4(y)$, we have

$$\begin{aligned}J_4(y) &\geq J_4(\bar{y}) + \nabla J_4(\bar{y})^\top (y - \bar{y})\\ &= J_4(\bar{y}) + \tfrac{1}{c}\max\{\bar{y}, 0\}^\top (y - \bar{y}),\end{aligned} \tag{18}$$

and, lastly, by convexity of $[h_i(x_i)]_s$ together with $\hbar_{i,s}^{k+1} \in \partial([h_i(x_i)]_s)(x_i^{k+1})$, we have

$$[h_i(x_i)]_s \geq [h_i(x_i^{k+1})]_s + \hbar_{i,s}^{k+1\top}(x_i - x_i^{k+1}), \tag{19}$$

for all $s = 1, \dots, q$. Considering (19) in its vector form, multiplying by c, and adding $m_i^k - c\,(h_i(x_i^k) + \sigma_i^k + \gamma_i^k)$ on both sides we obtain

$$J_5(x_i) \geq J_5(x_i^{k+1}) + c\,H_i^{k+1\top}(x_i - x_i^{k+1}). \tag{20}$$

Owing to (20) together with the fact that $y_1 \geq y_2$ implies $J_4(y_1) \geq J_4(y_2)$, we have

$$\begin{aligned}J_4(J_5(x_i)) &\geq J_4(J_5(x_i^{k+1}) + c\,H_i^{k+1\top}(x_i - x_i^{k+1}))\\ &\stackrel{(a)}{\geq} J_4(J_5(x_i^{k+1})) + \max\{J_5(x_i^{k+1}), 0\}^\top H_i^{k+1\top}(x_i - x_i^{k+1})\\ &\stackrel{(14)}{=} J_4(J_5(x_i^{k+1})) + \mu_i^{k+1\top} H_i^{k+1\top}(x_i - x_i^{k+1})\end{aligned} \tag{21}$$

where (a) is due to (18) setting $y = J_5(x_i^{k+1}) + c\,H_i^{k+1\top}(x_i - x_i^{k+1})$ and $\bar{y} = J_5(x_i^{k+1})$. Summing (17) and (21) and recalling the definition of $\tilde{J}_2$ we obtain

$$\begin{aligned}\tilde{J}_2(x_i) &\geq \tilde{J}_2(x_i^{k+1}) + (A_i^\top \lambda_i^{k+1} + H_i \mu_i^{k+1})^\top (x_i - x_i^{k+1})\\ &\stackrel{(16)}{=} \tilde{J}_2(x_i^{k+1}) + \eta_i^{k+1\top}(x_i - x_i^{k+1}),\end{aligned}$$

which proves that $\eta_i^{k+1} \in \partial\tilde{J}_2(x_i^{k+1})$. Recalling (15), by Lemma 1 setting $z = x_i$, $J_1(x_i) = f_i(x_i) + \mathcal{I}_{X_i}(x_i)$, and $J_2(x_i) = \tilde{J}_2(x_i)$, x_i^{k+1} must also satisfy

$$x_i^{k+1} \in \arg\min_{x_i \in X_i} f_i(x_i) + \tilde{J}_2(x_i), \tag{22}$$

which is a convex minimization problem, whose cost function depends on quantities that are available at iteration k.

Therefore, updates (3c)–(3e) can be equivalently implemented using (22), (13), (11), and (12), which are equal to Steps 10–15 in Algorithm 1. The consensus updates (3a)–(3b) are trivially equivalent to Steps 6–9.

Lastly, we need to make sure that the initialization of Algorithm 1 matches that of the Proximal-Tracking in Falsone and Prandini (2022). As for the dual variables, the initialization $(\lambda_i^0, \mu_i^0) \in \mathbb{R}^p \times \mathbb{R}^q$ is the same (cf. Steps 3 and 4). Then, according to Falsone and Prandini (2022), (u_i^0, v_i^0) can be chosen arbitrarily in $\mathbb{R}^p \times \mathbb{R}^q$ while $(d_i^0, g_i^0) = (u_i^0, v_i^0)$. Since by (10b) and (10c) $u_i^k = -(A_i x_i^k - b_i)$ and $v_i^k = h_i(x_i^k) - \sigma_i^k$, we then select $x_i^0 \in \mathbb{R}^{n_i}$ and $\sigma_i^0 \in \mathbb{R}^q$ arbitrarily (cf. Step 2), so that $u_i^0 = -(A_i x_i^0 - b_i)$ and $v_i^0 = -h_i(x_i^0) - \sigma_i^0$ are arbitrary vectors, and then set $(d_i^0, g_i^0) = (-(A_i x_i^0 - b_i), -h_i(x_i^0) - \sigma_i^0) = (u_i^0, v_i^0)$ (cf. Steps 3 and 4).

This shows that Algorithm 1 is equivalent to the iterative scheme in (3), thus concluding the proof. □

Proof of Theorem 2 (Primal Optimality)

Under Assumptions 1–4, owing to Corollary 1,

$$\lim_{k\to\infty} (\lambda_i^k, \mu_i^k) = (\lambda^\star, \mu^\star), \tag{23}$$

for all $i = 1, \dots, N$. By Steps 14 and 15 together with Steps 6 and 7,

$$\begin{aligned}(\lambda_i^{k+1}, \mu_i^{k+1}) &= (\ell_i^k, m_i^k) - c\,(d_i^{k+1}, g_i^{k+1}) \\ &= \sum_{j\in\mathcal{N}_i} w_{ij}(\lambda_j^k, \mu_j^k) - c\,(d_i^{k+1}, g_i^{k+1}) \\ &= \sum_{j=1}^N w_{ij}(\lambda_j^k, \mu_j^k) - c\,(d_i^{k+1}, g_i^{k+1}),\end{aligned}$$

where the last equality is due to $w_{ij} = 0$ if $(i,j) \notin \mathcal{E}$. Summing the previous relation over $i = 1, \dots, N$,

$$\begin{aligned}\sum_{i=1}^N (\lambda_i^{k+1}, \mu_i^{k+1}) &= \sum_{i=1}^N \sum_{j=1}^N w_{ij}(\lambda_j^k, \mu_j^k) - c \sum_{i=1}^N (d_i^{k+1}, g_i^{k+1}) \\ &\overset{(a)}{=} \sum_{j=1}^N \sum_{i=1}^N w_{ij}(\lambda_j^k, \mu_j^k) - c \sum_{i=1}^N (d_i^{k+1}, g_i^{k+1}) \\ &\overset{(b)}{=} \sum_{j=1}^N (\lambda_j^k, \mu_j^k) - c \sum_{i=1}^N (d_i^{k+1}, g_i^{k+1}) \\ &\overset{(c)}{=} \sum_{j=1}^N (\lambda_j^k, \mu_j^k) - c \sum_{i=1}^N (u_i^{k+1}, v_i^{k+1}),\end{aligned}$$

where in (a) we exchanged the two summations, in (b) we used the property $\sum_{i=1}^N w_{ij} = 1$ granted by Assumption 4, and in (c) we used the tracking property in Falsone and Prandini (2022, Lemma 3), which states that, under Assumption 4, $\sum_{i=1}^N (d_i^k, g_i^k) = \sum_{i=1}^N (u_i^k, v_i^k)$ for all $k \geq 0$. Taking the limit on both sides of the previous relation and recalling (23), yields

$$\lim_{k\to\infty} \sum_{i=1}^N (u_i^k, v_i^k) = 0. \tag{24}$$

Summing (10b) and (10c) in the proof of Theorem 1 over $i = 1, \dots, N$, gives

$$-\sum_{i=1}^N u_i^k = \sum_{i=1}^N A_i x_i^k - b_i \tag{25a}$$

$$-\sum_{i=1}^N v_i^k = \sum_{i=1}^N h_i(x_i^k) + \sigma_i^k \overset{(a)}{\geq} \sum_{i=1}^N h_i(x_i^k), \tag{25b}$$

where (a) is due to $\sigma_i^k \geq 0$ for all $i = 1, \dots, N$ and for all $k \geq 0$ (cf. Step 11). Given (24) and (25), we have

$$\lim_{k\to\infty} \sum_{i=1}^N A_i x_i^k = b, \tag{26a}$$

$$\lim_{k\to\infty} \sum_{i=1}^N h_i(x_i^k) + \sigma_i^k = 0, \tag{26b}$$

$$\limsup_{k\to\infty} \sum_{i=1}^N h_i(x_i^k) \leq 0, \tag{26c}$$

which, together with $x_i^k \in X_i$ for all $i = 1, \dots, N$ and for all $k \geq 0$, implies that the sequence $\{(x_1^k, \dots, x_N^k)\}_{k\geq 0}$ is, in the limit, approaching the feasible set of $\mathcal{P}$.

By (10a) in the proof of Theorem 1 and by (2),

$$\begin{aligned}\varphi_i(\lambda_i^k, \mu_i^k) &= f_i(x_i^k) + {\lambda_i^k}^\top (A_i x_i^k - b_i) + {\mu_i^k}^\top h_i(x_i^k) \\ &\overset{(a)}{=} f_i(x_i^k) + {\lambda_i^k}^\top (A_i x_i^k - b_i) + {\mu_i^k}^\top (h_i(x_i^k) + \sigma_i^k),\end{aligned}$$

where in (a) we added ${\mu_i^k}^\top \sigma_i^k \overset{(10d)}{=} 0$ on the right hand side. Summing the previous relation over $i = 1, \dots, N$, yields

$$\begin{aligned}\sum_{i=1}^N \varphi_i(\lambda_i^k, \mu_i^k) = &\sum_{i=1}^N f_i(x_i^k) + \sum_{i=1}^N {\lambda_i^k}^\top (A_i x_i^k - b_i) \\ &+ \sum_{i=1}^N {\mu_i^k}^\top (h_i(x_i^k) + \sigma_i^k).\end{aligned} \tag{27}$$

Since, by Corollary 1, sequences $\{A_i x_i^k - b_i\}_{k\geq 0}$ and $\{h_i(x_i^k) + \sigma_i^k\}_{k\geq 0}$ are convergent and thus bounded for all $i = 1, \dots, N$, then, by (23), (26a) and (26b), we have

$$\lim_{k\to\infty} \sum_{i=1}^N {\lambda_i^k}^\top (A_i x_i^k - b_i) = 0, \tag{28a}$$

$$\lim_{k\to\infty} \sum_{i=1}^N {\mu_i^k}^\top (h_i(x_i^k) + \sigma_i^k) = 0. \tag{28b}$$

By taking the lim sup on both sides of (27) and using (28), we obtain

$$\begin{aligned}\limsup_{k\to\infty} \sum_{i=1}^N f_i(x_i^k) &= \limsup_{k\to\infty} \sum_{i=1}^N \varphi_i(\lambda_i^k, \mu_i^k) \\ &\overset{(a)}{\leq} \sum_{i=1}^N \varphi_i(\lambda^\star, \mu^\star) \\ &\overset{(b)}{=} f^\star,\end{aligned} \tag{29}$$

where (a) is due to lower semi-continuity of $-\varphi_i$ for all $i = 1, \dots, N$ granted by closedness (see Rockafellar, 1970, p. 52) and (b) is due to strong duality, granted by Assumption 2. On the other hand, by strong duality and (2),

$$\begin{aligned}f^\star &= \sum_{i=1}^N \varphi_i(\lambda^\star, \mu^\star) \\ &\leq \sum_{i=1}^N f_i(x_i^k) + {\lambda^\star}^\top (A_i x_i^k - b_i) + {\mu^\star}^\top h_i(x_i^k),\end{aligned}$$

from which, taking the lim inf on both sides and using (26) together with $\mu^\star \geq 0$, we get

$$f^\star \leq \liminf_{k\to\infty} \sum_{i=1}^N f_i(x_i^k). \tag{30}$$

From (29) and (30), we have $\lim_{k\to\infty} \sum_{i=1}^N f_i(x_i^k) = f^\star$.

This shows that the sequence $\{(x_1^k, \dots, x_N^k)\}_{k\geq 0}$ is, in the limit, approaching both the feasible set and the optimal value of $\mathcal{P}$, thus concluding the proof. □

Proof of Corollary 2 (Primal Limit Points)

Recall that $X = X_1 \times \cdots \times X_N$ and define the set $C^\alpha = \{(x_1, \dots, x_N) : \varsigma(x_1, \dots, x_N) \le \alpha\}$, where

$$\varsigma(x_1, \dots, x_N) = \max\Bigg\{ \max_{s=1,\dots,q} \sum_{i=1}^{N} [h_i(x_i)]_s, \max_{s=1,\dots,p} \left| \sum_{i=1}^{N} [A_i x_i]_s - [b]_s \right|, \sum_{i=1}^{N} f_i(x_i) - f^\star \Bigg\}.$$

Since under Assumption 1 $\varsigma(x_1, \dots, x_N)$ is convex, closed and proper, then, the set C^α is convex and closed for any $\alpha \ge 0$.

Furthermore, $C^\alpha \supseteq C^0$ is also non-empty for any $\alpha \ge 0$, since the set of optimal solutions to $\mathcal{P}$ can be expressed as

$$X^\star = X \cap C^0 \tag{31}$$

and, under the additional assumption of Corollary 2, it is non-empty.

Let rec(·) denote the recession cone of its argument (see Rockafellar, 1970, p. 61). By the closedness of X and C^0 we have that $X^\star$ in (31) is closed and the following chain of equalities holds for every $\alpha \ge 0$:

$$\{0\} \overset{(a)}{=} \mathrm{rec}(X^\star) \overset{(b)}{=} \mathrm{rec}(X) \cap \mathrm{rec}(C^0) \overset{(c)}{=} \mathrm{rec}(X) \cap \mathrm{rec}(C^\alpha) \tag{32}$$

where (a) is due to the boundedness assumption on $X^\star$ and Rockafellar (1970, Theorem 8.4), (b) is due to Rockafellar (1970, Corollary 8.3.3), and (c) holds since $\mathrm{rec}(C^\alpha) = \mathrm{rec}(C^0)$ for all $\alpha \ge 0$ by Rockafellar (1970, Theorem 8.7). Invoking again Rockafellar (1970, Theorem 8.4) shows that $X^\alpha = X \cap C^\alpha$ is bounded for any $\alpha \ge 0$.

Under Assumptions 1–4 and $c > 0$, by Theorem 2 we have that $\lim_{k\to\infty} \sum_{i=1}^{N} f_i(x_i^k) = f^\star$, $\lim_{k\to\infty} \sum_{i=1}^{N} A_i x_i^k = b$, and $\limsup_{k\to\infty} \sum_{i=1}^{N} h_i(x_i^k) \le 0$, meaning that there exist $\bar{\alpha} \ge 0$ such that $\sum_{i=1}^{N} f_i(x_i^k) - f^\star \le \bar{\alpha}$, $| \sum_{i=1}^{N} [A_i(x_i^k)]_s - [b]_s | \le \bar{\alpha}$, $s = 1, \dots, p$, and $\sum_{i=1}^{N} [h_i(x_i^k)]_s \le \bar{\alpha}$, $s = 1, \dots, q$, for all $k \ge 0$, which implies that $\varsigma(x_1^k, \dots, x_N^k) \le \bar{\alpha}$ and, hence, $(x_1^k, \dots, x_N^k) \in X^{\bar{\alpha}}$ for all $k \ge 0$.

Since, by (32), X^α is bounded for all $\alpha \ge 0$, then $X^{\bar{\alpha}}$ is bounded and $\{(x_1^k, \dots, x_N^k)\}_{k\ge0}$ is a bounded sequence. The sequence $\{(x_1^k, \dots, x_N^k)\}_{k\ge0}$ then admits a convergent subsequence with the corresponding limit point. However, by Theorem 2, any limit point is feasible for $\mathcal{P}$ and achieves the optimal cost and is, therefore, an optimal solution of $\mathcal{P}$, thus proving the first part of the statement.

Since the set of limit points is a subset of $X^\star$, if $X^\star = \{x^\star\}$ is a singleton, then we have only one limit point and $\{(x_1^k, \dots, x_N^k)\}_{k\ge0}$ converges to it. This observation proves the second statement of the corollary and thus concludes the proof. □

References

Bertsekas, Dimitri P. (1999). *Nonlinear programming*. Athena Scientific.

Bertsekas, Dimitri P. (2015). *Convex optimization algorithms*. Athena Scientific.

Bertsekas, Dimitri P., & Tsitsiklis, John N. (1989). *Parallel and distributed computation: Numerical methods*: *Vol. 23*, NJ: Prentice hall Englewood Cliffs.

Chang, Tsung-Hui (2016). A proximal dual consensus ADMM method for multi-agent constrained optimization. *IEEE Transactions on Signal Processing*, *64*(14), 3719–3734.

Chang, Tsung-Hui, Nedić, Angelia, & Scaglione, Anna (2014). Distributed constrained optimization by consensus-based primal-dual perturbation method. *IEEE Transactions on Automatic Control*, *59*(6), 1524–1538.

Falsone, Alessandro, Margellos, Kostas, Garatti, Simone, & Prandini, Maria (2017). Dual decomposition for multi-agent distributed optimization with coupling constraints. *Automatica*, *84*, 149–158.

Falsone, Alessandro, Notarnicola, Ivano, Notarstefano, Giuseppe, & Prandini, Maria (2020). Tracking-ADMM for distributed constraint-coupled optimization. *Automatica*, *117*, Article 108962.

Falsone, Alessandro, & Prandini, Maria (2020). A distributed dual proximal minimization algorithm for constraint-coupled optimization problems. *IEEE Control Systems Letters*, *5*(1), 259–264.

Falsone, Alessandro, & Prandini, Maria (2022). Distributed decision-coupled constrained optimization via Proximal-Tracking. *Automatica*, *135*, Article 109938.

Kia, Solmaz S, Van Scoy, Bryan, Cortes, Jorge, Freeman, Randy A, Lynch, Kevin M, & Martinez, Sonia (2019). Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems Magazine*, *39*(3), 40–72.

Liang, Shu, Wang, Le Yi, & Yin, George (2020). Distributed smooth convex optimization with coupled constraints. *IEEE Transactions on Automatic Control*, *63*(1), 347–353.

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

Mota, João F. C., Xavier, João M. F., Aguiar, Pedro M. Q., & Püschel, Markus (2013). D-ADMM: A communication-efficient distributed algorithm for separable optimization. *IEEE Transactions on Signal Processing*, *61*(10), 2718–2723.

Nedić, Angelia, Olshevsky, Alex, & Shi, Wei (2017). Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, *27*(4), 2597–2633.

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

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

Notarnicola, Ivano, & Notarstefano, Giuseppe (2020). Constraint-coupled distributed optimization: A relaxation and duality approach. *IEEE Transactions on Control of Network Systems*, *7*(1), 483–492.

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

Rockafellar, Ralph Tyrell (1970). *Convex analysis*. Princeton University Press.

Romao, Licio, Margellos, Kostas, Notarstefano, Giuseppe, & Papachristodoulou, Antonis (2021). Subgradient averaging for multi-agent optimisation with different constraint sets. *Automatica*, *131*, Article 109738.

Sinkhorn, Richard, & Knopp, Paul (1967). Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, *21*(2), 343–348.

Su, Yanxu, Wang, Qingling, & Sun, Changyin (2021). Distributed primal-dual method for convex optimization with coupled constraints. *IEEE Transactions on Signal Processing*, *70*, 523–535.

Vujanic, Robin, Esfahani, Peyman Mohajerin, Goulart, Paul J, Mariéthoz, Sébastien, & Morari, Manfred (2016). A decomposition method for large scale MILPs, with performance guarantees and a power system application. *Automatica*, *67*, 144–156.

Wang, Jianzheng, & Hu, Guoqiang (2022). Composite optimization with coupling constraints via dual proximal gradient method with applications to asynchronous networks. *International Journal of Robust and Nonlinear Control*, *32*(8), 4752–4770.

Zhu, Minghui, & Martínez, Sonia (2010). Discrete-time dynamic average consensus. *Automatica*, *46*(2), 322–329.

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

Alessandro Falsone received his Master degree in Automation and Control Engineering in 2013, and his Ph.D. degree in Information Technology, in 2018, both with laude and from Politecnico di Milano. During his Ph.D. studies he spent three months as a visiting researcher at the University of Oxford. Since 2018, he is an assistant professor at the Dipartimento di Elettronica, Informazione e Bioingegneria at Politecnico di Milano. His current research interests include distributed optimization and control, data-based optimal control of stochastic systems, and nonlinear model identification. In 2018 he was the recipient of the Dimitris N. Chorafas Prize for his Ph.D. thesis. In 2019 he received the IEEE CSS Italy Chapter Best Young Author Journal Paper Award.

Maria Prandini received the Ph.D. degree in Information Technology from the University of Brescia, Italy, in 1998. She was a postdoctoral researcher at the University of California at Berkeley from 1998 to 2000. She also held visiting positions at Delft University of Technology (1998), Cambridge University (2000), UC Berkeley (2005), ETH Zurich (2006), and University of Oxford (2022). In 2002, she joined Politecnico di Milano, where she is Full Professor since 2018. She has been Chair of the Automation and Control Engineering Program for two consecutive triennia starting from 2019. She was elected Fellow of the IEEE in 2020 and IEEE Control Systems Society Distinguished Member in 2018. In 2017, she was August-Wilhelm Scheer Visiting Professor and Honorary fellow of the TUM Institute for Advanced Studied. She was nominated Visiting Professor in Engineering at the University of Oxford for the triennium 2022–2025. She has been active in the IEEE Control Systems Society (CSS), the International Federation of Automatic Control (IFAC), and the Association for Computing Machinery (ACM), contributing to their activities in different roles. She is IFAC President-elect for the triennium 2023–26. Previously, she was Vice-President for conference activities for IFAC (2020–23) and IEEE CSS (2016 and 2017), and a member of SIGBED Board of Directors (2019–21). Her research interests include stochastic hybrid systems, randomized algorithms, distributed and data-driven optimization, multi-agent systems, and the application of control theory to transportation and energy systems.