Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Combining ADMM and tracking over networks for distributed constraint-coupled optimization ⋆

Alessandro Falsone * **Ivano Notarnicola** ** **Giuseppe Notarstefano** **
Maria Prandini *

* *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Via Giuseppe Ponzio, 34/5, 20133, Milano, Italy. (e-mail: name.surname@polimi.it)*
** *Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione, Alma Mater Studiorum Università di Bologna, Viale del Risorgimento 2, 40136, Bologna, Italy. (e-mail: name.surname@unibo.it)*

Abstract: In this paper, we propose a novel distributed algorithm to address constraint-coupled optimization problems in which agents in a network aim at cooperatively minimizing the sum of local objective functions subject to individual constraints and a common, linear coupling constraint. Our optimization scheme embeds a dynamic average consensus protocol in the (parallel) Alternating Direction Method of Multipliers (ADMM) to design a fully distributed algorithm. More precisely, the dual variable update step of the master node in ADMM is now performed locally by the agent, which update their own copy of the dual variable in a consensus-based scheme using a dynamic average mechanism to track the coupling constraint violation. Under convexity, we show convergence of the primal solution estimates to an optimal solution of the constraint-coupled target problem. A numerical example supports the theoretical results.

Keywords: Distributed Optimization, Constraint-Coupled Optimization, ADMM

1. INTRODUCTION

This paper investigates a distributed *constraint-coupled* optimization set-up often arising in network control applications, where agents in a network want to minimize the sum of local cost functions, each one depending on a local variable, subject to individual constraints and a common linear coupling constraint. The presence of the coupling constraint, involving all the decision variables in the network, makes the problem solution nontrivial especially in a distributed context, where agents can communicate only with their neighbors.

Despite of the relevance of the constraint-coupled set-up, until recently, most of the effort in the distributed optimization literature has been devoted to solve *decision-coupled* optimization problems where the sum of local cost functions depends on a common decision variable, which introduces the coupling. Various methods have been proposed to address decision-coupled problems, such as consensus-based methods leveraging (sub)gradient iterations and proximal operators (Johansson et al. (2008); Nedić and Ozdaglar (2009); Nedić et al. (2010); Zanella et al. (2011); Shi et al. (2015); Margellos et al. (2018)), algorithms based on Lagrangian duality (Duchi et al. (2012); Necoara and Nedelcu (2015); Zhu and Martínez (2012); Mateos-Núnez and Cortés (2017)), and strategies based on the Alternating Direction Method Multipliers (ADMM) [1] (Mota et al. (2013); Ling and Ribeiro (2014); Shi et al. (2014); Jakovetić et al. (2015); Iutzeler et al. (2016); Makhdoumi and Ozdaglar (2017)). The available distributed approaches typically suffer from slow convergence rate due to, e.g., the use of diminishing step-size rules. Recently (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)), a tracking technique based on the dynamic average consensus – originally proposed in Zhu and Martínez (2010) and more deeply discussed in Kia et al. (2019) – has been combined with gradient schemes to design distributed optimization algorithms with constant step-size for decision-coupled problems, significantly improving their convergence rate.

A possibility to address a constraint-coupled problem on a network is then to interpret it as a decision-coupled problem and apply the available solution strategies. This, however has two major drawbacks. First, one would need to stack the local decision variables of all agents in a common decision vector, which then needs to be stored, updated, and exchanged among neighboring agents, thus wasting memory, computational resources, and bandwidth. Secondly, each agent would require also some information about other agents' constraints, thus raising privacy issues. These drawbacks call for novel and efficient strategies to *directly* address constraint-coupled optimization over networks by leveraging the structure of the considered set-up. Next, we review some recent works addressing this scenario using Lagrangian duality.

⋆ 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.

[1] For more details regarding ADMM we refer the interested reader to Bertsekas and Tsitsiklis (1989); Boyd et al. (2011).

In Chang et al. (2014) a distributed consensus-based primal-dual perturbation algorithm is proposed. In Falsone et al. (2017) a distributed dual subgradient algorithm is described. An alternative approach based on successive duality steps is investigated in Notarnicola and Notarstefano (2020). All these works adopt a diminishing step-size to guarantee convergence to a primal optimal solution, which, however, dramatically reduces the overall convergence rate. The work in Simonetto and Jamali-Rad (2016) is similar to Falsone et al. (2017) but uses a constant step-size and shows convergence only to a neighborhood of the optimal solution. In Chang (2016) a strategy combining consensus-ADMM and proximal operators allowing for a constant step-size is introduced, while Liang et al. (2020) a primal-dual algorithm with constant step-size is proposed. Tracking mechanisms have been also employed to solve constraint-coupled problems based on augmented Lagrangian approaches, Kia (2017).

In this paper, 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 variable (Mota et al. (2013); Ling and Ribeiro (2014); Shi et al. (2014); Jakovetić et al. (2015); Iutzeler et al. (2016); Makhdoumi and Ozdaglar (2017)), our Tracking-ADMM distributed algorithm embeds a tracking mechanism into the parallel ADMM for constraint-coupled problems. Our algorithm has the following appealing features:

- (i) no parameter tuning is needed, in fact our algorithm works for all the choices of a constant penalty parameter and no other coefficients are necessary;
- (ii) agents solve optimization problems depending on their local small-sized decision vector and asymptotically compute only their portion of an optimal solution to the given problem;
- (iii) the local estimate of the coupling constraint violation, provided by the tracking mechanism, gives each agent an approximate local assessment on the amount of infeasibility, which can be useful, e.g., in designing distributed (receding horizon) control schemes.

It is worth mentioning that the recent paper Zhang and Zavlanos (2018) uses a tracking mechanism similar to the one proposed in this paper. However, the algorithm proposed in Zhang and Zavlanos (2018) exhibits a (non-zero) steady state error, which can be set by the user by carefully selecting the number of consensus steps per iteration to be performed. Furthermore, the proposed approach compares favorably in simulation against other methods in terms of convergence speed.

The rest of the paper is organized as follows. In Section 2 we present the problem set-up and some preliminaries on ADMM. In Section 3 we introduce our novel Tracking-ADMM distributed algorithm along with its convergence guarantees. In Section 4 we test our algorithm in a numerical example, and in Section 5 we draw some conclusions. Due to space limitation, we give here only a sketch of the proof of the main result. A detailed proof with all technical derivations can be found in Falsone et al. (2020).

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).

2.1 Optimization Problem and Assumptions

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 vector $x_i \in \mathbb{R}^{n_i}$ 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, the following mathematical program can be posed

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

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

In order to deal with the coupling constraint $\sum_{i=1}^N A_i x_i = b$ the dual of $\mathcal{P}$ can be introduced. To this end, let $\boldsymbol{x} = [x_1^\top \cdots x_N^\top]^\top$, consider a vector $\lambda \in \mathbb{R}^p$ of Lagrange multipliers and let

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

be 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

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

with the i-th contribution $\varphi_i : \mathbb{R}^p \to \mathbb{R}$ defined as

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

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

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

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

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 $\lambda^\star$. □

Notice that, under Assumption 2, by the Saddle Point Theorem in (Bertsekas and Tsitsiklis, 1989, pag. 665) we have that $\boldsymbol{x}^\star \in X$ and

$$L(\boldsymbol{x}^\star,\lambda) \le L(\boldsymbol{x}^\star,\lambda^\star) \le L(\boldsymbol{x},\lambda^\star), \tag{3}$$

for all $\boldsymbol{x} \in X$ and for all λ, where $X = X_1 \times \cdots \times X_N$.

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

2.2 The ADMM 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 $d_0 \in \mathbb{R}^p$ and $\lambda_0 \in \mathbb{R}^p$, the ADMM algorithm is a parallel scheme in which, at each iteration $k \geq 0$, a set of m processors and a master node perform the following two alternate steps. First, each processor $i \in \{1, \dots, N\}$ computes and sends to the master node the minimizer of the following optimization problem

$$x_{i,k+1} \in \operatorname*{argmin}_{x_i \in X_i} \Big\{ f_i(x_i) + \lambda_k^\top A_i x_i + \frac{c}{2} \|A_i x_i - A_i x_{i,k} + d_k\|^2 \Big\}, \tag{4a}$$

where $c > 0$ is a (constant) penalty parameter. Then, the master node updates and broadcasts back to the processors the following two vectors

$$d_{k+1} = \frac{1}{N} \sum_{i=1}^{N} (A_i x_{i,k+1} - b_i) \tag{4b}$$

$$\lambda_{k+1} = \lambda_k + c\, d_{k+1}, \tag{4c}$$

where the parameter c is the same as in (4a). The reader should note that d_{k+1} has no dynamics and can be seen as the average of the local contributions $A_i x_{i,k+1} - b_i$ to the coupling constraint. This "average" structure will be exploited in the design of our Tracking-ADMM distributed algorithm.

The evolution of (4) is analyzed in (Bertsekas and Tsitsiklis, 1989, pp. 254-256) and its convergence property is reported below, cf. (Bertsekas and Tsitsiklis, 1989, Proposition 4.2).

Proposition 1. Let Assumptions 1 and 2 hold. Then, any limit point $\boldsymbol{x}^\star = [x_1^{\star\top} \cdots x_N^{\star\top}]^\top$ of the primal sequences $\{x_{i,k}\}_{k\geq0}$, $i \in \{1, \dots, N\}$, generated by (4a), is an optimal solution (vector) of $\mathcal{P}$, and the dual sequence $\{\lambda_k\}_{k\geq0}$, generated by (4c), converges to an optimal solution $\lambda^\star$ of $\mathcal{D}$. □

Notice that, since the limit points of the primal sequence are optimal, they are necessarily feasible for the coupling constraint. Thus, it follows that $\{d_k\}_{k\geq0}$ converges to zero. Moreover, it is worth mentioning that no requirement on the penalty parameter c is necessary, albeit its value can affect the convergence rate of ADMM.

Note that the ADMM algorithm (4) is parallel since it requires a master node. This hampers the applicability of ADMM to a distributed computation framework, where peer agents communicate only with neighbors in a communication graph.

In the next section, we devise our fully distributed Tracking-ADMM.

3. TRACKING-ADMM DISTRIBUTED ALGORITHM

We assume that the N agents communicate according to a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, N\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. The presence of edge (i,j) in $\mathcal{E}$ models the fact that agent i receives information from agent j. 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 $i \in \{1, \dots, N\}$ to ease the notation. Also, 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. □

3.1 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 (4c) for λ_k in the parallel ADMM represents a gradient ascent step on $\mathcal{D}$. Its distributed counterpart can be obtained by employing a consensus-based gradient iteration, as done in, e.g., Nedić and Ozdaglar (2009). Thus, we let each agent i maintain a vector $\lambda_{i,k} \in \mathbb{R}^p$ representing a local version (or copy) of λ_k in (4c). If d_{k+1} were available to each agent, $\lambda_{i,k} \in \mathbb{R}^p$ could be updated according to a consensus-based scheme to force agreement of the local copies, i.e.,

$$\lambda_{i,k+1} = \sum_{j \in \mathcal{N}_i} w_{ij}\, \lambda_{j,k} + c\, d_{k+1}, \tag{5}$$

for all $i = 1, \dots, N$, where $w_{ij} \in \mathbb{R}$, $j \in \mathcal{N}_i$, are proper coefficients describing how agent i weights the information received by its neighbor j.

However, update (5) cannot be implemented in a fully distributed scheme since $d_{k+1} \in \mathbb{R}^p$ is not locally available and should be computed by a master node, as in (4b). In order to overcome this issue, we equip agent i with a local, auxiliary variable, denoted by $d_{i,k} \in \mathbb{R}^p$, that serves as a local estimate of d_k. Since the vector d_k is the average value of $A_i x_{i,k} - b_i$, $i = 1, \dots, N$, (cf. eq. (4b)), we propose to update $d_{i,k}$ according to a (distributed) dynamic average consensus mechanism Zhu and Martínez (2010); Kia et al. (2019). In this way, the variable $d_{i,k}$ 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 (5) makes the update of $\lambda_{i,k}$ fully distributed. Formally, the update law for $d_{i,k}$ is reported in Steps 6 and 9 while the corresponding update for $\lambda_{i,k}$ is given in Steps 7 and 10 in Algorithm 1.

Clearly, since the centralized quantities λ_k and d_k have been replaced by local counterparts, the local minimization needs to be consistently adapted. Specifically, we propose to implement the local minimization to compute $x_{i,k+1}$ as shown in Step 8, where λ_k and d_k of the original centralized update (4a), are replaced by the local averages $\ell_{i,k}$ and $\delta_{i,k}$, respectively (cf. Step 6 and 7).

Algorithm 1 summarizes the proposed Tracking-ADMM from the perspective of agent i.

Some remarks are in order. First, we point out that all the steps in the distributed algorithm are well posed. Specifically, in Step 8, $c > 0$ is a constant penalty parameter and the minimization is well defined in view of Assumption 1. Moreover, all the updates are fully distributed, in the sense

Algorithm 1 Tracking-ADMM for agent i

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: $k \leftarrow k+1$

that they can always be performed based on information locally known or collected via neighboring communications at each iteration k.

Consistently with other tracking-based approaches as, e.g., the ones mentioned in the introduction, the initialization of the local trackers $d_{i,k}$ as per Step 3 is crucial for guaranteeing converge to a feasible point for $\mathcal{P}$. In absence of prior information, one may set $b_i = b/N$ for all $i = 1,\dots,N$, as b and N are known to every agent.

3.2 Algorithm Convergence

Before stating the convergence result for our Tracking-ADMM distributed algorithm, we introduce some additional assumptions on the consensus weights associated to the communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

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

- $\sum_{i=1}^N w_{ij} = 1$ for all $j \in \mathcal{V}$,
- $\sum_{j=1}^N w_{ij} = 1$ for all $i \in \mathcal{V}$,

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}. Assumption 4 translates into requiring the consensus matrix $\mathcal{W}$ to be symmetric and doubly-stochastic, i.e., $\mathcal{W}\mathbf{1}_N = \mathcal{W}^\top \mathbf{1}_N = \mathbf{1}_N$, with $\mathbf{1}_N$ being the N-dimensional vector with all entries equal to one. We should point out that this assumption is 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. The matrix $\mathcal{W}$ is positive semidefinite. □

Note that Assumption 5 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.

The main result of the paper, i.e., the convergence of Algorithm 1, is summarized in the following theorem.

Theorem 1. Under Assumptions 1-5, the sequences generated by Tracking-ADMM are such that:

(i) any limit point of the primal sequences $\{x_{i,k}\}_{k\geq 0}$, $i = 1,\dots,N$, is an optimal solution $\boldsymbol{x}^\star = [x_1^{\star\top} \cdots x_N^{\star\top}]^\top$ of $\mathcal{P}$;
(ii) each dual sequence $\{\lambda_{i,k}\}_{k\geq 0}$ converges to a (common) optimal solution $\lambda^\star$ of $\mathcal{D}$, for all $i = 1,\dots,N$;
(iii) each tracker sequence $\{d_{i,k}\}_{k\geq 0}$ converges to zero, for all $i = 1,\dots,N$. □

Like the parallel ADMM, also the proposed Tracking-ADMM works for any choice of $c > 0$, but its actual value can affect the convergence rate.

We point out that Theorem 1 guarantees that the sequences $\{x_{i,k}\}_{k\geq 0}$ are asymptotically optimal, hence feasible for the local and the coupling constraint, without the need of any recovery procedure and without requiring strict convexity of the primal objective functions.

Also, the local tracker convergence ensures that each agent is able to locally estimate the amount of infeasibility of the current primal iterates for the coupling constraint. This can be very useful in applications where feasibility up to a given tolerance is sufficient.

3.3 Sketch of the proof

In this subsection we sketch the main idea behind the proof of Theorem 1. The complete proof along with all preparatory results can be found in Falsone et al. (2020).

Algorithm 1 can be regarded as a discrete-time dynamical system constituted by two parts: a) two consensus and update steps for $d_{i,k}$ and $\lambda_{i,k}$ (see Steps 6 and 7 together with Steps 9 and 10, respectively) giving rise to a linear dynamics in $d_{i,k}$ and $\lambda_{i,k}$, $i = 1,\dots,m$; and b) a minimization step for $x_{i,k}$ (see Steps 8) which originates a nonlinear dynamics.

Starting from a), in the first part of the proof we study the evolution of the consensus errors, i.e., the distance of $d_{i,k}$ and $\lambda_{i,k}$ from their respective network averages $\bar{d}_k = (1/N)\sum_{i=1}^N d_{i,k}$ and $\bar{\lambda}_k = (1/N)\sum_{i=1}^N \lambda_{i,k}$, and show that they evolve according to an asymptotically stable dynamics.

Next, we focus on the nonlinear update step in point b). By the optimality condition for the minimizer $x_{i,k+1}$ of Step 8, we can derive an inequality which relates the quantity $\|\bar{\lambda}_k - \lambda^\star\|^2$ (i.e., the dual optimality error of the network average $\bar{\lambda}_k$) across iterations with the consensus errors. Notably, if the consensus errors were equal to zero, then the inequality would result in the map identified by one iteration of Algorithm 1 being firmly quasi-non-expansive (Bauschke et al., 2011, Definition 4.1).

Since the distributed nature of Algorithm 1 gives rise to non-zero consensus errors, then, in the remainder of the proof, we study the interaction between the previously derived inequality and the linear system representing the evolution of the consensus errors. Specifically, we build a proper candidate storage function (satisfying suitable constraints) to show convergence of Algorithm 1 resorting to a Lyapunov approach.

4. NUMERICAL STUDY

In this section we provide a numerical example to corroborate the results in Theorem 1 and showcase the performance of Algorithm 1.

Consider a multi-agent linear program given by

$$\begin{aligned} \min_{x_1,\dots,x_N} \quad & \sum_{i=1}^{N} \gamma_i^\top x_i \\ \text{subject to:} \quad & \sum_{i=1}^{N} A_i x_i = b \\ & \mathrm{LB}_i \le x_i \le \mathrm{UB}_i \quad i \in \{1,\dots,N\}, \end{aligned} \tag{6}$$

with $N = 10$ (agents), $n_i = 2$ decision variables each, and $p = 3$ coupling constraints. We can generate an instance of the multi-agent program (6) as follows. For each agent $i = 1,\dots,N$, the coefficients γ_i of its linear cost function are drawn according to a uniform distribution in $[0, 10]^2$. The lower and upper bounds LB_i and UB_i on its local decision vector x_i are also drawn according to a uniform distribution over $[-15, -5]^2$ and $[5, 15]^2$, respectively. Each entry of the matrix A_i is drawn from a Gaussian distribution with zero mean and variance 100, and vector b is drawn according to a uniform distribution over $[0, 50]^3$. All extractions are performed independently.

The optimization problem (6) clearly fits the structure of $\mathcal{P}$ and therefore Algorithm 1 can be applied to compute an optimal solution in a distributed way.

The communication graph is generated at random according to the Erdos-Renyi model with probability 0.25. The positive-definite, doubly stochastic matrix $\mathcal{W}$ compliant with the adjacency matrix of the (undirected and connected) graph is obtained according to the procedure described in Sinkhorn and Knopp (1967).

We run Algorithm 1 for 5000 iterations using the following values for the penalty parameter: $c = 10^{-1}$, $c = 10^{-2}$, $c = 10^{-3}$, $c = 10^{-4}$, and $c = 10^{-5}$.

In Figure 1 we report, on a logarithmic scale, the relative error

$$\frac{|\sum_{i=1}^{N} \gamma_i^\top x_{i,k} - f^\star|}{|f^\star|}$$

between the cost achieved by the primal solution estimates and the optimal cost $f^\star$ computed by a centralized algorithm (upper plot), and the relative constraints violation

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

(lower plot), for all the different values of the penalty parameter c (different solid line colors). For comparison purposes we also report the fastest run of the algorithm proposed in Chang (2016) (dashed lines). As it can be seen from the picture, even though the behavior of the sequences is not monotonic, the cost error and constraint violation eventually converge to zero exponentially fast (linearly in logarithmic scale), until they hit the the solver numerical precision [2]. From the picture we can also see how the value of c affects the convergence rate, with $c = 10^{-1}$ being the slowest case and $c = 10^{-3}$ the fastest. Despite of the fact that the convergence rate is affected by c, the proposed Tracking-ADMM converges to an optimal solution of (6) in all cases. From the picture we can also see how for most values of c the proposed method outperforms the one in Chang (2016) in terms of convergence rate.

Finally, for $c = 10^{-3}$, we also plot in Figure 2 the quantities

$$\sqrt{\sum_{i=1}^{N} \|d_{i,k} - \bar{d}_k\|^2} \quad \text{and} \quad \sqrt{\sum_{i=1}^{N} \|\lambda_{i,k} - \bar{\lambda}_k\|^2},$$

which represent the norms of the consensus errors of $d_{i,k}$ and $\lambda_{i,k}$ with respect to their network averages $\bar{d}_k$ and $\bar{\lambda}_k$, respectively, for the first 1000 iterations. From the picture the reader can see how the agents eventually reach consensus both on $d_{i,k}$ and $\lambda_{i,k}$ also with an exponential rate.

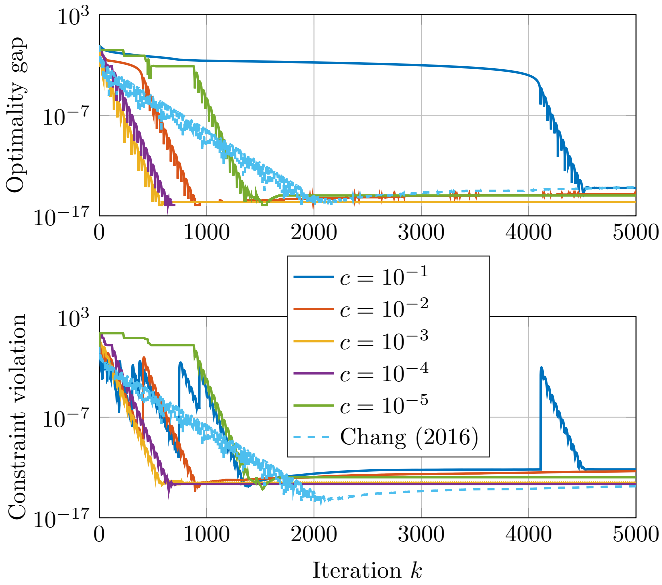

Fig. 1. Relative error between the cost achieved by the primal solution estimates and the optimal cost (upper plot) and relative constraint violation (lower plot), on a logarithmic scale, for different values of c of Tracking-ADMM (solid lines) and for the fastest run of Chang (2016) (dashed lines).

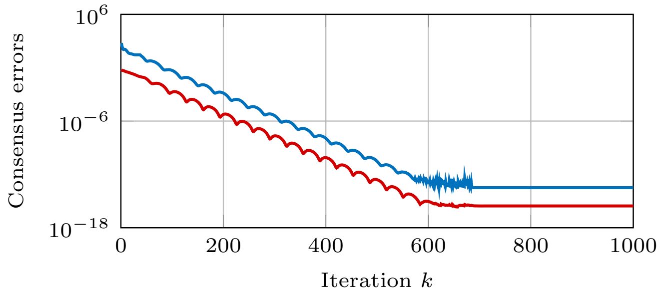

Fig. 2. Consensus error for $d_{i,k}$ (blue) and $\lambda_{i,k}$ (red) across the first 1000 iterations for the case $c = 10^{-3}$.

[2] Step 8 of Algorithm 1 is solved using the dual simplex optimizer of IBM ILOG CPLEX 12.9 with the minimum achievable tolerance of 10^{-9} both for optimality and feasibility.

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 vector) 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. Each node 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 estimate of the coupling-constraint violation. Numerical computations corroborated the theoretical results.

REFERENCES

Bauschke, H.H., Combettes, P.L., et al. (2011). *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer.

Bertsekas, D.P. and Tsitsiklis, J.N. (1989). *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ.

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

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

Chang, T.H., Nedić, A., and Scaglione, A. (2014). Distributed constrained optimization by consensus-based primal-dual perturbation method. *IEEE Trans. on Automatic Control*, 59(6), 1524–1538.

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

Duchi, J.C., Agarwal, A., and Wainwright, M.J. (2012). Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Trans. on Automatic Control*, 57(3), 592–606.

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

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

Iutzeler, F., Bianchi, P., Ciblat, P., and Hachem, W. (2016). Explicit convergence rate of a distributed alternating direction method of multipliers. *IEEE Trans. on Automatic Control*, 61(4), 892–904.

Jakovetić, D., Moura, J.M., and Xavier, J.M. (2015). Linear convergence rate of a class of distributed augmented lagrangian algorithms. *IEEE Trans. on Automatic Control*, 60(4), 922–936.

Johansson, B., Keviczky, T., Johansson, M., and Johansson, K.H. (2008). Subgradient methods and consensus algorithms for solving convex optimization problems. In *IEEE Conference on Decision and Control (CDC)*, 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., and Martinez, S. (2019). Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems Magazine*, 39(3), 40–72.

Liang, S., Yin, G., et al. (2020). Distributed smooth convex optimization with coupled constraints. *IEEE Transactions on Automatic Control*, 65(1), 347–353.

Ling, Q. and Ribeiro, A. (2014). Decentralized dynamic optimization through the alternating direction method of multipliers. *IEEE Trans. on Signal Processing*, 5(62), 1185–1197.

Makhdoumi, A. and Ozdaglar, A. (2017). Convergence rate of distributed ADMM over networks. *IEEE Trans. on Automatic Control*, 62(10), 5082–5095.

Margellos, K., Falsone, A., Garatti, S., and Prandini, M. (2018). Distributed constrained optimization and consensus in uncertain networks via proximal minimization. *IEEE Trans. on Automatic Control*, 63(5), 1372–1387.

Mateos-Núnez, D. and Cortés, J. (2017). Distributed saddle-point subgradient algorithms with Laplacian averaging. *IEEE Trans. on Automatic Control*, 62(6), 2720–2735.

Mota, J.F., Xavier, J.M., Aguiar, P.M., and Püschel, M. (2013). D-ADMM: A communication-efficient distributed algorithm for separable optimization. *IEEE Trans. on Signal Processing*, 61(10), 2718–2723.

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

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

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

Nedić, A., Ozdaglar, A., and Parrilo, P.A. (2010). Constrained consensus and optimization in multi-agent networks. *IEEE Trans. on Automatic Control*, 55(4).

Notarnicola, I. and Notarstefano, G. (2020). Constraint-coupled distributed optimization: a relaxation and duality approach. *IEEE Trans. on Control of Network Systems*, 7(1), 483–492.

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

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

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

Simonetto, A. and Jamali-Rad, H. (2016). Primal recovery from consensus-based dual decomposition for distributed convex optimization. *Journal of Optimization Theory and Applications*, 168(1), 172–197.

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

Varagnolo, D., Zanella, F., Cenedese, A., Pillonetto, G., and Schenato, L. (2016). Newton-Raphson consensus for distributed convex optimization. *IEEE Trans. on Automatic Control*, 61(4), 994–1009.

Xi, C., Xin, R., and Khan, U.A. (2018). ADD-OPT: Accelerated distributed directed optimization. *IEEE Trans. on Automatic Control*, 63(5), 1329–1339.

Xu, J., Zhu, S., Soh, Y.C., and Xie, L. (2018). Convergence of asynchronous distributed gradient methods over stochastic networks. *IEEE Trans. on Automatic Control*, 63(2), 434–448.

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

Zhang, Y. and Zavlanos, M.M. (2018). A consensus-based distributed augmented lagrangian method. In *2018 IEEE Conference on Decision and Control (CDC)*, 1763–1768.

Zhu, M. and Martínez, S. (2010). Discrete-time dynamic average consensus. *Automatica*, 46(2), 322–329.

Zhu, M. and Martínez, S. (2012). On distributed convex optimization under inequality and equality constraints. *IEEE Trans. on Automatic Control*, 57(1), 151–164.