Tracking-ADMM for Distributed Constraint-Coupled Optimization[☆]

Alessandro Falsone[a], Ivano Notarnicola[b], Giuseppe Notarstefano[b], Maria Prandini[a]

[a] *Dip. di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Via Ponzio 34/5, 20133 Milano, Italy.*
[b] *Dip. di Ingegneria dell'Energia Elettrica e dell'Informazione "G. Marconi", Alma Mater Studiorum Università di Bologna, Viale del Risorgimento 2, 40136 Bologna, Italy.*

Abstract

We consider constraint-coupled optimization problems in which agents of a network aim to cooperatively minimize the sum of local objective functions subject to individual constraints and a common linear coupling constraint. We propose a novel optimization algorithm that embeds a dynamic average consensus protocol in the parallel Alternating Direction Method of Multipliers (ADMM) to design a fully distributed scheme for the considered set-up. The dynamic average mechanism allows agents to track the time-varying coupling constraint violation (at the current solution estimates). The tracked version of the constraint violation is then used to update local dual variables in a consensus-based scheme mimicking a parallel ADMM step. Under convexity, we prove that all limit points of the agents' primal solution estimates form an optimal solution of the constraint-coupled (primal) problem. The result is proved by means of a Lyapunov-based analysis simultaneously showing consensus of the dual estimates to a dual optimal solution, convergence of the tracking scheme and asymptotic optimality of primal iterates. A numerical study on optimal charging schedule of plug-in electric vehicles corroborates the theoretical results.

Keywords: Distributed Optimization, Constraint-Coupled Optimization, ADMM

1. Introduction

Given the increasing penetration of network-connected devices and cloud-computing services in various application domains (e.g., energy, transportation, manufacturing), systems are becoming more and more large-scale, composed by different interacting agents, and complex. Efficiently operating these systems is challenging: the computational burden associated to the computation of an optimal strategy for the entire system may be prohibitive for a single processing unit and privacy issues may arise if the different agents are forced to disclose sensitive information to the entity in charge of solving the problem. This challenge has been taken up by the control community, which has directed its efforts in devising distributed algorithms for the resolution of such optimization problems.

In this paper we investigate a set-up that is relevant to several network control applications. We consider agents which can communicate only with their neighbors in a network. Each agent knows its local cost function and constraints, that depend on its individual decision variables. Moreover, the agents' decision variables are coupled by a linear constraint. Each agent knows only how its local decision variables affect the coupling constraint. The agents collectively aim at minimizing the sum of the local cost functions subject to both local and coupling constraints. The presence of the coupling element makes the problem solution challenging, especially in the considered distributed context in which no central authority communicating with all the agents is present.

Even though constraint-coupled problems arise naturally in practical applications, until recently, the main effort in distributed optimization has been devoted to solve optimization problems in which there is no coupling constraint but the agents are required to agree on a *common decision*. For this set-up, consensus methods based on (sub)gradient iterations and proximal operators (see Johansson et al. (2008); Nedić and Ozdaglar (2009); Nedić et al. (2010); Zanella et al. (2011); Jakovetić et al. (2014); Shi et al. (2015); Nedić and Olshevsky (2015); Margellos et al. (2018)), distributed algorithms based on duality (see Duchi et al. (2012); Zhu and Martínez (2012)), and distributed approaches based on the Alternating Direction Method Multipliers (ADMM) (see Mota et al. (2013); Ling and Ribeiro (2014); Shi et al. (2014); Jakovetić et al. (2015); Iutzeler et al. (2016); Makhdoumi and Ozdaglar (2017)) have been proposed. Recently, distributed gradient schemes for problems with common decision variables have also been combined with a technique known as dynamic average consensus (firstly pro-

[☆] This research was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 638992 - OPT4SMART) and by the European Commission, under the project UnCoVerCPS, grant number 643921.

Email addresses: alessandro.falsone@polimi.it (Alessandro Falsone), ivano.notarnicola@unibo.it (Ivano Notarnicola), giuseppe.notarstefano@unibo.it (Giuseppe Notarstefano), maria.prandini@polimi.it (Maria Prandini)

posed in Zhu and Martínez (2010) and further elaborated in Kia et al. (2019)), resulting in the so-called gradient-tracking schemes, see, e.g., Di Lorenzo and Scutari (2016); Varagnolo et al. (2016); Nedić et al. (2017); Qu and Li (2018); Xu et al. (2018); Xi et al. (2018). Differently from consensus-based gradient approaches, gradient-tracking methods achieve linear convergence rate thanks to a constant step-size, but at the expense of imposing more restrictive assumptions on the cost functions.

Applying the techniques designed for the common decision set-up to the constraint-coupled one considered in this paper is possible but it would require to define as common decision vector the collection of all local decision vectors. Agents would then have to store and update the entire solution estimate of the problem rather than the portion associated with their decision variables only, and would have all to know the global coupling constraint. These major drawbacks hamper the applicability of those methods to various applications characterized by local decision variables, thus calling for novel and efficient distributed optimization strategies.

Recently developed approaches directly tackling constraint-coupled problems are mostly based on Lagrangian duality. The works Necoara and Nedelcu (2015); Alghunaim et al. (2018); Sherson et al. (2019) consider a similar set-up, but the proposed algorithms rely on the sparsity pattern of the coupling constraints. Moreover, in Sherson et al. (2019), smoothness of the cost function is assumed. In Chang et al. (2014); Mateos-Núnez and Cortés (2017), primal-dual approaches are proposed but they need a diminishing step-size to achieve convergence. In the very recent paper Liang et al. (2019), a primal-dual algorithm with constant step-size is proposed under smoothness assumption on the cost and constraints functions. In Simonetto and Jamali-Rad (2016); Falsone et al. (2017) the dual problem is tackled via distributed subgradient algorithms, in Chang (2016); Wang and Ong (2017) by means of consensus-ADMM schemes, while an alternative approach based on successive duality steps has been investigated in Notarnicola and Notarstefano (2019); however, all these algorithms typically exhibit a slow convergence rate for the local decision variables. In Necoara and Suykens (2009) an interior point framework is exploited, but the developed algorithm requires a central unit. In He et al. (2015); Chang et al. (2018), ADMM-based algorithms where the local decisions variables are updated in a sequential fashion are proposed, but require a central unit to adjust the dual variables. In Chen and Yang (2019) a distributed continuous-time algorithm based on Lagrangian flow and consensus is proposed, but it is tailored to a special case of our set-up. The tracking mechanism has been also employed in Kia (2017) to solve constraint-coupled problems based on an augmented Lagrangian approach in a continuous-time setting, but the considered set-up does not allow for nonsmooth costs and local constraints. Finally, in the very recent papers Zhang and Zavlanos (2018); Carli and Dotoli (2019), two ideas similar to the one proposed in this paper are introduced. However, both approaches require the agents to perform multiple communication rounds and the number of rounds has to be carefully tuned to bound a non-zero steady state error between the algorithm solution and the optimal one.

The contributions of this paper are as follows. We propose a novel, fully distributed optimization algorithm to solve *constraint-coupled problems* over networks by means of an ADMM-based approach. Differently from distributed ADMM schemes for problems with common decision variables, we design our Tracking-ADMM distributed algorithm by embedding a tracking mechanism into the parallel ADMM designed for constraint-coupled problems with a central unit, thus allowing to get rid of it.

The resulting algorithm has the following appealing features: (i) convergence is guaranteed for all the (constant) choices of a penalty parameter and no other coefficients are necessary; (ii) agents solve optimization problems depending on their local (few) decision variables and asymptotically compute only their portion of an optimal (hence feasible) solution to the given problem; (iii) the local estimate of the coupling constraint violation gives each agent a local assessment on the amount of infeasibility, which can be useful, e.g., in designing distributed (receding horizon) control schemes.

The convergence proof of our Tracking-ADMM for constraint-coupled problems relies on control systems analysis tools. By explicitly relying on Lyapunov theory for linear systems, we are able to find a constrained solution to a suitable discrete Lyapunov equation that leads to an aggregate descent condition allowing us to prove: (i) the (exact) convergence of the dual variables to a dual optimal solution and (ii) that any limit point of the primal sequences is an optimal solution of the original constraint-coupled problem. This novel approach allows us to derive a clean and elegant proof of the asymptotic optimality of our algorithm.

The rest of the paper is organized as follows. In Section 2 we present the problem set-up and in Section 2.2 we revise the ADMM algorithm. In Section 3 we introduce our novel Tracking-ADMM distributed algorithm and analyze its convergence properties, discussing the main steps of the proof. In Section 4 we apply our algorithm on a realistic application related to the optimal charging schedule for a fleet of electric vehicles. In Section 5 we draw some conclusions and finally, in Appendix A we report the proofs of all the results stated in the body of the paper.

Notation. The vector in $\mathbb{R}^n$ containing all ones is denoted by $\mathbb{1}_n$. The identity matrix of order n is denoted by I_n. The Kronecker product is denoted by $\otimes$. For a matrix S we write $S^\top$ to denote its transpose, $S \succ 0$ to denote that S is positive definite, $\rho(S)$ to denote the spectral radius of S, and $\|S\|$ for its spectral norm. We write $\text{blkdiag}(S_1, \ldots, S_n)$ to refer to the block-diagonal matrix with $S_1, \ldots, S_n$ as blocks. For a vector v, $\|v\|$ is the Euclidean norm of v, and, for any matrix $S \succ 0$, $\|v\|_S$ is the weighted norm of v, i.e., $\|v\|_S^2 = v^\top S v$.

2. Constraint-Coupled Optimization

In this section we introduce the optimization set-up and recall some preliminaries about the Alternating Direction Method of Multipliers (ADMM). We refer the reader to Bertsekas and Tsitsiklis (1989); Boyd et al. (2011) for an in-depth discussion on ADMM.

2.1. Optimization Problem and Assumptions

We consider a system composed of N agents which are willing to cooperatively solve an optimization program formulated over the entire system. Each agent has to set its local decision variables $x_i \in \mathbb{R}^{n_i}$ so as to minimize the sum of local objective functions $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$, while satisfying local constraints $X_i \subset \mathbb{R}^{n_i}$ as well as a linear constraint that couples the decisions of all the agents. Formally, we address the following mathematical program

$$\begin{aligned} \min_{x_1,\dots,x_N} \quad & \sum_{i=1}^{N} f_i(x_i) \\ \text{subject to:} \quad & \sum_{i=1}^{N} A_i x_i = b \\ & x_i \in X_i \qquad i = 1,\dots,N, \end{aligned} \tag{$\mathcal{P}$}$$

where $A_i \in \mathbb{R}^{p \times n_i}$ and $b \in \mathbb{R}^p$ specify the coupling constraint.

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

Assumption 1 (Convexity and compactness)**.** *For all $i = 1,\dots,N$, the function f_i is convex and the set X_i is convex and compact.* □

Let $\boldsymbol{x} = [x_1^\top \cdots x_N^\top]^\top$, consider a vector $\lambda \in \mathbb{R}^p$ of Lagrange multipliers and define

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

the Lagrangian function obtained by dualizing the coupling constraint $\sum_{i=1}^N A_i x_i = b$. Then, the dual problem of $\mathcal{P}$ is

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

where $X = X_1 \times \cdots \times X_N$, and the i-th contribution φ_i is defined as

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

the vectors $b_1, \dots, b_N$ being such that $\sum_{i=1}^N b_i = b$.

The next assumption ensures that $\mathcal{P}$ and $\mathcal{D}$ are well-posed.

Assumption 2 (Existence of optimal solutions)**.** *Problem $\mathcal{P}$ admits an optimal solution $\boldsymbol{x}^\star = [x_1^{\star\top} \cdots x_N^{\star\top}]^\top$ and problem $\mathcal{D}$ admits an optimal solution $\lambda^\star$.* □

Next, we revise the popular ADMM algorithm which provides an effective way to solve $\mathcal{P}$ by splitting the computation over N agents coordinated by a central unit.

2.2. The ADMM Parallel Algorithm

A version of the ADMM algorithm specifically tailored to problem $\mathcal{P}$ is presented in (Bertsekas and Tsitsiklis, 1989, pag. 254, eq. (4.75)) and is reported here with our notation for the reader's convenience. Given initial values $x_{i,0} \in X_i$, $d_0 = (1/N)\sum_{i=1}^N (A_i x_{i,0} - b_i)$, and $\lambda_0 \in \mathbb{R}^p$, at each iteration $k \geq 0$, a set of N agents and a central unit perform the following two steps. First, all agents, $i = 1,\dots,N$, compute (in parallel) a minimizer of the following optimization problem

$$\begin{aligned} x_{i,k+1} \in \operatorname*{argmin}_{x_i \in X_i} \Big\{ & f_i(x_i) + \lambda_k^\top A_i x_i \\ & + \frac{c}{2} \|A_i x_i - A_i x_{i,k} + d_k\|^2 \Big\}, \end{aligned} \tag{3a}$$

where $c > 0$ is a (constant) penalty parameter. Then, each agent sends the quantity $A_i x_{i,k+1} - b_i$ to the central unit, which computes and broadcasts back to all agents the following two quantities

$$d_{k+1} = \frac{1}{N} \sum_{i=1}^{N} (A_i x_{i,k+1} - b_i) \tag{3b}$$

$$\lambda_{k+1} = \lambda_k + c\, d_{k+1}, \tag{3c}$$

where the parameter c is the same as in (3a). We shall point out that d_{k+1} has no dynamics as it is the *average* of the local contributions $A_i x_{i,k+1} - b_i$ to the coupling constraint and it measures the infeasibility of the current tentative solutions $x_{i,k+1}$, $i = 1,\dots,N$. Its "average" structure is crucial and will be exploited in the design of our Tracking-ADMM distributed algorithm.

The evolution of (3) is analyzed in (Bertsekas and Tsitsiklis, 1989, pp. 254-256) and its convergence property is reported below.

Proposition ((Bertsekas and Tsitsiklis, 1989, Proposition 4.2))**.** *Let Assumptions 1 and 2 hold. Then, any limit point of the primal sequence $\{[x_{1,k}^\top \cdots x_{N,k}^\top]^\top\}_{k \geq 0}$ generated by* (3a)*, is an optimal solution of $\mathcal{P}$, and the dual sequence $\{\lambda_k\}_{k \geq 0}$, generated by* (3c)*, converges to an optimal solution of $\mathcal{D}$.* □

Notice that, since all limit points of the primal sequence are optimal, they are necessarily feasible for the coupling constraint. It thus follows that the sequence $\{d_k\}_{k \geq 0}$ generated by (3b) converges to zero. Moreover, we shall stress that no requirement on the penalty parameter c is necessary for the convergence result to hold, albeit its value can affect the convergence rate of ADMM.

Finally, note that the algorithm described by (3) requires a central unit to compute (3b) and (3c). This hampers the applicability of ADMM to a distributed computation framework, where agents communicate only with neighbors according to a (typically sparse) communication graph. Schemes requiring a central unit are usually referred to as *parallel* schemes. For this reason in the rest of the paper we will refer to (3) as the *parallel ADMM.*

3. Tracking-ADMM Distributed Algorithm

In this section we propose our novel distributed optimization algorithm for constraint-coupled problems. We first introduce the distributed framework and then describe the algorithm along with its convergence properties.

3.1. Distributed Computation Framework

Assume that, at each iteration k, the N agents communicate according to a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, N\}$ is the set of nodes, representing the agents, 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 does not change across iterations and, consequently, $\mathcal{E}$ does not depend on the iteration index k. We denote by $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$ the set of *neighbors* of agent i in $\mathcal{G}$, assuming that $(i, i) \in \mathcal{E}$ for all $i = 1, \ldots, N$. We 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.* □

Each edge $(i, j) \in \mathcal{E}$ has an associated weight w_{ij}, which measures how much agent i values the information received by agent j. For those $(i, j) \notin \mathcal{E}$ we set $w_{ij} = 0$, which models the fact that agent i does not receive any information from agent j. We impose the following assumption on the network weights.

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

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

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., Nedić and Ozdaglar (2009); Nedić et al. (2010).

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

Assumption 5. *$\mathcal{W}$ is positive semidefinite.* □

Note that this assumption is not too restrictive. Indeed, starting from any consensus matrix $\mathcal{W}$ satisfying Assumption 4, we can easily construct (in a distributed way) the matrix $\frac{1}{2}(I + \mathcal{W})$, which satisfies both Assumptions 4 and 5 and matches the connectivity property of the communication graph. In Section 3.3.2 we show that Assumption 5 can also be satisfied if the agents perform two consecutive communication in the same iteration. Indeed, making two communications with any consensus matrix satisfying Assumption 4 is equivalent to a single communication with a consensus matrix satisfying both Assumptions 4 and 5.

3.2. Algorithm Description

In this section, we start from the parallel ADMM and gradually introduce the reader to our proposed algorithm to jointly gain insights about the underlying mechanism and motivate the role of the *consensus and tracking schemes.*

The update (3c) for λ_k in the parallel ADMM resembles a gradient step aiming at maximizing the cost function of $\mathcal{D}$. Distributed implementations of gradient-based approaches to solve optimization problems with common decision variables in the form of $\mathcal{D}$ are well known, see e.g. Nedić and Ozdaglar (2009). Typically, each agent i maintain a vector $\lambda_{i,k} \in \mathbb{R}^p$, representing a local version (or copy) of λ_k, which is iteratively updated according to a consensus-based scheme to force agreement of the local copies. If, at each iteration, the quantity d_{k+1} were available to all agents, then we could propose the following local update step of $\lambda_{i,k}$ for agent i

$$\lambda_{i,k+1} = \sum_{j \in \mathcal{N}_i} w_{ij}\, \lambda_{j,k} + c\, d_{k+1}, \tag{4}$$

for all $i = 1, \ldots, N$.

However, update (4) cannot be implemented in a fully distributed scheme since d_{k+1} is not locally available and should be computed by a central unit (cf. (3b)). In the same spirit of $\lambda_{i,k}$, a distributed counterpart for (3b) can be obtained by equipping each agent i with a local auxiliary quantity $d_{i,k} \in \mathbb{R}^p$, which serves as a local estimate of d_k. Since d_k is the average of $A_i x_{i,k} - b_i$, $i = 1, \ldots, N$ (with $\sum_{i=1}^{N} b_i = b$), we propose to update $d_{i,k}$ according to a (distributed) dynamic average consensus mechanism, see Zhu and Martínez (2010); Kia et al. (2019):

$$d_{i,k+1} = \sum_{j \in \mathcal{N}_i} w_{ij}\, d_{j,k} + (A_i x_{i,k+1} - b_i) - (A_i x_{i,k} - b_i), \tag{5}$$

initialized with $d_{i,0} = A_i x_{i,0} - b_i$, for all $i = 1, \ldots, N$.

In this way, $d_{i,k}$ acts as a *distributed tracker* of the (time-varying) signal $(1/N) \sum_{i=1}^{N} (A_i x_{i,k} - b_i)$. Using $d_{i,k+1}$ in place of d_{k+1} in (4) makes the update of $\lambda_{i,k}$ fully distributed.

Algorithm 1 Tracking-ADMM

```
1: Initialization
2:     $x_{i,0} \in X_i$
3:     $d_{i,0} = A_i x_{i,0} - b_i$
4:     $\lambda_{i,0} \in \mathbb{R}^p$
5: Repeat until convergence
6:     $\delta_{i,k} = \sum_{j \in \mathcal{N}_i} w_{ij}\, d_{j,k}$
7:     $\ell_{i,k} = \sum_{j \in \mathcal{N}_i} w_{ij}\, \lambda_{j,k}$
8:     $x_{i,k+1} \in \operatorname*{argmin}_{x_i \in X_i} \Big\{ f_i(x_i) + \ell_{i,k}^\top A_i x_i + \frac{c}{2}\|A_i x_i - A_i x_{i,k} + \delta_{i,k}\|^2 \Big\}$
9:     $d_{i,k+1} = \delta_{i,k} + A_i x_{i,k+1} - A_i x_{i,k}$
10:    $\lambda_{i,k+1} = \ell_{i,k} + c\, d_{i,k+1}$
11:    $k \leftarrow k + 1$
```

Clearly, since we propose to replace the centralized quantities λ_k and d_k with their local counterparts, the local minimization in (3a) has to be adjusted accordingly. The Tracking-ADMM is formally summarized in Algorithm 1 from the perspective of agent i. Specifically, the adapted local minimization to compute $x_{i,k+1}$ is shown in Step 8, where λ_k and d_k in the original centralized update (3a), are replaced by the local averages $\ell_{i,k}$ and $\delta_{i,k}$, respectively (cf. Steps 6 and 7).

Some remarks are in order. First, we shall stress that all update steps (cf. Steps 8-10) are fully distributed, as they only use quantities that are either locally known to agent i or collected by agent i via neighboring communications (cf. Steps 6 and 7). Moreover, the minimization in Step 8 is always well defined in view of Assumption 1.

While the initialization of $x_{i,k}$ and $\lambda_{i,k}$ can be arbitrary, the correct initialization of $d_{i,k}$ as per Step 3 is crucial, consistently with other tracking-based approaches as, e.g., the ones mentioned in the introduction (cf. Di Lorenzo and Scutari (2016); Varagnolo et al. (2016); Nedić et al. (2017); Qu and Li (2018); Xu et al. (2018); Xi et al. (2018)). Sensible values for initializing Algorithm 1 are given by $x_{i,0} \in \operatorname{argmin}_{x_i \in X_i} f_i(x_i)$, $d_{i,0} = A_i x_{i,0} - b/N$ (which corresponds to setting $b_i = b/N$), and $\lambda_{i,0} = 0$.

Finally, the parameter $c > 0$ in Step 8 and 10 is constant and is similar to the step-size of a gradient-based method, but, differently from gradient-like approaches, its value can be arbitrary. Like for the parallel ADMM, the actual value of c can, however, affect the convergence rate, as shown in Section 4.

3.3. Algorithm Analysis

In this subsection we analyze the proposed Tracking-ADMM algorithm and state its convergence properties. To ease the exposition, we state and discuss the main steps of the convergence analysis, deferring their proofs to Appendix A.

3.3.1. Aggregate Reformulation of Tracking-ADMM

First of all we shall rewrite the execution of Algorithm 1 by all agents in a compact form.

Recalling that $\boldsymbol{x} = [x_1^\top \cdots x_N^\top]^\top$, let us consistently denote with bold symbols the vectors collecting the corresponding quantity of all agents, i.e.,

$$\begin{aligned}
\boldsymbol{x}_k &= [x_{1,k}^\top \cdots x_{N,k}^\top]^\top, \\
\boldsymbol{d}_k &= [d_{1,k}^\top \cdots d_{N,k}^\top]^\top, \qquad & \boldsymbol{\delta}_k &= [\delta_{1,k}^\top \cdots \delta_{N,k}^\top]^\top, \\
\boldsymbol{\lambda}_k &= [\lambda_{1,k}^\top \cdots \lambda_{N,k}^\top]^\top, \qquad & \boldsymbol{\ell}_k &= [\ell_{1,k}^\top \cdots \ell_{N,k}^\top]^\top.
\end{aligned}$$

Then, the evolution of Algorithm 1 over the whole multi-agent network can be compactly written as

$$\begin{aligned}
\boldsymbol{x}_{k+1} \in \operatorname*{argmin}_{\boldsymbol{x} \in X} \Big\{ & f(\boldsymbol{x}) + (W\boldsymbol{\lambda}_k)^\top A_{\mathrm{d}} \boldsymbol{x} \\
& + \frac{c}{2}\|A_{\mathrm{d}}\boldsymbol{x} - A_{\mathrm{d}}\boldsymbol{x}_k + W\boldsymbol{d}_k\|^2 \Big\} \qquad (6a) \\
\boldsymbol{d}_{k+1} = W\boldsymbol{d}_k & + A_{\mathrm{d}}\boldsymbol{x}_{k+1} - A_{\mathrm{d}}\boldsymbol{x}_k \qquad (6b) \\
\boldsymbol{\lambda}_{k+1} = W\boldsymbol{\lambda}_k & + c\,\boldsymbol{d}_{k+1}, \qquad (6c)
\end{aligned}$$

where $f(\boldsymbol{x}) = \sum_{i=1}^N f_i(x_i)$, $A_{\mathrm{d}} = \operatorname{blkdiag}(A_1, \ldots, A_N)$, $X = X_1 \times \cdots \times X_N$, and $W = \mathcal{W} \otimes I_p$. Note that, in (6), we used $\boldsymbol{\delta}_k = W\boldsymbol{d}_k$ and $\boldsymbol{\ell}_k = W\boldsymbol{\lambda}_k$, which represent the network-wide formulations of Steps 6 and 7 of Algorithm 1, respectively.

As can be seen from (6), Tracking-ADMM describes a dynamical system composed of two parts: a nonlinear dynamics given by the optimization step (cf. (6a)) and a linear dynamics given by a twofold consensus-based step (cf. (6b) and (6c)). In the forthcoming analysis, we will investigate the structural properties of these two parts separately. Then we show how to combine them to prove convergence of the proposed algorithm to an optimal solution of the original problem $\mathcal{P}$.

3.3.2. Discussion on Assumption 5

Before starting with the convergence analysis, we now show that Assumption 5 is not restrictive.

Suppose to substitute Steps 6 and 7 with

$$\delta'_{i,k} = \textstyle\sum_{j \in \mathcal{N}_i} w_{ij}\, d_{j,k} \quad \text{and} \quad \ell'_{i,k} = \textstyle\sum_{j \in \mathcal{N}_i} w_{ij}\, \lambda_{j,k}$$

and then perform the updates

$$\delta_{i,k} = \textstyle\sum_{j \in \mathcal{N}_i} w_{ij}\, \delta'_{j,k} \quad \text{and} \quad \ell_{i,k} = \textstyle\sum_{j \in \mathcal{N}_i} w_{ij}\, \ell'_{j,k}$$

before Step 8. These two steps can then be compactly written as

$$\begin{aligned}
\boldsymbol{\delta}_k &= W^2\boldsymbol{d}_k = (\mathcal{W} \otimes I_p)^2\boldsymbol{d}_k = (\mathcal{W}^2 \otimes I_p)\boldsymbol{d}_k, \\
\boldsymbol{\ell}_k &= W^2\boldsymbol{\lambda}_k = (\mathcal{W} \otimes I_p)^2\boldsymbol{\lambda}_k = (\mathcal{W}^2 \otimes I_p)\boldsymbol{\lambda}_k,
\end{aligned}$$

which are equivalent to a single update round with $\mathcal{W}^2$ in place of $\mathcal{W}$. Since, under Assumption 4, $\mathcal{W}$ is symmetric and doubly stochastic, then $\mathcal{W}^2$ is symmetric, doubly stochastic, and positive semidefinite, thus satisfying Assumption 5. Letting $\mathcal{E}^2$ be the set of edges (i,j) such that there exists a $\kappa \in \mathcal{V}$ for which $(i,\kappa),(\kappa,j) \in \mathcal{E}$, it is easy to show that the graph $\mathcal{G}^2 = (\mathcal{V},\mathcal{E}^2)$ satisfies Assumption 3 and $\mathcal{W}^2$ satisfies Assumption 4 with $\mathcal{E}^2$ in place of $\mathcal{E}$.

Therefore, Assumption 5 can be easily satisfied by simply performing two update rounds rather than one. We choose not to alter Algorithm 1 to keep the notation light and avoid writing $\mathcal{W}^2$ in place of $\mathcal{W}$ everywhere.

3.3.3. Network Average Quantities

We start the analysis of Tracking-ADMM by stating some important properties regarding the network averages of the local variables $d_{i,k}$ and $\lambda_{i,k}$, which are defined as

$$\bar{d}_k = \frac{1}{N}\sum_{i=1}^{N} d_{i,k} \quad \text{and} \quad \bar{\lambda}_k = \frac{1}{N}\sum_{i=1}^{N} \lambda_{i,k}, \tag{7}$$

respectively. The following lemmas highlight key features of $\bar{d}_k$ and $\bar{\lambda}_k$.

Lemma 1 (Tracking property)**.** *Under Assumption 4 it holds that*

$$\bar{d}_k = \frac{1}{N}\left(\sum_{i=1}^{N} A_i x_{i,k} - b\right), \tag{8}$$

for all $k \geq 0$. □

Lemma 2 (Average dual update)**.** *Under Assumption 4, it holds that*

$$\bar{\lambda}_{k+1} = \bar{\lambda}_k + c\,\bar{d}_{k+1}, \tag{9}$$

for all $k \geq 0$. □

As can be seen by a formal comparison, (8) and (9) mimic the update step (3b) and (3c) of the parallel ADMM. Therefore, by virtue of Lemmas 1 and 2, running Algorithm 1 results in the network behaving, on average, like the parallel ADMM.

3.3.4. Consensus Errors as Linear Dynamical Systems

The introduction of $\bar{d}_k$ and $\bar{\lambda}_k$ is also useful to analyze the behavior of $d_{i,k}$ and $\lambda_{i,k}$. Specifically, as usually done in consensus-based optimization algorithms, we shall study the consensus error, i.e., the distance between the agents' local quantities $d_{i,k}$ and $\lambda_{i,k}$ and their respective network average $\bar{d}_k$ and $\bar{\lambda}_k$.

To ease the notation, let us introduce the following vectors stacking N copies of the respective average quantities

$$\begin{aligned}\bar{\boldsymbol{d}}_k &= \mathbb{1}_N \otimes \bar{d}_k,\\ \bar{\boldsymbol{\lambda}}_k &= \mathbb{1}_N \otimes \bar{\lambda}_k,\end{aligned}$$

which are useful when comparing the aggregate vectors $\boldsymbol{d}_k$ and $\boldsymbol{\lambda}_k$ with their averages $\bar{d}_k$ and $\bar{\lambda}_k$. Then, define the consensus errors as

$$\begin{aligned}\boldsymbol{e}_k^{\boldsymbol{d}} &= \boldsymbol{d}_k - \bar{\boldsymbol{d}}_k,\\ \boldsymbol{e}_k^{\boldsymbol{\lambda}} &= \boldsymbol{\lambda}_k - \bar{\boldsymbol{\lambda}}_k,\end{aligned}$$

and the auxiliary sequence

$$\boldsymbol{z}_k = A_{\rm d}\boldsymbol{x}_k - \bar{\boldsymbol{d}}_k. \tag{10}$$

From the expression of $\boldsymbol{d}_{k+1}$ in (6b), we can write

$$\begin{aligned}\boldsymbol{e}_{k+1}^{\boldsymbol{d}} &= \boldsymbol{d}_{k+1} - \bar{\boldsymbol{d}}_{k+1}\\ &= W\boldsymbol{d}_k + A_{\rm d}\boldsymbol{x}_{k+1} - A_{\rm d}\boldsymbol{x}_k - \bar{\boldsymbol{d}}_{k+1} \pm \bar{\boldsymbol{d}}_k\\ &\overset{\text{(a)}}{=} W\boldsymbol{d}_k - \bar{\boldsymbol{d}}_k + [\boldsymbol{z}_{k+1} - \boldsymbol{z}_k]\\ &\overset{\text{(b)}}{=} W\boldsymbol{e}_k^{\boldsymbol{d}} + [\boldsymbol{z}_{k+1} - \boldsymbol{z}_k],\end{aligned} \tag{11}$$

where in (a) we used the definition of $\boldsymbol{z}_{k+1}$ and $\boldsymbol{z}_k$ (cf. (10)) and in (b) we exploited the following

$$\begin{aligned}W\bar{\boldsymbol{d}}_k &= (\mathcal{W} \otimes I_p)(\mathbb{1}_N \otimes \bar{d}_k)\\ &\overset{\text{(c)}}{=} (\mathcal{W}\mathbb{1}_N \otimes I_p\bar{d}_k)\\ &\overset{\text{(d)}}{=} (\mathbb{1}_N \otimes \bar{d}_k) = \bar{\boldsymbol{d}}_k,\end{aligned}$$

which holds due to the mixed-product property of the Kronecker product in (c) and the doubly stochasticity of $\mathcal{W}$ under Assumption 4 in (d).

Finally, letting $\mathcal{O}$ be the $N \times N$ matrix with all entries equal to $1/N$ and noticing that

$$\begin{aligned}(\mathcal{O} \otimes I_p)\boldsymbol{e}_k^{\boldsymbol{d}} &= \left(\frac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top \otimes I_p\right)[\boldsymbol{d}_k - \bar{\boldsymbol{d}}_k]\\ &= \mathbb{1}_N \otimes \frac{1}{N}\sum_{i=1}^{N}[d_{i,k} - \bar{d}_k]\\ &= \mathbb{1}_N \otimes \frac{1}{N}[N\bar{d}_k - N\bar{d}_k] = 0,\end{aligned}$$

we can subtract the quantity $(\mathcal{O}\otimes I_p)\boldsymbol{e}_k^{\boldsymbol{d}} = 0$ from the right hand side of (11) to get

$$\boldsymbol{e}_{k+1}^{\boldsymbol{d}} = \tilde{W}\boldsymbol{e}_k^{\boldsymbol{d}} + [\boldsymbol{z}_{k+1} - \boldsymbol{z}_k], \tag{12}$$

where we set $\tilde{W} = W - (\mathcal{O} \otimes I_p)$.

With an analogous reasoning, we can also show that

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

where, in (a), we used (6c) and (9) pre-multiplied by $\mathbb{1}_N\otimes$.

Under Assumptions 3 and 4, as a consequence of the Perron-Frobenius theorem, we have that $\|\mathcal{W} - \mathcal{O}\| < 1$, and thus

$$\rho(\tilde{W}) \leq \|\tilde{W}\| = \|(\mathcal{W} - \mathcal{O}) \otimes I_p\| = \|\mathcal{W} - \mathcal{O}\|\|I_p\| < 1.$$

Thus, all eigenvalues of $\tilde{W}$ lie within the unit circle. This implies that the dynamical systems in (12) and (13), describing the evolution of the consensus errors, are asymptotically stable. The importance of this fact is twofold: i) (12) and (13) enjoys the input-to-state stability property: if the sequence $\{\boldsymbol{z}_{k+1} - \boldsymbol{z}_k\}_{k\geq 0}$ is bounded then also $\{\boldsymbol{e}_k^{\boldsymbol{d}}\}_{k\geq 0}$ and $\{\boldsymbol{e}_k^{\boldsymbol{\lambda}}\}_{k\geq 0}$ are bounded; ii) if $\boldsymbol{z}_{k+1} - \boldsymbol{z}_k$ vanishes, then also $\boldsymbol{e}_k^{\boldsymbol{d}}$ and $\boldsymbol{e}_k^{\boldsymbol{\lambda}}$ do.

Point i) is formalized in the following lemma.

Lemma 3 (Bounded sequences)**.** *Under Assumptions 1, 3 and 4 we have that*

(i) the sequences $\{\boldsymbol{x}_k\}_{k\geq 0}$ and $\{\boldsymbol{z}_k\}_{k\geq 0}$ are bounded;

(ii) the sequences $\{\boldsymbol{e}_k^{\boldsymbol{d}}\}_{k\geq 0}$ and $\{\boldsymbol{e}_k^{\boldsymbol{\lambda}}\}_{k\geq 0}$ are bounded. □

The results in Lemmas 1, 2, and 3 can be interpreted as follows. The sequences $\{\boldsymbol{d}_k\}_{k\geq 0}$ and $\{\boldsymbol{\lambda}_k\}_{k\geq 0}$ generated by (6b) and (6c) evolve in a (possibly large) neighborhood of their averages $\bar{d}_k$ and $\bar{\lambda}_k$, which, in turn, evolve according to a parallel ADMM iteration.

3.3.5. Optimality Characterization

We now focus our attention on the analysis of the (nonlinear) update step in (6a). The following result represents the distributed counterpart of the key inequality leveraged in the proof of the parallel ADMM.

Proposition 1 (Local optimality)**.** *Under Assumptions 1, 2, and 4 we have that*

$$\begin{aligned} &\|\bar{\boldsymbol{\lambda}}_{k+1} - \boldsymbol{\lambda}^\star\|^2 + 2c[\boldsymbol{z}_{k+1} - A_\mathrm{d}\boldsymbol{x}^\star]^\top \boldsymbol{e}_{k+1}^{\boldsymbol{\lambda}} \\ &\qquad \leq \|\bar{\boldsymbol{\lambda}}_k - \boldsymbol{\lambda}^\star\|^2 - \|\bar{\boldsymbol{\lambda}}_{k+1} - \bar{\boldsymbol{\lambda}}_k\|^2, \end{aligned} \tag{14}$$

for any optimal solution pair $(\boldsymbol{x}^\star, \lambda^\star)$ for $\mathcal{P}$ and $\mathcal{D}$, where we set $\boldsymbol{\lambda}^\star = \mathbb{1}_N \otimes \lambda^\star$. □

If we ignore the cross term $2c[\boldsymbol{z}_{k+1} - A_\mathrm{d}\boldsymbol{x}^\star]^\top \boldsymbol{e}_{k+1}^{\boldsymbol{\lambda}}$, then (14) tells us that as long as $\bar{\boldsymbol{\lambda}}_k$ changes from one iteration to the next one, the local dual estimates are, on average, getting closer to the optimal solution of $\mathcal{D}$. The additional cross term depends on the consensus error $\boldsymbol{e}_{k+1}^{\boldsymbol{\lambda}} = \boldsymbol{\lambda}_{k+1} - \bar{\boldsymbol{\lambda}}_{k+1}$ and is clearly a distinctive feature of Tracking-ADMM (as opposed to the parallel ADMM) due to its distributed nature. In the parallel ADMM, a condition similar to (14) can be derived and leveraged to prove the algorithm convergence. However, as expected, this is not the case here: (14) is not sufficient to prove convergence of the proposed scheme and we are required to study the interplay between the minimization step in (6a) and the two consensus steps in (6b) and (6c).

3.3.6. Tracking-ADMM Convergence

The convergence of Algorithm 1 is divided into two results. The first theorem proves that the two consensus mechanism are successful, i.e., agents eventually agree on a common value for the tracker and for the dual variables (cf. points (i) and (ii)), and the algorithm converges to a solution that is feasible for $\mathcal{P}$ (cf. point (iii)); the second theorem proves that the primal and dual solution obtained by Algorithm 1 are optimal for $\mathcal{P}$ and $\mathcal{D}$, respectively.

Theorem 1 (Convergence)**.** *Under Assumptions 1-5, the sequences generated by Tracking-ADMM satisfy:*

(i) $\lim_{k\to\infty} \|\boldsymbol{e}_k^{\boldsymbol{d}}\| = 0,$

(ii) $\lim_{k\to\infty} \|\boldsymbol{e}_k^{\boldsymbol{\lambda}}\| = 0,$

(iii) $\lim_{k\to\infty} \|\bar{d}_k\| = 0,$

(iv) $\{\|\bar{\boldsymbol{\lambda}}_k - \boldsymbol{\lambda}^\star\|^2 + c^2\|\boldsymbol{z}_k - A_\mathrm{d}\boldsymbol{x}^\star\|^2\}_{k\geq 0}$ *is convergent,*

for any optimal solution pair $(\boldsymbol{x}^\star, \lambda^\star)$ for $\mathcal{P}$ and $\mathcal{D}$, with $\boldsymbol{\lambda}^\star = \mathbb{1}_N \otimes \lambda^\star$. □

The main idea behind the proof of Theorem 1 is the following. We start by considering (12) and (13) together as a single dynamical system for the consensus updates (the consensus system). Since the consensus system is asymptotically stable, then we can build a positive-definite quadratic Lyapunov function which monotonically decreases along its free evolution. However, the consensus system is *not* autonomous, thus the inequality describing the variation of such Lyapunov function across iterations contains terms that are not defined in sign, so that it is no longer necessarily decreasing. The idea is to properly combine such inequality with (14) of Proposition 1 so as to "balance out" the terms that are not defined in sign. In this way, we obtain an aggregate descent condition that allows us to prove the asymptotic stability of the overall (nonlinear) dynamical system modeling our Tracking-ADMM algorithm.

The reader should note that point (i) of Theorem 1 implies that, for all $i = 1, \ldots, N$, $d_{i,k}$ tends to $\bar{d}_k$ as $k \to \infty$ which, together with point (iii), yields $\|d_{i,k}\| \to 0$ as $k \to \infty$, for all $i = 1, \ldots, N$. In light of (8), this ensures that each agent is able to locally assess the amount of infeasibility of the current primal iterates for the coupling constraint. This feature is of great importance in those applications where feasibility up to a given tolerance is sufficient.

Finally, the following result shows that by running Algorithm 1 agents are able to compute optimal solutions to $\mathcal{P}$ and $\mathcal{D}$ in a fully distributed way.

Theorem 2 (Optimality)**.** *Under Assumptions 1-5, the sequences generated by Tracking-ADMM are such that*

(i) any limit point of the primal sequence $\{\boldsymbol{x}_k\}_{k\geq 0}$ is an optimal solution $\boldsymbol{x}^\star$ of $\mathcal{P}$;

(ii) each dual sequence $\{\lambda_{i,k}\}_{k\geq 0}$, $i = 1,\dots,N$, *converges to the same optimal solution* $\lambda^\star$ *of* $\mathcal{D}$. □

We point out that Theorem 2 guarantees that the sequences $\{x_{i,k}\}_{k\geq 0}$ are asymptotically optimal, hence feasible for both local and the coupling constraints. Notice also that Algorithm 1 works with (not necessarily strictly) convex primal cost functions and does not require any primal recovery procedure.

4. Numerical Study

To demonstrate the effectiveness of our approach, we test Tracking-ADMM on a modified version of the Plug-in Electric Vehicles Optimal Charging Schedule problem described in Vujanic et al. (2016). A fleet of N electric vehicles has to select an overnight charging schedule so as to minimize the sum of the electricity costs for charging the internal battery of each vehicle while complying with vehicle-level constraints (user requirements on the final state of charge and battery physical limitations) and a network-wide constraint (maximum power that the grid can deliver). For simplicity we focus on the "only charging" case of the charging problem in Vujanic et al. (2016). Differently from Vujanic et al. (2016), where the vehicles can decide only whether to charge or not their internal battery at a fixed charging rate, here we allow each vehicle to optimize also the charging rate. The resulting optimization program can then be formalized as follows

$$\begin{aligned} \min_{\xi_1,\dots,\xi_N} \quad & \sum_{i=1}^{N} \gamma_i^\top \xi_i \\ \text{subject to:} \quad & \sum_{i=1}^{N} \tilde{A}_i \xi_i \leq b \\ & \xi_i \in \Xi_i \qquad i = 1,\dots,N, \end{aligned} \tag{15}$$

where X_i are bounded polyhedral sets. We refer the reader to Vujanic et al. (2016) for the precise formulations of the quantities in (15).

The coupling constraints in (15) is an inequality, which makes (15) not fitting the formulation in $\mathcal{P}$. However, we can easily turn the coupling constraint into an equality by introducing p additional slack variables for each vehicle. Problem (15) thus becomes

$$\begin{aligned} \min_{\substack{\xi_1,\dots,\xi_N,\\ s_1,\dots,s_N}} \quad & \sum_{i=1}^{N} \gamma_i^\top \xi_i \\ \text{subject to:} \quad & \sum_{i=1}^{N} (\tilde{A}_i \xi_i + s_i) = b \\ & \left.\begin{aligned} & \xi_i \in \Xi_i \\ & 0 \leq s_i \leq \bar{s} \end{aligned}\right. \quad i = 1,\dots,N, \end{aligned} \tag{16}$$

which now fits $\mathcal{P}$ with the identifications $x_i = [\xi_i^\top \; s_i^\top]^\top$, $f_i(x_i) = \gamma_i^\top \xi_i$, $A_i = [\tilde{A}_i \; I_p]$, and $X_i = \Xi_i \times S_i$ where $S_i = \{s_i : 0 \leq s_i \leq \bar{s}\}$, for some $\bar{s} \geq b$, for all $i = 1,\dots,N$.

In our simulation we considered $N = 50$ vehicles. Each vehicle i has 24 decision variables ξ_i plus 24 additional slack variables s_i ($n_i = 48$) and the local constraint set X_i is defined by 245 inequalities. The number of coupling constraint, and consequently the number of Lagrange multipliers, is $p = 24$. The communication network and the weight matrix $\mathcal{W}$ have been generated at random so as to satisfy Assumptions 3, 4, and 5.

We run our Tracking-ADMM for 10^4 iterations, initialized as suggested in Section 3.2, using different values of the penalty parameter $c \in \{10^{-6}, 10^{-5.5}, 10^{-5}, 10^{-4.5}, 10^{-4}\}$. Note that Step 8 of Algorithm 1 is solved using the *network optimizer* of IBM ILOG CPLEX 12.9 with the minimum achievable tolerance of 10^{-11} both for optimality and feasibility.

In Figure 1 we report, for different values of c, the behavior across iterations of the relative optimality gap

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

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 (upper plot), and the relative violation

$$\frac{\|\sum_{i=1}^{N} A_i x_{i,k} - b\|}{\|b\|}$$

of the joint constraints (lower plot). As it can be seen from the picture, the proposed algorithm achieves feasibility and optimality in all cases, albeit the actual value of the penalty parameter affect the convergence speed. Note that, differently from the approaches based on dual decomposition, the primal iterates generated by Algorithm 1 achieve feasibility of the coupling constraint without requiring any primal recovery procedure (which is known to deteriorate the convergence speed). Similarly to the parallel ADMM, Figure 1 shows that there is a range of values for the penalty parameter c which result in a faster convergence rate, while outside the range the speed may be slower. However, we shall emphasize that tolerance thresholds usually employed in practice (e.g., 10^{-6}) are achieved quite early for most values of c.

In Figure 2 we also plot the evolution of the consensus errors $\boldsymbol{e}_k^{\boldsymbol{\lambda}}$ (upper plot) and $\boldsymbol{e}_k^{\boldsymbol{d}}$ (lower plot), for different values of c (different colors). From the picture the reader can easily see how the errors asymptotically vanish, in accordance to Theorem 1.

For comparison purposes we also run (Chang, 2016, Algorithm 1) on the same problem. In Figure 3 we report the behavior across iterations of the relative optimality gap (upper plot) and the relative violation of the joint constraint (lower plot) for different values of the penalty parameter of (Chang, 2016, Algorithm 1), denoted here as c'. For a fair comparison we show the runs of (Chang, 2016, Algorithm 1) associated to those values of

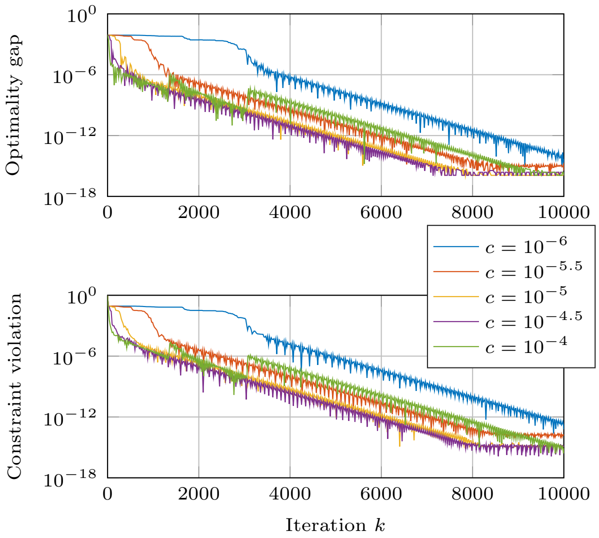

Figure 1: Relative difference between the cost achieved by the primal tentative solution and the optimal cost (upper plot) and relative violation of the joint constraints (lower plot), across iterations.

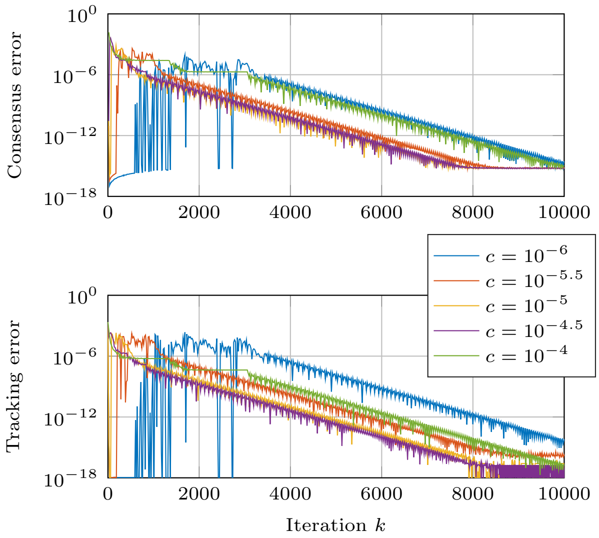

Figure 2: Evolution of the consensus errors $\boldsymbol{e}_k^{\boldsymbol{\lambda}}$ (upper plot) and $\boldsymbol{e}_k^{\boldsymbol{d}}$ (lower plot), across iterations.

the penalty parameter that achieve the best performance: $c' \in \{10^2, 10^{2.5}, 10^3, 10^{3.5}, 10^4\}$. By comparing Figures 1 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 Figure 1, which are steeper than those in Figure 3, irrespective of the value of the penalty coefficient. To ease the comparison, we also report in Figure 3 the best run of Tracking-ADMM ($c = 10^{-4.5}$).

5. Conclusions

In this paper we have proposed a novel distributed method to solve constraint-coupled convex optimization problems in which the sum of local cost functions needs to be minimized while satisfying both individual constraints (involving one component of the decision variables) and a common linear coupling constraint (involving all the components). The distributed algorithm combines the (parallel) ADMM algorithm tailored for this class of optimization problems with a dynamic tracking mechanism. We proved that each agent asymptotically computes an optimal dual solution and its portion of an optimal solution to the target (primal) problem. Moreover, the tracking scheme allows agents to obtain a local measure of the coupling-constraint violation. Numerical computations corroborated the theoretical results.

References

Alghunaim, S., Yuan, K., Sayed, A., 2018. Dual coupled diffusion for distributed optimization with affine constraints, in: IEEE Conference on Decision and Control (CDC), pp. 829–834.

Bertsekas, D.P., Tsitsiklis, J.N., 1989. Parallel and distributed computation: numerical methods. Prentice hall Englewood Cliffs, NJ.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundation and Trends® in Machine learning 3, 1–122.

Carli, R., Dotoli, M., 2019. Distributed alternating direction method of multipliers for linearly constrained optimization over a network. IEEE Control Systems Letters 4, 247–252.

Chang, T.H., 2016. A proximal dual consensus ADMM method for multi-agent constrained optimization. IEEE Transactions on Signal Processing 64, 3719–3734.

Chang, T.H., Nedić, A., Scaglione, A., 2014. Distributed constrained optimization by consensus-based primal-dual perturbation method. IEEE Transactions on Automatic Control 59, 1524–1538.

Chang, X., Liu, S., Zhao, P., Li, X., 2018. Convergent prediction–correction-based ADMM for multi-block separable convex programming. Journal of Computational and Applied Mathematics 335, 270–288.

Chen, G., Yang, Q., 2019. Distributed constrained optimization for multi-agent networks with nonsmooth objective functions. Systems & Control Letters 124, 60–67.

Di Lorenzo, P., Scutari, G., 2016. NEXT: In-network nonconvex optimization. IEEE Transactions on Signal and Information Process. over Networks 2, 120–136.

Duchi, J.C., Agarwal, A., Wainwright, M.J., 2012. Dual averaging for distributed optimization: Convergence analysis and network scaling. IEEE Transactions on Automatic Control 57, 592–606.

Falsone, A., Margellos, K., Garatti, S., Prandini, M., 2017. Dual decomposition for multi-agent distributed optimization with coupling constraints. Automatica 84, 149–158.

He, B., Hou, L., Yuan, X., 2015. On full Jacobian decomposition of the augmented Lagrangian method for separable convex programming. SIAM Journal on Optimization 25, 2274–2312.

Iutzeler, F., Bianchi, P., Ciblat, P., Hachem, W., 2016. Explicit convergence rate of a distributed alternating direction method of multipliers. IEEE Transactions on Automatic Control 61, 892–904.

Jakovetić, D., Moura, J.M., Xavier, J.M., 2015. Linear convergence rate of a class of distributed augmented Lagrangian algorithms. IEEE Transactions on Automatic Control 60, 922–936.

Jakovetić, D., Xavier, J., Moura, J.M., 2014. Fast distributed gradient methods. IEEE Transactions on Automatic Control 59, 1131–1146.

Johansson, B., Keviczky, T., Johansson, M., Johansson, K.H., 2008.

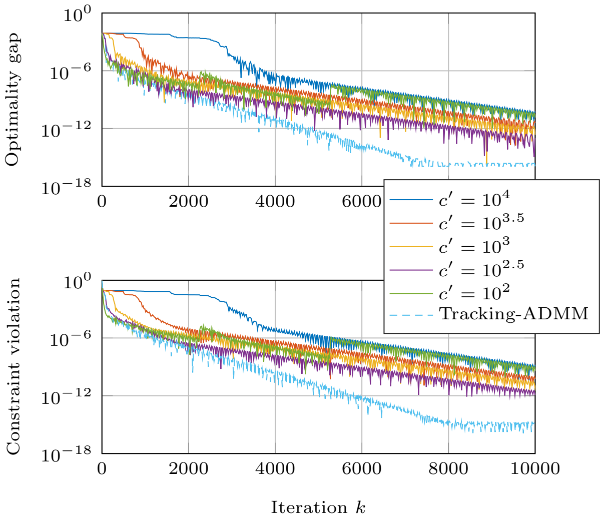

Figure 3: Relative difference between the cost achieved by the primal tentative solution and the optimal cost (upper plot) and relative violation of the joint constraints (lower plot), across iterations of (Chang, 2016, Algorithm 1) (solid lines). For comparison purposes, we also report the behavior of our Tracking-ADMM with $c = 10^{-4.5}$ (dashed line).

Subgradient methods and consensus algorithms for solving convex optimization problems, in: IEEE Conference on Decision and Control (CDC), pp. 4185–4190.

Kia, S.S., 2017. Distributed optimal in-network resource allocation algorithm design via a control theoretic approach. Systems & Control Letters 107, 49–57.

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

Liang, S., Wang, L.Y., Yin, G., 2019. Distributed smooth convex optimization with coupled constraints. IEEE Transactions on Automatic Control .

Ling, Q., Ribeiro, A., 2014. Decentralized dynamic optimization through the alternating direction method of multipliers. IEEE Transactions on Signal Processing 5, 1185–1197.

Makhdoumi, A., Ozdaglar, A., 2017. Convergence rate of distributed ADMM over networks. IEEE Transactions on Automatic Control 62, 5082–5095.

Margellos, K., Falsone, A., Garatti, S., Prandini, M., 2018. Distributed constrained optimization and consensus in uncertain networks via proximal minimization. IEEE Transactions on Automatic Control 63, 1372–1387.

Mateos-Núnez, D., Cortés, J., 2017. Distributed saddle-point subgradient algorithms with Laplacian averaging. IEEE Transactions on Automatic Control 62, 2720–2735.

Mota, J.F., Xavier, J.M., Aguiar, P.M., Püschel, M., 2013. D-ADMM: A communication-efficient distributed algorithm for separable optimization. IEEE Transactions on Signal Processing 61, 2718–2723.

Necoara, I., Nedelcu, V., 2015. On linear convergence of a distributed dual gradient algorithm for linearly constrained separable convex problems. Automatica 55, 209–216.

Necoara, I., Suykens, J., 2009. Interior-point Lagrangian decomposition method for separable convex optimization. Journal of Optimization Theory and Applications 143, 567.

Nedić, A., Olshevsky, A., 2015. Distributed optimization over time-varying directed graphs. IEEE Transactions on Automatic Control 60, 601–615.

Nedić, A., Olshevsky, A., Shi, W., 2017. Achieving geometric convergence for distributed optimization over time-varying graphs. SIAM J. on Optimization 27, 2597–2633.

Nedić, A., Ozdaglar, A., 2009. Distributed subgradient methods for multi-agent optimization. IEEE Transactions on Automatic Control 54, 48–61.

Nedić, A., Ozdaglar, A., Parrilo, P.A., 2010. Constrained consensus and optimization in multi-agent networks. IEEE Transactions on Automatic Control 55.

Notarnicola, I., Notarstefano, G., 2019. Constraint-coupled distributed optimization: a relaxation and duality approach. IEEE Transactions on Control of Network Systems , 1–10.

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

Sherson, T.W., Heusdens, R., Kleijn, W.B., 2019. On the distributed method of multipliers for separable convex optimization problems. IEEE Transactions on Signal and Information Processing over Networks 5, 495–510.

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

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

Simonetto, A., Jamali-Rad, H., 2016. Primal recovery from consensus-based dual decomposition for distributed convex optimization. Journal of Optimization Theory and Applications 168, 172–197.

Varagnolo, D., Zanella, F., Cenedese, A., Pillonetto, G., Schenato, L., 2016. Newton-Raphson consensus for distributed convex optimization. IEEE Transactions on Automatic Control 61, 994–1009.

Vujanic, R., Esfahani, P.M., Goulart, P.J., Mariéthoz, S., Morari, M., 2016. A decomposition method for large scale MILPs, with performance guarantees and a power system application. Automatica 67, 144–156.

Wang, Z., Ong, C.J., 2017. Distributed model predictive control of linear discrete-time systems with local and global constraints. Automatica 81, 184–195.

Xi, C., Xin, R., Khan, U.A., 2018. ADD-OPT: Accelerated distributed directed optimization. IEEE Transactions on Automatic Control 63, 1329–1339.

Xu, J., Zhu, S., Soh, Y.C., Xie, L., 2018. Convergence of asynchronous distributed gradient methods over stochastic networks. IEEE Transactions on Automatic Control 63, 434–448.

Zanella, F., Varagnolo, D., Cenedese, A., Pillonetto, G., Schenato, L., 2011. Newton-Raphson consensus for distributed convex optimization, in: IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), pp. 5917–5922.

Zhang, Y., Zavlanos, M.M., 2018. A consensus-based distributed augmented Lagrangian method, in: IEEE Conference on Decision and Control (CDC), pp. 1763–1768.

Zhu, M., Martínez, S., 2010. Discrete-time dynamic average consensus. Automatica 46, 322–329.

Zhu, M., Martínez, S., 2012. On distributed convex optimization under inequality and equality constraints. IEEE Transactions on Automatic Control 57, 151–164.

Appendix A. Proofs

Proof of Lemma 1 (Tracking property)

We prove (8) by induction. For $k = 0$, given the initialization in Step 3, we have that

$$\bar{d}_0 = \frac{1}{N}\sum_{i=1}^{N} d_{i,0} = \frac{1}{N}\sum_{i=1}^{N} (A_i x_{i,0} - b_i)$$

$$= \frac{1}{N}\left(\sum_{i=1}^{N} A_i x_{i,0} - b\right)$$

which proves that (8) holds for $k = 0$. Assume now that (8) holds up to k. If we can prove that (8) holds also for $k + 1$, then the proof is completed. Indeed,

$$\begin{aligned}
\bar{d}_{k+1} &= \frac{1}{N}\sum_{i=1}^{N} d_{i,k+1}\\
&\overset{\text{(a)}}{=} \frac{1}{N}\sum_{i=1}^{N}\Big(\sum_{j=1}^{N} w_{ij} d_{j,k} + A_i x_{i,k+1} - A_i x_{i,k}\Big)\\
&\overset{\text{(b)}}{=} \frac{1}{N}\sum_{j=1}^{N}\Big(\sum_{i=1}^{N} w_{ij}\Big) d_{j,k} + \frac{1}{N}\sum_{i=1}^{N}(A_i x_{i,k+1} - A_i x_{i,k})\\
&\overset{\text{(c)}}{=} \bar{d}_k + \frac{1}{N}\sum_{i=1}^{N}(A_i x_{i,k+1} - A_i x_{i,k})\\
&\overset{\text{(d)}}{=} \frac{1}{N}\Big(\sum_{i=1}^{N} A_i x_{i,k+1} - b\Big),
\end{aligned}$$

where in (a) we used Steps 9 and 6 in (b) we exchanged the summations and we used column stochasticity of the weights (cf. Assumption 4), in (c) we used the definition of $\bar{d}_k$ and the induction step in (d). □

Proof of Lemma 2 (Average dual update)

By definition of $\bar{\lambda}_{k+1}$, we can write

$$\bar{\lambda}_{k+1} = \frac{1}{N}\sum_{i=1}^{N}\lambda_{i,k+1}$$

which can be further elaborated as follows

$$\begin{aligned}
\bar{\lambda}_{k+1} &\overset{\text{(a)}}{=} \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{j=1}^{N} w_{ij}\lambda_{j,k} + c\, d_{i,k+1}\right)\\
&\overset{\text{(b)}}{=} \frac{1}{N}\sum_{j=1}^{N}\left(\sum_{i=1}^{N} w_{ij}\right)\lambda_{j,k} + c\frac{1}{N}\sum_{i=1}^{N} d_{i,k+1}\\
&\overset{\text{(c)}}{=} \bar{\lambda}_k + c\,\bar{d}_{k+1},
\end{aligned}$$

where in (a) we used the definition of $\lambda_{i,k+1}$ in Step 10, in (b) we switched the summations, and in (c) we used column stochasticity of the weights (cf. Assumption 4) together with the definition of $\bar{\lambda}_k$ and $\bar{d}_{k+1}$. □

Proof of Lemma 3 (Bounded sequences)

By (6a) we have that $\boldsymbol{x}_k \in X$ for all k and, under Assumption 1, X is bounded. Therefore the sequence $\{\boldsymbol{x}_k\}_{k\geq 0}$ is bounded. Under Assumption 4, from (8) in Lemma 1 we have that $\bar{d}_k = (1/N)(A\boldsymbol{x}_k - b)$ and, therefore, also $\{\bar{\boldsymbol{d}}_k\}_{k\geq 0}$ is bounded since $\bar{\boldsymbol{d}}_k = \mathbb{1}_N \otimes \bar{d}_k$. Moreover, since $\boldsymbol{z}_k = A_{\mathrm{d}}\boldsymbol{x}_k - \bar{\boldsymbol{d}}_k$, then also $\{\boldsymbol{z}_k\}_{k\geq 0}$ is bounded. This proves point (i).

Being the sequence $\{\boldsymbol{z}_k\}_{k\geq 0}$ bounded, we have that $\{\boldsymbol{z}_{k+1} - \boldsymbol{z}_k\}_{k\geq 0}$ is also bounded. Therefore, exploiting the input-to-state stability property of the dynamical system in (12) under Assumptions 3 and 4, we can conclude that $\{\boldsymbol{e}_k^{\boldsymbol{d}}\}_{k\geq 0}$ is a bounded sequence. Finally, owing to the input-to-state stability of the dynamical system in (13) under Assumptions 3 and 4 together with boundedness of $\{\boldsymbol{e}_k^{\boldsymbol{d}}\}_{k\geq 0}$, we also have that $\{\boldsymbol{e}_k^{\boldsymbol{\lambda}}\}_{k\geq 0}$ is bounded, thus proving point (ii) and concluding the proof. □

Proof of Proposition 1 (Local optimality)

By (Bertsekas and Tsitsiklis, 1989, Lemma 4.1) applied to (6a), we have that $\boldsymbol{x}_{k+1}$ satisfies

$$\begin{aligned}
f(\boldsymbol{x}_{k+1}) &+ [W\boldsymbol{\lambda}_k + c(A_{\mathrm{d}}\boldsymbol{x}_{k+1} - A_{\mathrm{d}}\boldsymbol{x}_k + W\boldsymbol{d}_k)]^\top A_{\mathrm{d}}\boldsymbol{x}_{k+1}\\
&\leq f(\boldsymbol{x}) + [W\boldsymbol{\lambda}_k + c(A_{\mathrm{d}}\boldsymbol{x}_{k+1} - A_{\mathrm{d}}\boldsymbol{x}_k + W\boldsymbol{d}_k)]^\top A_{\mathrm{d}}\boldsymbol{x},
\end{aligned}$$

for any $\boldsymbol{x} \in X$. Recalling that $\boldsymbol{d}_{k+1} = W\boldsymbol{d}_k + A_{\mathrm{d}}\boldsymbol{x}_{k+1} - A_{\mathrm{d}}\boldsymbol{x}_k$ and $\boldsymbol{\lambda}_{k+1} = W\boldsymbol{\lambda}_k + c\,\boldsymbol{d}_{k+1}$ (cf. (6b) and (6c), respectively), we have

$$f(\boldsymbol{x}_{k+1}) + \boldsymbol{\lambda}_{k+1}^\top A_{\mathrm{d}}\boldsymbol{x}_{k+1} \leq f(\boldsymbol{x}) + \boldsymbol{\lambda}_{k+1}^\top A_{\mathrm{d}}\boldsymbol{x},$$

for any $\boldsymbol{x} \in X$. Under Assumption 2, we can set $\boldsymbol{x}$ equal to some optimal solution $\boldsymbol{x}^\star$ of $\mathcal{P}$, to obtain

$$f(\boldsymbol{x}_{k+1}) + \boldsymbol{\lambda}_{k+1}^\top[A_{\mathrm{d}}\boldsymbol{x}_{k+1} - A_{\mathrm{d}}\boldsymbol{x}^\star] \leq f(\boldsymbol{x}^\star).$$

Under Assumption 2, by the Saddle Point Theorem in (Bertsekas and Tsitsiklis, 1989, pag. 665) we have that

$$L(\boldsymbol{x}^\star, \lambda) \leq L(\boldsymbol{x}^\star, \lambda^\star) \leq L(\boldsymbol{x}, \lambda^\star), \tag{A.1}$$

for all $\boldsymbol{x} \in X$ and for all λ. Given (A.1) and the fact that $L(\boldsymbol{x}^\star, \lambda^\star) = f(\boldsymbol{x}^\star)$, we have that

$$\begin{aligned}
f(\boldsymbol{x}_{k+1}) + \boldsymbol{\lambda}_{k+1}^\top[A_{\mathrm{d}}\boldsymbol{x}_{k+1} - A_{\mathrm{d}}\boldsymbol{x}^\star] &\leq f(\boldsymbol{x}^\star)\\
&\leq f(\boldsymbol{x}) + {\lambda^\star}^\top[A\boldsymbol{x} - b],
\end{aligned}$$

for all $\boldsymbol{x} \in X$ and any $\lambda^\star$. Writing $b = A\boldsymbol{x}^\star$ and noticing that ${\lambda^\star}^\top[A\boldsymbol{x} - A\boldsymbol{x}^\star] = {\boldsymbol{\lambda}^\star}^\top[A_{\mathrm{d}}\boldsymbol{x} - A_{\mathrm{d}}\boldsymbol{x}^\star]$, it holds

$$\begin{aligned}
f(\boldsymbol{x}_{k+1}) &+ \boldsymbol{\lambda}_{k+1}^\top[A_{\mathrm{d}}\boldsymbol{x}_{k+1} - A_{\mathrm{d}}\boldsymbol{x}^\star]\\
&\leq f(\boldsymbol{x}) + {\boldsymbol{\lambda}^\star}^\top[A_{\mathrm{d}}\boldsymbol{x} - A_{\mathrm{d}}\boldsymbol{x}^\star].
\end{aligned}$$

Setting $\boldsymbol{x} = \boldsymbol{x}_{k+1}$, simplifying $f(\boldsymbol{x}_{k+1})$, and bringing everything on the left-hand side, we have

$$[\boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}^\star]^\top[A_{\mathrm{d}}\boldsymbol{x}_{k+1} - A_{\mathrm{d}}\boldsymbol{x}^\star] \leq 0.$$

Adding and subtracting $\bar{\boldsymbol{\lambda}}_{k+1}^\top[A_{\mathrm{d}}\boldsymbol{x}_{k+1} - A_{\mathrm{d}}\boldsymbol{x}^\star]$ yields

$$\begin{aligned}
&[\bar{\boldsymbol{\lambda}}_{k+1} - \boldsymbol{\lambda}^\star]^\top[A_{\mathrm{d}}\boldsymbol{x}_{k+1} - A_{\mathrm{d}}\boldsymbol{x}^\star]\\
&\quad + [\boldsymbol{\lambda}_{k+1} - \bar{\boldsymbol{\lambda}}_{k+1}]^\top[A_{\mathrm{d}}\boldsymbol{x}_{k+1} - A_{\mathrm{d}}\boldsymbol{x}^\star] \leq 0.
\end{aligned} \tag{A.2}$$

Focusing on the first term in (A.2), we have

$$[\bar{\boldsymbol{\lambda}}_{k+1} - \boldsymbol{\lambda}^\star]^\top[A_{\mathrm{d}}\boldsymbol{x}_{k+1} - A_{\mathrm{d}}\boldsymbol{x}^\star]$$

$$
\begin{aligned}
&\overset{\text{(a)}}{=} \sum_{i=1}^{N} [\bar\lambda_{k+1} - \lambda^\star]^\top [A_i x_{i,k+1} - A_i x_i^\star] \\
&\overset{\text{(b)}}{=} [\bar\lambda_{k+1} - \lambda^\star]^\top \sum_{i=1}^{N} [A_i x_{i,k+1} - A_i x_i^\star] \\
&\overset{\text{(c)}}{=} [\bar\lambda_{k+1} - \lambda^\star]^\top [A\boldsymbol{x}_{k+1} - A\boldsymbol{x}^\star] \\
&\overset{\text{(d)}}{=} [\bar\lambda_{k+1} - \lambda^\star]^\top [A\boldsymbol{x}_{k+1} - b] \\
&\overset{\text{(e)}}{=} [\bar\lambda_{k+1} - \lambda^\star]^\top N \bar d_{k+1} \\
&\overset{\text{(f)}}{=} \frac{N}{c} [\bar\lambda_{k+1} - \lambda^\star]^\top [\bar\lambda_{k+1} - \bar\lambda_k] \\
&\overset{\text{(g)}}{=} \frac{1}{c} [\bar{\boldsymbol{\lambda}}_{k+1} - \boldsymbol{\lambda}^\star]^\top [\bar{\boldsymbol{\lambda}}_{k+1} - \bar{\boldsymbol{\lambda}}_k], \qquad \text{(A.3)}
\end{aligned}
$$

where (a) is due to the block structure of the vectors involved in the product, (b) is given by the fact that $[\bar\lambda_{k+1} - \lambda^\star]$ does not depend on i, (c) is given by the definition of A, (d) is given by the fact that $A\boldsymbol{x}^\star = b$, (e) is due to (8) in Lemma 1, (f) is given by (9) in Lemma 2, and (g) uses the definitions of $\bar{\boldsymbol{\lambda}}_k$ and $\boldsymbol{\lambda}^\star$. Owing to the fact that

$$
\begin{aligned}
[\boldsymbol{\lambda}_{k+1} - \bar{\boldsymbol{\lambda}}_{k+1}]^\top \bar{\boldsymbol{d}}_{k+1} &= \sum_{i=1}^{N} [\lambda_{i,k+1} - \bar\lambda_{k+1}]^\top \bar d_{k+1} \\
&= \left(\sum_{i=1}^{N} [\lambda_{i,k+1} - \bar\lambda_{k+1}] \right)^\top \bar d_{k+1} \\
&= [N\bar\lambda_{k+1} - N\bar\lambda_{k+1}]^\top \bar d_{k+1} \\
&= 0
\end{aligned}
$$

and considering the second term in (A.2), we have

$$
\begin{aligned}
&[\boldsymbol{\lambda}_{k+1} - \bar{\boldsymbol{\lambda}}_{k+1}]^\top [A_{\mathrm{d}}\boldsymbol{x}_{k+1} - A_{\mathrm{d}}\boldsymbol{x}^\star] \\
&\quad = [\boldsymbol{\lambda}_{k+1} - \bar{\boldsymbol{\lambda}}_{k+1}]^\top [A_{\mathrm{d}}\boldsymbol{x}_{k+1} - \bar{\boldsymbol{d}}_{k+1} - A_{\mathrm{d}}\boldsymbol{x}^\star] \\
&\quad \overset{\text{(a)}}{=} [\boldsymbol{\lambda}_{k+1} - \bar{\boldsymbol{\lambda}}_{k+1}]^\top [\boldsymbol{z}_{k+1} - A_{\mathrm{d}}\boldsymbol{x}^\star] \qquad \text{(A.4)}
\end{aligned}
$$

where in (a) we used the definition of $\boldsymbol{z}_{k+1}$ (cf. (10)). Using (A.3) and (A.4) in (A.2) and multiplying by $2c$ we get

$$
\begin{aligned}
&2[\bar{\boldsymbol{\lambda}}_{k+1} - \boldsymbol{\lambda}^\star]^\top [\bar{\boldsymbol{\lambda}}_{k+1} - \bar{\boldsymbol{\lambda}}_k] \\
&\quad + 2c[\boldsymbol{\lambda}_{k+1} - \bar{\boldsymbol{\lambda}}_{k+1}]^\top [\boldsymbol{z}_{k+1} - A_{\mathrm{d}}\boldsymbol{x}^\star] \le 0.
\end{aligned} \qquad \text{(A.5)}
$$

Finally, using

$$
2[a-b]^\top [a-c] = \|a-b\|^2 + \|a-c\|^2 - \|c-b\|^2
$$

with the identifications $a = \bar{\boldsymbol{\lambda}}_{k+1}$, $b = \boldsymbol{\lambda}^\star$ and $c = \bar{\boldsymbol{\lambda}}_k$, (14) follows, thus concluding the proof. □

Proof of Theorem 1 (Convergence)

The dynamical evolution of $\boldsymbol{e}_k^{\boldsymbol{d}}$ and $\boldsymbol{e}_k^{\boldsymbol{\lambda}}$, given in equations (12) and (13), respectively, can be arranged as

$$
\boldsymbol{e}_{k+1} = F\boldsymbol{e}_k + G(\boldsymbol{u}_{k+1} - \boldsymbol{u}_k), \qquad \text{(A.6)}
$$

with

$$
\boldsymbol{e}_k = \begin{bmatrix} \boldsymbol{e}_k^{\boldsymbol{\lambda}} \\ c\,\boldsymbol{e}_k^{\boldsymbol{d}} \end{bmatrix}, \quad F = \begin{bmatrix} \tilde W & \tilde W \\ 0 & \tilde W \end{bmatrix}, \quad G = \begin{bmatrix} I \\ I \end{bmatrix},
$$

and $\boldsymbol{u}_k = c[\boldsymbol{z}_k - A_{\mathrm{d}}\boldsymbol{x}^\star]$. Let $P \in \mathbb{R}^{2Np \times 2Np}$ such that $P = P^\top \succ 0$ and consider

$$
\begin{aligned}
\|G\boldsymbol{u}_{k+1} - \boldsymbol{e}_{k+1}\|_P^2 &\overset{\text{(a)}}{=} \|G\boldsymbol{u}_k - \boldsymbol{e}_k + (I-F)\boldsymbol{e}_k\|_P^2 \\
&\overset{\text{(b)}}{=} \|G\boldsymbol{u}_k - \boldsymbol{e}_k\|_P^2 \\
&\quad + 2[G\boldsymbol{u}_k - \boldsymbol{e}_k]^\top P(I-F)\boldsymbol{e}_k \\
&\quad + \boldsymbol{e}_k^\top (I - F^\top) P (I-F)\boldsymbol{e}_k \\
&\overset{\text{(c)}}{=} \|G\boldsymbol{u}_k - \boldsymbol{e}_k\|_P^2 - \|\boldsymbol{e}_k\|_{P-F^\top PF}^2 \\
&\quad + 2\boldsymbol{u}_k^\top G^\top P(I-F)\boldsymbol{e}_k, \qquad \text{(A.7)}
\end{aligned}
$$

where in (a) we used the dynamics (A.6) and added $\pm\boldsymbol{e}_k$ inside the norm, in (b) we expanded the square, while in (c) we performed some algebraic manipulations. Summing (A.7) to the optimality-based inequality (14) (cf. Proposition 1) yields

$$
\begin{aligned}
&\|\bar{\boldsymbol{\lambda}}_{k+1} - \boldsymbol{\lambda}^\star\|^2 + 2\boldsymbol{u}_{k+1}^\top H\boldsymbol{e}_{k+1} + \|G\boldsymbol{u}_{k+1} - \boldsymbol{e}_{k+1}\|_P^2 \\
&\quad \le \|\bar{\boldsymbol{\lambda}}_k - \boldsymbol{\lambda}^\star\|^2 + 2\boldsymbol{u}_k^\top G^\top P(I-F)\boldsymbol{e}_k + \|G\boldsymbol{u}_k - \boldsymbol{e}_k\|_P^2 \\
&\qquad - \|\bar{\boldsymbol{\lambda}}_{k+1} - \bar{\boldsymbol{\lambda}}_k\|^2 - \|\boldsymbol{e}_k\|_{P-F^\top PF}^2, \qquad \text{(A.8)}
\end{aligned}
$$

with $H = [I \quad 0]$. Notice that, choosing P such that $P - F^\top PF \succ 0$, if the terms at $k+1$ were analogous to the terms at k, then (A.8) would describe a descent condition. To enforce this, we need to find a P such that

$$
\begin{aligned}
&H = G^\top P(I-F), && \text{(A.9a)} \\
&P - F^\top PF \succ 0, && \text{(A.9b)} \\
&P \succ 0. && \text{(A.9c)}
\end{aligned}
$$

Satisfying (A.9) translates into (A.8) being a non-expansive condition for the sequence $\{\|\bar{\boldsymbol{\lambda}}_{k+1} - \boldsymbol{\lambda}^\star\|^2 + 2\boldsymbol{u}_{k+1}^\top H\boldsymbol{e}_{k+1} + \|G\boldsymbol{u}_{k+1} - \boldsymbol{e}_{k+1}\|_P^2\}_{k\ge 0}$, i.e., a stability condition for the nonlinear dynamical system in (6), which represents the evolution of Algorithm 1.

Take P partitioned in blocks as follows

$$
P = \begin{bmatrix} P_1 & P_2 \\ P_2^\top & P_3 \end{bmatrix}.
$$

Owing to the fact that all eigenvalues of $\tilde W = W - (\mathcal{O} \otimes I_p)$ lies in the open unit circle, thus $(I - \tilde W)$ is invertible and, from the equality $MM^{-1} = M^{-1}M = I$ with $M = (I - \tilde W)$, we have the following identities

$$
\begin{aligned}
&\tilde W (I - \tilde W)^{-1} = (I - \tilde W)^{-1} - I, && \text{(A.10a)} \\
&(I - \tilde W)^{-1} \tilde W = (I - \tilde W)^{-1} - I. && \text{(A.10b)}
\end{aligned}
$$

Noticing that

$$(I-F)^{-1} = \begin{bmatrix} (I-\tilde{W})^{-1} & (I-\tilde{W})^{-1}\tilde{W}(I-\tilde{W})^{-1} \\ 0 & (I-\tilde{W})^{-1} \end{bmatrix},$$

condition (A.9a) can be rewritten as $G^\top P = H(I-F)^{-1}$ and translates into the following constraints on the blocks of P:

$$P_1 + P_2^\top = (I-\tilde{W})^{-1} \tag{A.11a}$$

$$\begin{aligned} P_2 + P_3 &= (I-\tilde{W})^{-1}\tilde{W}(I-\tilde{W})^{-1} \\ &\overset{\text{(a)}}{=} (I-\tilde{W})^{-2} - (I-\tilde{W})^{-1} \end{aligned} \tag{A.11b}$$

$$P_1 + P_2^\top + P_2 + P_3 = (I-\tilde{W})^{-2}, \tag{A.11c}$$

where in (a) we used (A.10a), while (A.11c) follows by simply summing (A.11a) and (A.11b) and will be useful in the forthcoming discussion. Note that, from (A.11a), we can further deduce that $P_2 = P_2^\top$ since both P_1 and $(I-\tilde{W})^{-1}$ are symmetric matrices owing to P being symmetric and Assumption 4, respectively.

We can thus express the matrix $P - F^\top P F$ as

$$\begin{aligned} &\begin{bmatrix} P_1 - \tilde{W}P_1\tilde{W} & P_2 - \tilde{W}(P_1+P_2)\tilde{W} \\ P_2^\top - \tilde{W}(P_1+P_2^\top)\tilde{W} & P_3 - \tilde{W}(P_1+P_2^\top+P_2+P_3)\tilde{W} \end{bmatrix} \\ &\overset{\text{(a)}}{=} \begin{bmatrix} P_1 - \tilde{W}P_1\tilde{W} & P_2 - \tilde{W}(I-\tilde{W})^{-1}\tilde{W} \\ P_2^\top - \tilde{W}(I-\tilde{W})^{-1}\tilde{W} & P_3 - \tilde{W}(I-\tilde{W})^{-2}\tilde{W} \end{bmatrix} \\ &\overset{\text{(b)}}{=} \begin{bmatrix} P_1 - \tilde{W}P_1\tilde{W} & \tilde{W} - (P_1 - I) \\ \tilde{W} - (P_1 - I) & P_1 - I \end{bmatrix}, \end{aligned} \tag{A.12}$$

where in (a) we used (A.11) and in (b) we leveraged algebraic identities in (A.10) together with (A.11a) and (A.11b) to express P_2 and P_3 as a function of P_1 only.

Next, we find a value for $P_1 \succ 0$ ensuring that $P - F^\top P F$ is positive definite. This requirement can be posed in terms of its blocks by means of the Schur complement lemma, i.e., $P - F^\top P F \succ 0$ if and only if

$$P_1 - I \succ 0, \tag{A.13a}$$

$$\begin{aligned} &P_1 - \tilde{W}P_1\tilde{W} - (\tilde{W} - (P_1 - I))(P_1 - I)^{-1}(\tilde{W} - (P_1 - I)) \\ &\quad = 2\tilde{W} + I - \tilde{W}(P_1 + (P_1 - I)^{-1})\tilde{W} \succ 0. \end{aligned} \tag{A.13b}$$

Take $P_1 = 2I$ and let $T\Lambda T^\top = \tilde{W}$ be the eigendecomposition of the symmetric matrix $\tilde{W}$, with $TT^\top = I$ and Λ being a diagonal matrix containing the eigenvalues of $\tilde{W}$. Condition (A.13a) is trivially satisfied by this choice of P_1, while (A.13b) becomes

$$\begin{aligned} 2\tilde{W} + I - \tilde{W}(2I + I)\tilde{W} &= 2\tilde{W} + I - 3\tilde{W}^2 \\ &= T(I + 2\Lambda - 3\Lambda^2)T^\top \succ 0, \end{aligned}$$

which is equivalent to $I + 2\Lambda - 3\Lambda^2 \succ 0$ and always holds true when all the eigenvalues of $\tilde{W}$ lies within $(-1/3, 1)$. In view of Assumption 5, this condition is guaranteed.

Finally, using again Schur complement lemma, it is also easy to prove that $P_1 = 2I$ satisfies condition (A.9c). In fact, we have that

$$P = \begin{bmatrix} 2I & (I-\tilde{W})^{-1} - 2I \\ (I-\tilde{W})^{-1} - 2I & (I-\tilde{W})^{-2} - 2(I-\tilde{W})^{-1} + 2I \end{bmatrix} \succ 0$$

if and only if

$$2I \succ 0, \tag{A.14a}$$

$$\begin{aligned} &(I-\tilde{W})^{-2} - 2(I-\tilde{W})^{-1} + 2I - \frac{1}{2}\left((I-\tilde{W})^{-1} - 2I\right)^2 \\ &\quad = \frac{1}{2}(I-\tilde{W})^{-2} \succ 0. \end{aligned} \tag{A.14b}$$

Condition (A.14b) is satisfied since $\tilde{W}$ has all eigenvalues in the open unit circle and hence $I - \tilde{W} \succ 0$.

From (A.9a) we can rewrite (A.8) as

$$\begin{aligned} &\|\bar{\boldsymbol{\lambda}}_{k+1} - \boldsymbol{\lambda}^\star\|^2 + 2\boldsymbol{u}_{k+1}^\top H \boldsymbol{e}_{k+1} + \|G\boldsymbol{u}_{k+1} - \boldsymbol{e}_{k+1}\|_P^2 \\ &\quad \leq \|\bar{\boldsymbol{\lambda}}_k - \boldsymbol{\lambda}^\star\|^2 + 2\boldsymbol{u}_k^\top H \boldsymbol{e}_k + \|G\boldsymbol{u}_k - \boldsymbol{e}_k\|_P^2 \\ &\qquad - \|\bar{\boldsymbol{\lambda}}_{k+1} - \bar{\boldsymbol{\lambda}}_k\|^2 - \|\boldsymbol{e}_k\|_Q^2, \end{aligned} \tag{A.15}$$

where we set $Q = P - F^\top P F \succ 0$. Rearranging some terms, and summing (A.15) from $k = 0$ to $M - 1$, with $M \in \mathbb{N}$, we have

$$\begin{aligned} &\sum_{k=0}^{M-1} \|\bar{\boldsymbol{\lambda}}_{k+1} - \bar{\boldsymbol{\lambda}}_k\|^2 + \|\boldsymbol{e}_k\|_Q^2 \\ &\qquad \leq \|\bar{\boldsymbol{\lambda}}_0 - \boldsymbol{\lambda}^\star\|^2 + 2\boldsymbol{u}_0^\top H \boldsymbol{e}_0 + \|G\boldsymbol{u}_0 - \boldsymbol{e}_0\|_P^2 \\ &\qquad\quad - \|\bar{\boldsymbol{\lambda}}_M - \boldsymbol{\lambda}^\star\|^2 - 2\boldsymbol{u}_M^\top H \boldsymbol{e}_M - \|G\boldsymbol{u}_M - \boldsymbol{e}_M\|_P^2 \\ &\qquad \leq \|\bar{\boldsymbol{\lambda}}_0 - \boldsymbol{\lambda}^\star\|^2 + \|G\boldsymbol{u}_0 - \boldsymbol{e}_0\|_P^2 + 2\mathcal{C}, \end{aligned}$$

where in the second inequality we neglected the terms $-\|\bar{\boldsymbol{\lambda}}_M - \boldsymbol{\lambda}^\star\|^2$ and $-\|G\boldsymbol{u}_M - \boldsymbol{e}_M\|_P^2$ in the right-hand side since they are non-positive (recall $P \succ 0$) and we used the fact that $|2\boldsymbol{u}_k^\top H \boldsymbol{e}_k| \leq \mathcal{C}$ for all $k \geq 0$ owing to the results of Lemma 3. Taking the limit as $M \to \infty$ finally yields

$$\sum_{k=0}^{\infty} \left(\|\bar{\boldsymbol{\lambda}}_{k+1} - \bar{\boldsymbol{\lambda}}_k\|^2 + \|\boldsymbol{e}_k\|_Q^2\right) < \infty,$$

which, recalling that $Q \succ 0$, implies that $\lim_{k\to\infty} \|\bar{\boldsymbol{\lambda}}_{k+1} - \bar{\boldsymbol{\lambda}}_k\| = 0$ and $\lim_{k\to\infty} \|\boldsymbol{e}_k\| = 0$. Hence, $\lim_{k\to\infty} \|\boldsymbol{e}_k^{\boldsymbol{d}}\| = 0$ and $\lim_{k\to\infty} \|\boldsymbol{e}_k^{\boldsymbol{\lambda}}\| = 0$, thus proving points (i) and (ii), respectively.

Since $\bar{\boldsymbol{d}}_{k+1} = \frac{1}{c}(\bar{\boldsymbol{\lambda}}_{k+1} - \bar{\boldsymbol{\lambda}}_k)$, we also have that $\lim_{k\to\infty} \|\bar{\boldsymbol{d}}_k\| = 0$ and therefore $\lim_{k\to\infty} \|\bar{d}_k\| = 0$, thus proving point (iii).

From (A.15) we also have that the sequence

$$\{\|\bar{\boldsymbol{\lambda}}_k - \boldsymbol{\lambda}^\star\|^2 + 2\boldsymbol{u}_k^\top H \boldsymbol{e}_k + \|G\boldsymbol{u}_k - \boldsymbol{e}_k\|_P^2\}_{k\geq 0}$$

is non-increasing, bounded below since $|2\boldsymbol{u}_k^\top H \boldsymbol{e}_k| \leq \mathcal{C}$ and $\|G\boldsymbol{u}_k - \boldsymbol{e}_k\|_P^2 \geq 0$ (recall $P \succ 0$), and, therefore, convergent. Since $\|\boldsymbol{e}_k\|$ is vanishing and $\{\boldsymbol{u}_k\}_{k\geq 0}$ is bounded, we have that also the sequence $\{\|\bar{\boldsymbol{\lambda}}_k - \boldsymbol{\lambda}^\star\|^2 + \|G\boldsymbol{u}_k\|_P^2\}_{k\geq 0}$ is convergent. A straightforward computation using (A.11c) shows that $G^\top P G = (I-\tilde{W})^{-2} \succ 0$ which, recalling the definition of $\boldsymbol{u}_k = c\,(\boldsymbol{z}_k - A_\mathrm{d}\boldsymbol{x}^\star)$, implies that the sequence

$$\{\|\bar{\boldsymbol{\lambda}}_k - \boldsymbol{\lambda}^\star\|^2 + c^2\|\boldsymbol{z}_k - A_\mathrm{d}\boldsymbol{x}^\star\|^2_{(I-\tilde{W})^{-2}}\}_{k\geq 0}$$

is convergent, which finally implies that

$$\{\|\bar{\boldsymbol{\lambda}}_k - \boldsymbol{\lambda}^\star\|^2 + c^2\|\boldsymbol{z}_k - A_\mathrm{d}\boldsymbol{x}^\star\|^2\}_{k\geq 0}$$

is convergent due to norm equivalence, thus proving point (iv) and concluding the proof. □

Proof of Theorem 2 (Optimality)

Following the first steps in the proof of Proposition 1, for any optimal solution $\boldsymbol{x}^\star$ for $\mathcal{P}$, we can write

$$f(\boldsymbol{x}_{k+1}) + \boldsymbol{\lambda}_{k+1}^\top[A_\mathrm{d}\boldsymbol{x}_{k+1} - A_\mathrm{d}\boldsymbol{x}^\star] \leq f(\boldsymbol{x}^\star).$$

By adding and subtracting $\bar{\boldsymbol{\lambda}}_{k+1}^\top[A_\mathrm{d}\boldsymbol{x}_{k+1} - A_\mathrm{d}\boldsymbol{x}^\star]$ to the left-hand side, we have

$$\begin{aligned} f(\boldsymbol{x}_{k+1}) &+ \bar{\boldsymbol{\lambda}}_{k+1}^\top[A_\mathrm{d}\boldsymbol{x}_{k+1} - A_\mathrm{d}\boldsymbol{x}^\star] \\ &+ {\boldsymbol{e}_{k+1}^{\boldsymbol{\lambda}}}^\top[A_\mathrm{d}\boldsymbol{x}_{k+1} - A_\mathrm{d}\boldsymbol{x}^\star] \leq f(\boldsymbol{x}^\star). \end{aligned}$$

Noticing that

$$\begin{aligned} \bar{\boldsymbol{\lambda}}_{k+1}^\top[A_\mathrm{d}\boldsymbol{x}_{k+1} - A_\mathrm{d}\boldsymbol{x}^\star] &= \bar{\lambda}_{k+1}^\top[A\boldsymbol{x}_{k+1} - A\boldsymbol{x}^\star] \\ &= \bar{\lambda}_{k+1}^\top[A\boldsymbol{x}_{k+1} - b] \\ &= \bar{\lambda}_{k+1}^\top\, N\, \bar{d}_{k+1}, \qquad \text{(A.16)} \end{aligned}$$

where the last inequality is due to Lemma 1, we have that

$$f(\boldsymbol{x}_{k+1}) + N\bar{\lambda}_{k+1}^\top\bar{d}_{k+1} + {\boldsymbol{e}_{k+1}^{\boldsymbol{\lambda}}}^\top[A_\mathrm{d}\boldsymbol{x}_{k+1} - A_\mathrm{d}\boldsymbol{x}^\star] \leq f(\boldsymbol{x}^\star).$$

From Theorem 1(iv) we have that the sequence

$$\{\|\bar{\boldsymbol{\lambda}}_k - \boldsymbol{\lambda}^\star\|^2 + c^2\|\boldsymbol{z}_k - A_\mathrm{d}\boldsymbol{x}^\star\|^2\}_{k\geq 0}$$

is bounded. Since $\{\boldsymbol{z}_k\}_{k\geq 0}$ is also bounded by Lemma 3, we have that $\{\bar{\boldsymbol{\lambda}}_k\}_{k\geq 0}$ and, therefore, also $\{\bar{\lambda}_k\}_{k\geq 0}$ are both bounded. Recalling that $\{\boldsymbol{x}_k\}_{k\geq 0}$ is bounded, by Theorem 1 points (iii) and (ii) respectively, it holds

$$\lim_{k\to\infty} \bar{\lambda}_{k+1}^\top\bar{d}_{k+1} = 0, \qquad \text{(A.17)}$$

$$\lim_{k\to\infty} {\boldsymbol{e}_{k+1}^{\boldsymbol{\lambda}}}^\top[A_\mathrm{d}\boldsymbol{x}_{k+1} - A_\mathrm{d}\boldsymbol{x}^\star] = 0, \qquad \text{(A.18)}$$

and therefore

$$\begin{aligned} \limsup_{k\to\infty} f(\boldsymbol{x}_{k+1}) &+ N\bar{\lambda}_{k+1}^\top\bar{d}_{k+1} + {\boldsymbol{e}_{k+1}^{\boldsymbol{\lambda}}}^\top[A_\mathrm{d}\boldsymbol{x}_{k+1} - A_\mathrm{d}\boldsymbol{x}^\star] \\ &= \limsup_{k\to\infty} f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}^\star). \qquad \text{(A.19)} \end{aligned}$$

Since $\boldsymbol{x}_k \in X$ for all $k \geq 0$ and, by Lemma 1 and Theorem 1(iii),

$$\lim_{k\to\infty} \|A\boldsymbol{x}_k - b\| = \lim_{k\to\infty} \|\bar{d}_k\| = 0,$$

we know that any limit point $\tilde{\boldsymbol{x}}$ of the sequence $\{\boldsymbol{x}_k\}_{k\geq 0}$ is feasible for $\mathcal{P}$ and, thus, satisfies $f(\tilde{\boldsymbol{x}}) \geq f(\boldsymbol{x}^\star)$. Combining this fact with (A.19) we can conclude that $f(\tilde{\boldsymbol{x}}) = f(\boldsymbol{x}^\star)$ for any limit point $\tilde{\boldsymbol{x}}$. Since all limit points of $\{\boldsymbol{x}_k\}_{k\geq 0}$ are feasible and achieve the optimal value, they are optimal solutions for $\mathcal{P}$, proving Theorem 2(i).

Next, we prove the convergence of the local dual variables estimates to an optimal dual solution. Similarly to the proof of Proposition 1, by applying (Bertsekas and Tsitsiklis, 1989, Lemma 4.1) to Step 8 we have that $x_{i,k+1}$ satisfies

$$\begin{aligned} f_i(x_{i,k+1}) &+ [\ell_{i,k} + c(A_i x_{i,k+1} - A_i x_{i,k} + \delta_{i,k})]^\top A_i x_{i,k+1} \\ &\leq f_i(x_i) + [\ell_{i,k} + c(A_i x_{i,k+1} - A_i x_{i,k} + \delta_{i,k})]^\top A_i x_i, \end{aligned}$$

for all $x_i \in X_i$. Noticing that $d_{i,k+1} = \delta_{i,k} + A_i x_{i,k+1} - A_i x_{i,k}$ and $\lambda_{i,k+1} = \ell_{i,k} + c\, d_{i,k+1}$ from Steps 9 and 10, respectively, and subtracting $\lambda_{i,k+1}^\top b_i$, we have

$$\begin{aligned} f_i(x_{i,k+1}) &+ \lambda_{i,k+1}^\top(A_i x_{i,k+1} - b_i) \\ &\leq f_i(x_i) + \lambda_{i,k+1}^\top(A_i x_i - b_i) \end{aligned}$$

for all $x_i \in X_i$, which is equivalent to

$$x_{i,k+1} \in \operatorname*{argmin}_{x_i\in X_i} f_i(x_i) + \lambda_{i,k+1}^\top(A_i x_i - b_i)$$

and, thus, by definition of $\varphi_i(\lambda)$ in (2), we have

$$\varphi_i(\lambda_{i,k+1}) = f_i(x_{i,k+1}) + \lambda_{i,k+1}^\top(A_i x_{i,k+1} - b_i),$$

if $b_1, \ldots, b_N$ are such that $\sum_{i=1}^N b_i = b$. Setting $b_i = A_i x_i^\star$, summing over $i = 1, \ldots, N$, adding and subtracting $\bar{\boldsymbol{\lambda}}_{k+1}^\top[A_\mathrm{d}\boldsymbol{x}_{k+1} - A_\mathrm{d}\boldsymbol{x}^\star]$ to the right hand side, and exploiting again (A.16), we obtain

$$\begin{aligned} \sum_{i=1}^N \varphi_i(\lambda_{i,k+1}) = f(\boldsymbol{x}_{k+1}) &+ N\,\bar{\lambda}_{k+1}^\top\bar{d}_{k+1} \\ &+ {\boldsymbol{e}_{k+1}^{\boldsymbol{\lambda}}}^\top[A_\mathrm{d}\boldsymbol{x}_{k+1} - A_\mathrm{d}\boldsymbol{x}^\star]. \end{aligned} \qquad \text{(A.20)}$$

Since the sequence $\{\bar{\lambda}_{k+1}\}_{k\geq 0}$ is bounded, it admits a convergent subsequence $\{(\bar{\lambda}_{k+1}, \boldsymbol{x}_{k+1})\}_{k\in\mathcal{K}}$ with $\mathcal{K} \subseteq \mathbb{N}$. Let $(\tilde{\lambda}, \tilde{\boldsymbol{x}})$ be its limit point. Now, taking the limit of (A.20) across $\mathcal{K}$, we have

$$\begin{aligned} \lim_{\mathcal{K}\ni k\to\infty} \sum_{i=1}^N \varphi_i(\lambda_{i,k+1}) = \lim_{k\in\mathcal{K}} \Big\{ f(\boldsymbol{x}_{k+1}) &+ N\,\bar{\lambda}_{k+1}^\top\bar{d}_{k+1} \\ &+ {\boldsymbol{e}_{k+1}^{\boldsymbol{\lambda}}}^\top[A_\mathrm{d}\boldsymbol{x}_{k+1} - A_\mathrm{d}\boldsymbol{x}^\star]\Big\} \end{aligned}$$

$$
\begin{aligned}
&\overset{\text{(a)}}{=} f(\tilde{\boldsymbol{x}}) \\
&\overset{\text{(b)}}{=} f(\boldsymbol{x}^\star) \\
&\overset{\text{(c)}}{\geq} \max_{\lambda} \sum_{i=1}^{N} \varphi_i(\lambda),
\end{aligned} \tag{A.21}
$$

where in (a) we used (A.17) and (A.18), the fact that limit points of $\{\boldsymbol{x}_k\}_{k\geq 0}$ achieve the optimal value in (b), and weak duality in (c). On the other hand, from (2), we have

$$
\begin{aligned}
\lim_{\mathcal{K}\ni k\to\infty} \varphi_i(\lambda_{i,k+1}) &\leq \lim_{\mathcal{K}\ni k\to\infty} \Big\{ f_i(x_i) + \lambda_{i,k+1}^\top (A_i x_i - b_i) \Big\} \\
&= f_i(x_i) + \Big(\lim_{\mathcal{K}\ni k\to\infty} \bar{\lambda}_{k+1} \Big)^\top (A_i x_i - b_i) \\
&= f_i(x_i) + \tilde{\lambda}^\top (A_i x_i - b_i),
\end{aligned} \tag{A.22}
$$

for all $x_i \in X_i$, where the first equality is due to $\lim_{k\to\infty} \lambda_{i,k} = \bar{\lambda}_k$, $i = 1, \dots, N$ (recall $\lim_{k\to\infty} \|\boldsymbol{e}_k^{\boldsymbol{\lambda}}\| = 0$), and in the last equality we plugged the limit point $\tilde{\lambda}$ of $\{\bar{\lambda}_{k+1}\}_{k\in\mathcal{K}}$. Setting x_i in (A.22) equal to the minimizer of $\varphi_i(\tilde{\lambda})$ and summing over $i = 1, \dots, N$ yields

$$
\lim_{\mathcal{K}\ni k\to\infty} \sum_{i=1}^{N} \varphi_i(\lambda_{i,k+1}) \leq \sum_{i=1}^{N} \varphi_i(\tilde{\lambda}) \leq \max_{\lambda} \sum_{i=1}^{N} \varphi_i(\lambda).
$$

Combining the latter with (A.21) gives

$$
\sum_{i=1}^{N} \varphi_i(\tilde{\lambda}) = \max_{\lambda} \sum_{i=1}^{N} \varphi_i(\lambda), \tag{A.23}
$$

which means that $\tilde{\lambda}$ is optimal for $\mathcal{D}$.

Recall that, from Theorem 1(iv), we have that the sequence

$$
\{\|\bar{\boldsymbol{\lambda}}_k - \boldsymbol{\lambda}^\star\|^2 + c^2\|\boldsymbol{z}_k - A_{\mathrm{d}}\boldsymbol{x}^\star\|^2\}_{k\geq 0}
$$

is convergent for any pair $\boldsymbol{x}^\star$ and $\lambda^\star$ which are optimal for $\mathcal{P}$ and $\mathcal{D}$, respectively. We can thus take $\boldsymbol{\lambda}^\star = \mathbb{1}_N \otimes \tilde{\lambda}$ and $\boldsymbol{x}^\star = \tilde{\boldsymbol{x}}$ and conclude that

$$
\lim_{\mathcal{K}\ni k\to\infty} \Big\{ \|\bar{\boldsymbol{\lambda}}_k - \boldsymbol{\lambda}^\star\|^2 + c^2\|\boldsymbol{z}_k - A_{\mathrm{d}}\boldsymbol{x}^\star\|^2 \Big\} = 0.
$$

But since $\{\|\bar{\boldsymbol{\lambda}}_k - \boldsymbol{\lambda}^\star\|^2 + c^2\|\boldsymbol{z}_k - A_{\mathrm{d}}\boldsymbol{x}^\star\|^2\}_{k\geq 0}$ is convergent, all its limit points are the same, and therefore

$$
\lim_{\mathcal{K}\ni k\to\infty} \Big\{ \|\bar{\boldsymbol{\lambda}}_k - \boldsymbol{\lambda}^\star\|^2 + c^2\|\boldsymbol{z}_k - A_{\mathrm{d}}\boldsymbol{x}^\star\|^2 \Big\} = 0,
$$

for some primal optimal solution $\boldsymbol{x}^\star$ and some optimal dual solution $\lambda^\star$, which implies that, for all $i = 1, \dots, N$,

$$
\begin{aligned}
&\lim_{k\to\infty} \|\lambda_{i,k} - \lambda^\star\| = 0 \\
&\lim_{k\to\infty} \|A_i x_{i,k} - A_i x_i^\star\| = 0
\end{aligned}
$$

since $\boldsymbol{e}_k^{\boldsymbol{\lambda}}$ is vanishing. So that Theorem 2(ii) follows and the proof is complete. □