Distributed algorithms for convex network optimization under non-sparse equality constraints

Ashish Cherukuri Jorge Cortés

***Abstract*— This paper studies a class of network optimization problems where the objective function is the summation of individual agents' convex functions and their decision variables are coupled by linear equality constraints. These constraints are not sparse, meaning that they do not match the pattern of the network adjacency matrix. We propose two approaches to design efficient distributed algorithms to solve the network optimization problem. Our first approach consists of transforming the non-sparse equality constraints into sparse ones by increasing the number of the agents' decision variables, yielding an exact reformulation of the original optimization problem. We discuss two reformulations, based on the addition of consensus variables and of constraint-mismatch variables, and discuss the scalability of the strategies resulting from them. Our second approach consists instead of sparsifying the non-sparse constraints by zeroing some coefficients, yielding an approximate reformulation of the original problem. We formally characterize the gap on the distance between the optimizers of the original and approximated problems as a function of the number of entries made zero in the constraints. Various simulations illustrate our results.**

I. Introduction

An increasing body of work deals with the design and analysis of distributed algorithms to solve constrained convex optimization problems for networks. These works rely on versatile algorithm design methods, such as primal-dual algorithms [1] or the alternating direction method of multipliers [2], to synthesize distributed strategies provided that the optimization problem is locally expressible. By locally expressible we mean that the objective function is the aggregate of functions available to the agents that depend on their state and those of their neighbors, and that each constraint can be evaluated with knowledge of the state of a given specific agent and those of its neighbors. The above mentioned design methods do not result in a distributed algorithm when the optimization problem is instead global in nature. This observation provides the motivation for our work here on solution approaches to overcome the challenges posed by global constraints.

Literature review

The use of distributed algorithms to solve network optimization problems span multiple areas, such as power networks [3], communication networks [4], and transportation networks [5], to name only a few. While many network optimization problems have a locally expressible structure, there are others that have a global nature, such as economic dispatch problem [6] and its variants [7]. Many distributed algorithms exist for constrained network optimization problems, see e.g., [8], [9], [10], [11] and references therein. In these works, either the constraints can be expressed locally or the size of the optimization variable is independent of the network size. In contrast, both conditions are not valid for the class of network optimization problems we consider here. The technical approach for our approximate formulations of the optimization problem using constraint sparsification is related to the vast literature on optimization problems with perturbations, see e.g. [12], and references therein. Most of these works establish analytical properties of the optimizer map when the optimization data is perturbed. In contrast, we derive quantifiable bounds on error in optimizers under perturbation to the feasibility set. The closest related work along these lines is [13], which derives error bounds for conic constraints.

Statement of contributions

Our starting point is the definition of a general class of convex optimization problems over networks. The objective function for this problem is the aggregate of local objective functions, each belonging to an agent, and the constraints include equalities that are non-sparse, meaning that their evaluation requires state information beyond that provided by neighboring agents. We propose two approaches to tackle the problem of designing distributed algorithms to solve the optimization problem. We assume any distributed algorithm to satisfy two requirements: (i) each agent can only communicate with its neighboring agents in the network; and, (ii) each agent knows and does not share with its neighbors the data about the constraints it is involved in, albeit it need not have access to states of all gents involved in the said constraints as the constraints are non-sparse.

Our first approach is to develop exact reformulations of the optimization in terms of problem data that is locally expressible. Within this approach, we present two exact reformulations. The first reformulation is based on ideas of consensus where each agent maintains an estimate of the whole network state, making sure that its estimate satisfies the constraints that are known to it while at the same time seeking to agree with its neighbors on a common value for it. While the reformulation has constraints that are locally expressible, any distributed strategy that solves this reformulation lacks scalability. This is because the state of each agent is of the same size as the network's and

Ashish Cherukuri and Jorge Cortés are with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, {acheruku,cortes}@ucsd.edu.

so, the size of the interchanged messages in the algorithm grows with the number of agents. This motivates the second reformulation, which is based on the introduction of constraint-mismatch variables in the state of each agent, one per constraint. We show that this reformulation is exact and results in agents having states whose dimension scales with the number of constraints. Therefore, distributed strategies for this reformulation are scalable, provided the number of constraints is independent of the network size.

Finally, for the case when this is not the case, we propose the second approach that makes the constraint sparse by zeroing the entries of the matrix that defines it. As a result of this process, the optimizer of the problem and its optimal value change. We provide results that characterize the size of this error in terms of the perturbation of the affine constraints due to the sparsification process. The first result here considers a general convex optimization problem with a convex compact constraint set and analyzes the affect of perturbation to this set on the optimizer. The error bound for this case is non-Lipschitzian. In the second result, we consider affine constraint and using the KKT conditions of the original and the perturbed problem, we provide a Lipschitz error bound on the distance between the optimizers of the original and the perturbed problem. Various simulations illustrate our results.

Organization

The paper is organized as follows. Section II presents the notation and basic concepts on graph theory. Section III describes the non-sparse network convex optimization problem and the requirements on the distributed algorithm to solve it. Section IV proposes two exact reformulations based on consensus variables and the introduction of constraint-mismatch variables, respectively. Section V instead describes the approach based on sparsifying the non-sparse constraints and characterizes the distance between the optimizers of the original and the approximated problem. Finally, Section VI describes our conclusions and ideas for future work.

II. PRELIMINARIES

This section introduces the notation and basic notions on algebraic graph theory used throughout the paper.

A. Notation

Let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$ and $\mathbb{Z}_{\geq 1}$ denote the set of real, real non-negative, and positive integer numbers, respectively. For a vector $x \in \mathbb{R}^n$, the i-th component is $x_i \in \mathbb{R}$. For two vectors $x, y \in \mathbb{R}^n$, $x \leq y$ is equivalent to $x_i \leq y_i$ for all $i \in \{1, \dots, n\}$. We follow a similar convention for strict inequality. The vector of all zeros and all ones of size n are denoted as $\mathbf{0}_n$ and $\mathbf{1}_n$, respectively. For two vectors $u \in \mathbb{R}^n$, $v \in \mathbb{R}^m$, the vector $(u; v) \in \mathbb{R}^{n+m}$ represents stacking them one after another. The cardinality of a set $\mathcal{S}$ is given by $|\mathcal{S}|$. For a matrix $A \in \mathbb{R}^{n \times m}$, the i-th row and the (i, j)-th element are denoted as $[A]_i$ and $[A]_{i,j}$, respectively. For $A \in \mathbb{R}^{n_1 \times n_2}$ and $B \in \mathbb{R}^{m_1 \times m_2}$, $A \otimes B \in \mathbb{R}^{n_1 m_1 \times n_2 m_2}$ is the Kronecker product.

B. Graph theory

Here we give a brief overview of necessary concepts from algebraic graph theory following [14]. A *(weighted) undirected graph* is a triplet $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathsf{A})$, where $\mathcal{V}$ is a finite set called the *vertex set*, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is called the *edge set* where $(i, j) \in \mathcal{E}$ if and only $(j, i) \in \mathcal{E}$, and $\mathsf{A} \in \mathbb{R}_{\geq 0}^{|\mathcal{V}| \times |\mathcal{V}|}$ the called the *adjacency matrix*. The set $\mathcal{N}_i \subset \mathcal{V}$ denotes the set of neighbors of a vertex i, that is, $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$. The adjacency matrix satisfies the property $[\mathsf{A}]_{i,j} = [\mathsf{A}]_{j,i} > 0$ if $(i, j) \in \mathcal{E}$ and $[\mathsf{A}]_{i,j} = 0$, otherwise. The positive value $[\mathsf{A}]_{i,j}$ for some $(i, j) \in \mathcal{E}$ is called the edge-weight of (i, j). The *weighted degree* of a vertex i is $d(i) = \sum_{j=1}^{n} [\mathsf{A}]_{i,j}$. The *weighted degree matrix* D is the diagonal matrix defined by $[\mathsf{D}]_{i,i} = d(i)$, for all $i \in \{1, \dots, n\}$. The *Laplacian* matrix is $\mathsf{L} = \mathsf{D} - \mathsf{A}$. Note that L is symmetric and satisfies $\mathsf{L}\mathbf{1}_n = 0$. A path is an ordered sequence of vertices such that two subsequent vertices form an edge. The undirected graph is *connected* if there is path between any two vertices of the graph. The Laplacian of a connected graph has 0 as a simple eigenvalue, with all other eigenvalues positive.

III. PROBLEM STATEMENT

Consider a network of $n \in \mathbb{Z}_{\geq 1}$ agents whose communication topology is represented by a connected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathsf{A})$. An edge $(i, j) \in \mathcal{E}$ represents two-way communication between agents i and j. Each agent i is associated with a decision variable $x_i \in \mathbb{R}$ and a cost function $f_i : \mathbb{R} \to \mathbb{R}$, which we assume convex and continuously differentiable. Consider the following *network optimization problem* for the group of agents

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^{n} f_i(x_i), \tag{1a}$$

$$\text{subject to} \quad Ax = b, \tag{1b}$$

$$x^m \leq x \leq x^M, \tag{1c}$$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $x^m, x^M \in \mathbb{R}^n$ satisfy $0 < x_i^m < x_i^M < \infty$ for all $i \in \{1, \dots, n\}$. We denote the set of feasible points and the set of optimizers of (1) by $\mathcal{F}$ and $\mathcal{F}^*$, respectively.

In problem (1), the objective function has a separable structure, that is, it is the summation of the individual agents' objectives, each depending on its own decision variable. However, we do not assume any sparsity structure for the matrix A and so the constraint (1b) is (possibly) global in nature, coupling the decision variables of all agents. Formally, a function $g : \mathbb{R}^n \to \mathbb{R}^m$ is *local* with respect to the graph $\mathcal{G}$ if for each $k \in \{1, \dots, m\}$ there exists $i \in \{1, \dots, n\}$ such that g_k is a function of $(x_i, \{x_j\}_{j \in \mathcal{N}_i})$. A constraint is local with respect to the graph $\mathcal{G}$ if the function defining it is local. Otherwise, the constraint is global. In words, each component of a local constraint depends only on the decision variables of some agent and its neighbors. While (1c) is a local constraint, (1b) need not be.

Our objective is to design distributed algorithms that allow the agents to find the optimizer of the problem (1). Informally, by distributed we mean that there is no central computing entity and instead each agent seeks to determine its decision variable at the global optimum of (1) by communicating with neighboring agents. Formally, a *distributed algorithm* has the following properties

(i) *local exchange of information*: each agent i can only communicate with its neighbors $\mathcal{N}_i$ in the graph $\mathcal{G}$; and
(ii) *private information*: each agent i only knows its cost function f_i, the min- and max-limits x_i^m and x_i^M, and the constraint components $([A]_k, b_k)$ for all $k \in \{1, \dots, m\}$ such that $[A]_{k,i} \neq 0$ (i.e., where its state is involved). This is private information for i that is not shared with its neighbors.

Distributed algorithms to solve convex optimization problems defined by separable objective functions and local constraints can be designed by using a variety of methods, such as primal-dual dynamics or alternating direction method of multipliers. However, such methods do not yield distributed strategies for problems of the form (1) because of the non-sparsity of the constraints. To tackle this problem, we propose two alternative approaches: exact reformulations and constraint sparsification.

We end this section with a motivating optimization problem from power systems that includes global affine constraints.

Example 3.1: (Economic dispatch problem: definition): Consider $n \in \mathbb{Z}_{\geq 1}$ power generators communicating over a connected weighted graph $\mathcal{G}$. Each generator $i \in \{1, \dots, n\}$ has a convex, continuously differentiable, cost function $f_i : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ associated with the power generation: given the power generated by i as $P_i \in \mathbb{R}_{\geq 0}$, the cost incurred by i is $f_i(P_i)$. The objective for the generators is to collectively meet a load $P_l \in \mathbb{R}_{\geq 0}$ while minimizing the total incurred cost of generation and satisfying the min- and max- generation constraint for each generator, that is, $P_i^m \leq P_i \leq P_i^M$ for all $i \in \{1, \dots, n\}$. Formally,

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^n f_i(P_i), && \text{(2a)} \\ \text{subject to} \quad & \sum_{i=1}^n P_i = P_l, && \text{(2b)} \\ & P^m \leq P \leq P^M. && \text{(2c)} \end{aligned}$$

Notice that the load balance condition (2b) is a global constraint. In our previous work [6], we have given a distributed algorithmic solution that finds the optimizer of the economic dispatch problem, where we use the special structure of the equality constraint to devise the solution strategy. In the following section, we give an exact reformulation that leads to the design of a distributed primal-dual algorithm. •

IV. Exact formulation as network optimization with local constrains

Here we develop two equivalent reformulations of the network optimization problem (1) such that the global constraint is expressed as a collection of local constraints. In both cases, the reformulation involves the introduction of additional variables.

A. Reformulation using consensus

The basic idea behind the reformulation using consensus is to have each agent maintain an estimate of the whole network state, rewrite the global constraint as a collection of local constraints that can be expressed in terms of these estimates, and introduce additional consensus constraints to make sure all agents agree on the same network state. Consequently, for each $i \in \{1, \dots, n\}$, we define a decision variable $\hat{x}^i \in \mathbb{R}^n$. We let $\hat{x} = (\hat{x}^1; \hat{x}^2; \dots; \hat{x}^n) \in (\mathbb{R}^n)^n$ be the collection of all decision variables. We let $\tilde{A}^i$ and $\tilde{b}^i$ denote the submatrices formed by the collection of rows $k \in \{1, \dots, m\}$ of A and b, respectively, for which $[A]_{k,i} \neq 0$. Recall that under the network model described in Section III, each agent i knows $(\tilde{A}^i, \tilde{b}^i)$. Consider the following optimization problem

$$\begin{aligned} \underset{\hat{x}}{\text{minimize}} \quad & \sum_{i=1}^n f_i(\hat{x}_i^i), && \text{(3a)} \\ \text{subject to} \quad & \tilde{A}^i \hat{x}^i = \tilde{b}^i, \ \forall i, && \text{(3b)} \\ & x_i^m \leq \hat{x}_i^i \leq x_i^M, \ \forall i, && \text{(3c)} \\ & (\mathsf{L} \otimes I_n)\hat{x} = \mathbf{0}_{n^2}. && \text{(3d)} \end{aligned}$$

Note that constraints (3b) and (3c) are local because $\hat{x}^i$ is the decision variable of agent i. Further, the constraint (3d) is local due to the sparsity structure of the Laplacian. For convenience, we denote the set of feasible points of (3) by $\mathcal{F}_c \subset \mathbb{R}^{n^2}$ and the set of optimizers by $\mathcal{F}_c^*$. The following result establishes the bijection between the optimizers of (1) and (3).

Proposition 4.1: (Exact reformulation using consensus): For problems (1) and (3), we have $(\mathcal{F}^*)^n = \mathcal{F}_c^*$.

Proof: Since $\mathcal{G}$ is connected, one has that $(\mathsf{L} \otimes I_n)\hat{x} = \mathbf{0}_{n^2}$ if and only if $\hat{x} = \mathbf{1}_n \otimes x$, with $x \in \mathbb{R}^n$. For $\hat{x} \in \mathcal{F}_c$, this fact together with (3b) implies that $Ax = b$ and $x^m \leq x \leq x^M$, i.e., $x \in \mathcal{F}$. Conversely, if $x \in \mathcal{F}$, then one can show that $\mathbf{1}_n \otimes x \in \mathcal{F}_c$, and the statement follows. ■

The appealing feature of problem (3) is that its objective function has a separable structure and, at the same time, the constraints are local, making it amenable to the design of distributed algorithms. However, the variable that each agent maintains has dimension n, the size of the network. This means that in order to execute any such distributed strategy, agents need to either communicate messages of order n or schedule n communication rounds to transmit messages of order 1. In either case, the volume of communication required by the algorithm executions does not scale well with the network size.

B. Reformulation using constraint-mismatch variables

The basic idea behind the reformulation using constraint-mismatch variables is to add one variable per non-sparse constraint to the state of each agent. In turn, this allows each agent to compute a proxy of the contribution of other agents to the satisfaction of that non-sparse constraint in a way that collectively makes their original state satisfy it.

We start with some notation. For each $k \in \{1, \dots, m\}$, let the vector $e^k \in \mathbb{R}^n$ be such that $e^k_i = 1$ if $[A]_{k,i} \neq 0$ and $e^k_i = 0$ otherwise. That is, e^k encodes agents whose decision variables are coupled in the k-th component of the constraint (1b). Note that e^k is known to each agent involved in the k-th constraint as each of these agents know $([A]_k, b_k)$ under the network model described in Section III. Corresponding to each $k \in \{1, \dots, m\}$, define a variable $y^k \in \mathbb{R}^n$ and let the decision variable for each agent $i \in \{1, \dots, n\}$ be $(x_i, \{y^k_i\}_{k=1}^m)$. Consider the following optimization problem

$$\underset{x, \{y^k\}_{k=1}^m}{\text{minimize}} \quad \sum_{i=1}^n f_i(x_i), \tag{4a}$$

$$\text{subject to} \quad \text{diag}([A]_k)x + \mathsf{L}y^k = \frac{b_k}{\mathbf{1}_n^\top e^k} e^k, \ \forall k, \tag{4b}$$

$$x^m \leq x \leq x^M. \tag{4c}$$

Note that the structure of the Laplacian and the way we have partitioned the auxiliary variables $\{y^k\}$ into each agents' decision variable makes the constraints in (4b) local. In fact, we can rewrite them as

$$[A]_{k,i}x_i + \sum_{j \in \mathcal{N}_i} [\mathsf{A}]_{i,j}(y^k_i - y^k_j) = \frac{b_k}{\mathbf{1}_n^\top e^k} e^k_i,$$

for all $k \in \{1, \dots, m\}$ and $i \in \{1, \dots, n\}$. For convenience, we denote the set of feasible points of the above optimization problem by $\mathcal{F}_r \subset \mathbb{R}^n \times \mathbb{R}^{nm}$ and its set of optimizers by $\mathcal{F}^*_r$. The next result establishes the bijection between the optimizers of (1) and (4).

Proposition 4.2: (Exact reformulation using constraint-mismatch variables): For problems (1) and (3), we have $\mathcal{F}^* = \Pi_x(\mathcal{F}^*_r)$, where Π_x is the projection onto the first n coordinates.

Proof: Note that it is enough to show $\Pi_x(\mathcal{F}^r) = \mathcal{F}$ because the objective function of (1) and (4) are same. Let $x \in \mathcal{F}$. Clearly, x satisfies (4c). Since x satisfies (1b), we get

$$\mathbf{1}_n^\top \Big(\text{diag}([A]_k)x - \frac{b_k}{\mathbf{1}_n^\top e^k} e^k \Big) = 0,$$

for all $k \in \{1, \dots, m\}$. This implies that $\text{diag}([A]_k)x - \frac{b_k}{\mathbf{1}_n^\top e^k} e^k$ belongs to the range space of L and so there exists $y^k \in \mathbb{R}^n$ such that

$$-\mathsf{L}y^k = \text{diag}([A]_k)x - \frac{b_k}{\mathbf{1}_n^\top e^k} e^k.$$

Collecting these y^k vectors, we get $(x, \{y^k\}_{k=1}^m) \in \mathcal{F}^r$. Thus, $x \in \Pi_x(\mathcal{F}^r)$ and so, $\mathcal{F} \subseteq \Pi_x(\mathcal{F}^r)$. Now let $x \in \Pi_x(\mathcal{F}^r)$. Then, x satisfies (1c). Further, there exists $\{y^k\}_{k=1}^m$ such that $(x, \{y^k\}_{k=1}^m)$ satisfy (4b). Pre-multiplying each equation of (4b) with $\mathbf{1}_n^\top$ yields $[A]_k x = b_k$ for all $k \in \{1, \dots, m\}$. That is, x satisfies (1b). Therefore,, $x \in \mathcal{F}$ and so $\Pi_x(\mathcal{F}^r) \subseteq \mathcal{F}$. This concludes the proof. ■

Distributed algorithms can be designed for (4) as the objective is separable and the constraints are local. There is however a key difference between reformulations (3) and (4), which is the size of the decision variables for each agent. In the former, this size is n while in the later it is $m+1$. Assuming that the time and communication complexity of a distributed algorithm increases with the size of the decision variables for each agent, the second formulation has therefore a clear advantage.

Remark 4.3: (Dual problem): Here we discuss the nature of the dual problem of (1) and explain why it is difficult to design a distributed algorithm for it. The Lagrangian of (1) is

$$L(x, \lambda^m, \lambda^M, \nu) = \sum_{i=1}^n f_i(x_i) + \nu^\top (Ax - b) + (\lambda^m)^\top (x^m - x) + (\lambda^M)^\top (x - x^M),$$

where $\lambda^m, \lambda^M \in \mathbb{R}^n_{\geq 0}$ and $\nu \in \mathbb{R}^m$ are Lagrange multipliers corresponding to the constraints (1b) and (1c), respectively. Let $D : \mathbb{R}^m \times \mathbb{R}^{2n}_{\geq 0} \to \mathbb{R}$ be the objective function of the dual problem. Then,

$$D(\lambda^m, \lambda^M, \nu) = \min_{x \in \mathbb{R}^n} L(x, \lambda^m, \lambda^M, \nu). \tag{5}$$

The dual problem is given as

$$\underset{\lambda^m, \lambda^M, \nu}{\text{maximize}} \quad D(\lambda^m, \lambda^M, \nu), \tag{6a}$$

$$\text{subject to} \quad \lambda^m, \lambda^M \geq \mathbf{0}_n. \tag{6b}$$

Since the constraints of the primal problem (1) are affine, the refined Slater condition is satisfied and so the duality gap between the primal and the dual problems is zero [15]. Note that the constraints of this dual problem are local. However, the objective function does not have the separable form as in (1) and so the formulation (6) is not amenable for distributed algorithms. To see why, we first define a partition of $(\lambda^m, \lambda^M, \nu)$ into decision variables of each of the agents. Let λ^m_i and λ^M_i be part of the decision variables of agent i. Further, assign ν_k, $k \in \{1, \dots, m\}$, to some agent i_k that has information of $([A]_k, b_k)$ (that is, $[A]_{k,i_k} \neq 0$ for all k). Note that the dual function $D(\lambda^m, \lambda^M, \nu) = L(x^*(\lambda^m, \lambda^M, \nu), \lambda^m, \lambda^M, \nu)$, where $x^*(\lambda^m, \lambda^M, \nu)$ is the minimizer of the convex function $L(\cdot, \lambda^m, \lambda^M, \nu)$. Thus, we get $\nabla_x L(x^*(\lambda^m, \lambda^M, \nu), \lambda^m, \lambda^M, \nu) = \mathbf{0}_n$ from first-order optimality condition. That is,

$$\begin{bmatrix} \nabla f_i(x_i^*(\lambda^m, \lambda^M, \nu)) \\ \vdots \\ \nabla f_n(x_n^*(\lambda^m, \lambda^M, \nu)) \end{bmatrix} + A^\top \nu - \lambda^m + \lambda^M = \mathbf{0}_n.$$

For simplicity, assume that the map ∇f_i is invertible for each i. Then, one can find $x_i^*(\lambda^m, \lambda^M, \nu)$, for all i, from the above equation and use it to obtain the dual objective function D. Following the algebra, one can show that $\frac{\partial D}{\partial \lambda_i}$ depends on ν_k for all k such that $[A]_{k,i} \neq 0$, and therefore one cannot express D as an addition of individual objective function of agents.

The above discussion shows that it is not straightforward to design a distributed algorithm for the dual problem (6). However, this does not completely rule out the possibility of coming up with a different partition for variables $(\lambda^m, \lambda^M, \nu)$ or a reformulation of (6) such that the resulting optimization has a separable cost and local constraints. •

Note that the two reformulations presented in this section can also be carried out, in a similar fashion, when the global constraint (1b) is an inequality instead of an equality. This would entail replacing the equality in (3b) and (4b) with inequalities.

Example 4.4: (Economic dispatch problem: comparison between reformulations): Consider the economic dispatch problem described in Example 3.1. For simplicity, assume that the inequality constraints (2c) are absent and the cost function for each generator i is $f_i(P_i) = c_i P_i^2$, $c_i > 0$. Here, we compare the two reformulations (3) and (4) as the size of the network increases. To do this, we use the same distributed algorithm design approach for each of the reformulations and compare the performance of the resulting strategies on the basis of two metrics. The first comparison metric is the number of iterations each algorithm takes to converge to an optimizer of that particular reformulation. The second metric is the volume of the message passed during each iteration of the algorithm (where the volume is measured in terms of the number of real variables interchanged). To perform this comparison, we select the primal-dual dynamics for the Lagrangian and the primal-dual dynamics for the augmented Lagrangian as the candidate distributed algorithm for reformulation (4) and (3), respectively. We consider four networks with number of generators 5, 15, 25, and 35. For each network, the cost coefficients are selected randomly in the interval $(0, 1]$ and the power demand for a network of size n is $n/2$. Figure 1 summarizes the results, showing how the formulation based on consensus does not scale well with the network size, while the formulation based on constraint-mismatch variables does. Note that the simulation results depend on the selection of distributed algorithm for each formulation, the choice of stepsize, and the communication graph. •

V. Approximate formulation via constraint sparsification

In this section we study an alternative approach to deal with network optimization problems with non-sparse constraints of the form (1). This approach is motivated by the observation that, in scenarios where the number m of constraints is of the same size as the number n of agents in the network, distributed algorithm designs resulting from exact reformulations are not scalable with the network size. For such scenarios, we propose to sparsify the constraints by zeroing entries in the rows of the matrix A in (1b). Formally, we consider the following perturbed version of (1)

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^{n} f_i(x_i), \tag{7a}$$

$$\text{subject to} \quad (A + A_p)x = b + b_p, \tag{7b}$$

$$x^m \leq x \leq x^M, \tag{7c}$$

where for generality, we also consider perturbation in the right-hand side of (7b). This process naturally results in a different optimizer and a different optimal value. Our aim in this section is to provide upper bounds between the optimizers of the original and the perturbed problems as a function of the constraint sparsification. Note that an optimizer of the perturbed problem (7) might not be a feasible point of the original problem (1). Therefore, when feasibility is more important than optimality, one could possibly additionally modify the objective function in (7) (while maintaining the separable structure) so that the distance of the optimizer of the perturbed problem to the original feasible set is minimized. Alternatively, one could neglect the infeasibility in cases where the decision variables of the agents are states of a network dynamics and the feasibility of constraints is ensured by the dynamics itself, such as for instance in power networks, where load satisfaction is ensured by primary and secondary controllers [16].

Our first result provides one such upper bound for a general convex optimization problem.

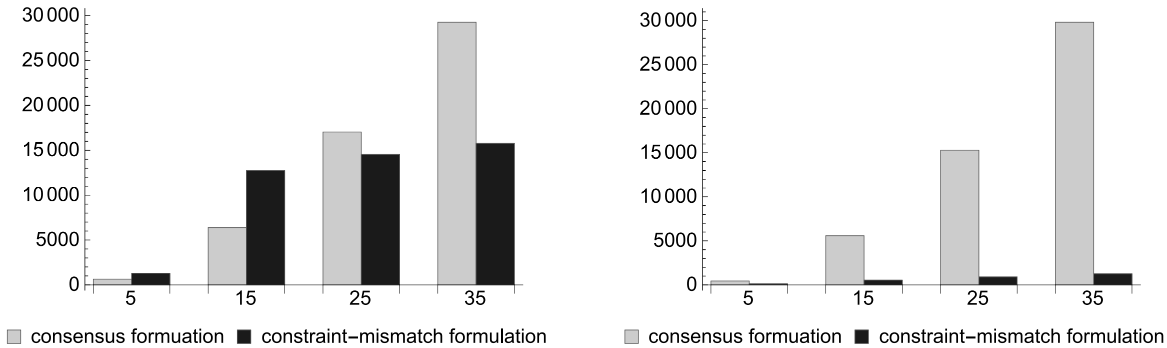
Proposition 5.1: (Perturbation to general convex optimization problem: upper bound between optimizers): Consider the two optimizations on $\mathbb{R}^n$,

$$\min\{f(x) \mid x \in \mathcal{F}_1\}, \tag{8a}$$

$$\min\{f(x) \mid x \in \mathcal{F}_2\}. \tag{8b}$$

where $\mathcal{F}_1$ and $\mathcal{F}_2$ are compact subsets of $\mathbb{R}^n$. Assume that $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable, strongly convex, satisfying $mI \preceq \nabla^2 f(x) \preceq MI$ for all $x \in \mathbb{R}^n$ and some constants $0 < m < M < \infty$. Let $x_1^*, x_2^* \in \mathbb{R}^n$ be the optimizers of (8a) and (8b), respectively. Then,

$$\|x_1^* - x_2^*\| \leq \sqrt{\frac{3}{2m}} \Big(M d(\mathcal{F}_1, \mathcal{F}_2)^2 + (G_1 + G_2) d(\mathcal{F}_1, \mathcal{F}_2) \Big)^{\frac{1}{2}} + d(\mathcal{F}_1, \mathcal{F}_2).$$



(a) Number of iterations

(b) Communication volume per iteration

Fig. 1. Illustration comparing the performance of candidate distributed algorithms for the consensus-based reformulation (3) and the constraint-mismatch reformulation (4) of the network optimization problem. For each network size, the initial condition is at the origin. The communication graph is the same for both algorithms. The step size is constant and the optimization is stopped once the distance to the optimizer decreases to 0.1% of the initial value. Plot (a) shows the number of iterations to convergence for the algorithms for different network size. For small networks, the algorithm for the consensus-based reformulation (3) is faster. However, the algorithm for the constraint-mismatch variable-based reformulation (4) starts outperforming it as the network size increases. Plot (b) shows the volume of message passed per each iteration of the algorithms, where volume refers to the total number of real-valued variables.

where

$$d(\mathcal{F}_1,\mathcal{F}_2) = \max\{\sup_{x\in\mathcal{F}_1}\inf_{y\in\mathcal{F}_2}\|x-y\|, \sup_{x\in\mathcal{F}_2}\inf_{y\in\mathcal{F}_1}\|x-y\|\},$$

and $G_1 = \max\{\|\nabla f(x)\| \mid x \in \mathcal{F}_1\}$, $G_2 = \max\{\|\nabla f(x)\| \mid x \in \mathcal{F}_2\}$.

Proof: Let $y_1 \in \mathcal{F}_1$ and $y_2 \in \mathcal{F}_2$ be such that

$$\|y_1 - x_2^*\| = \min_{y\in\mathcal{F}_1}\|y - x_2^*\|,$$
$$\|y_2 - x_1^*\| = \min_{y\in\mathcal{F}_2}\|y - x_1^*\|.$$

Then, by definition

$$\|y_1 - x_2^*\| \le d(\mathcal{F}_1,\mathcal{F}_2), \tag{9a}$$
$$\|y_2 - x_1^*\| \le d(\mathcal{F}_1,\mathcal{F}_2). \tag{9b}$$

The second-order condition for strong convexity gives

$$f(y_1) \ge f(x_1^*) + \nabla f(x_1^*)^\top (y_1 - x_1^*) + \frac{m}{2}\|y_1 - x_1^*\|^2.$$

Since x_1^* is the optimizer of f on $\mathcal{F}_1$ and $y_1 \in \mathcal{F}_1$, we deduce that $\nabla f(x_1^*)^\top (y_1 - x_1^*) \ge 0$. Employing this fact in the above inequality yields

$$f(y_1) \ge f(x_1^*) + \frac{m}{2}\|y_1 - x_1^*\|^2. \tag{10}$$

Note that from the second-order condition we also have

$$f(y_1) \le f(x_2^*) + \nabla f(x_2^*)^\top (y_1 - x_2^*) + \frac{M}{2}\|y_1 - x_2^*\|^2.$$

Using the above expression and (10) we get

$$f(x_1^*) + \frac{m}{2}\|y_1 - x_1^*\|^2 \le f(x_2^*) + \nabla f(x_2^*)^\top (y_1 - x_2^*) + \frac{M}{2} d(\mathcal{F}_1,\mathcal{F}_2)^2, \tag{11}$$

where we have used (9b) to bound $\|y_1 - x_2^*\|$. Similar line of reasoning gives us the following bound

$$f(x_2^*) + \frac{m}{2}\|y_2 - x_2^*\|^2 \le f(x_1^*) + \nabla f(x_1^*)^\top (y_2 - x_1^*) + \frac{M}{2} d(\mathcal{F}_1,\mathcal{F}_2)^2. \tag{12}$$

Adding (11) and (12), we obtain

$$\frac{m}{2}\Big(\|y_1 - x_1^*\|^2 + \|y_2 - x_2^*\|^2\Big) \le M d(\mathcal{F}_1,\mathcal{F}_2)^2 + \nabla f(x_2^*)^\top (y_1 - x_2^*) + \nabla f(x_1^*)^\top (y_2 - x_1^*). \tag{13}$$

Note that

$$a + b = (a^2 + b^2 + 2ab)^{1/2} \le (a^2 + b^2 + 2a^2 + 2b^2)^{1/2} = \sqrt{3}(a^2 + b^2)^{1/2}.$$

Therefore,

$$\|y_1 - x_1^*\| + \|y_2 - x_2^*\| \le \sqrt{3}\Big(\|y_1 - x_1^*\|^2 + \|y_2 - x_2^*\|^2\Big)^{1/2}.$$

Further, using (13) in the above expression results into

$$\|y_1 - x_1^*\| + \|y_2 - x_2^*\| \le \sqrt{\frac{6}{m}}\Big(M d(\mathcal{F}_1,\mathcal{F}_2)^2 + \nabla f(x_2^*)^\top (y_1 - x_2^*) + \nabla f(x_1^*)^\top (y_2 - x_1^*)\Big)^{1/2}.$$

Finally,

$$\begin{aligned}\|x_1^* - x_2^*\| &= \|\frac{1}{2}(x_1^* - y_1 + y_1 - x_2^*) + \frac{1}{2}(x_1^* - y_2 + y_2 - x_2^*)\|,\\ &\le \frac{1}{2}(\|y_1 - x_1^*\| + \|y_2 - x_2^*\|) + \frac{1}{2}(\|y_1 - x_2^*\| + \|y_2 - x_1^*\|),\\ &\le \frac{1}{2}\sqrt{\frac{6}{m}}\Big(M d(\mathcal{F}_1,\mathcal{F}_2)^2 + \nabla f(x_2^*)^\top (y_1 - x_2^*) + \nabla f(x_1^*)^\top (y_2 - x_1^*)\Big)^{1/2} + d(\mathcal{F}_1,\mathcal{F}_2).\end{aligned}$$

Using bounds on gradients, we obtain

$$\|x_1^* - x_2^*\| \leq \sqrt{\frac{3}{2m}}\Big(Md(\mathcal{F}_1,\mathcal{F}_2)^2 + (G_1+G_2)d(\mathcal{F}_1,\mathcal{F}_2)\Big)^{1/2} + d(\mathcal{F}_1,\mathcal{F}_2).$$

This completes the proof. ■

In the above result, one can take $\mathcal{F}_1$ as the feasibility set of the original optimization problem and $\mathcal{F}_2$ as its perturbed version. Since these sets are fairly general, the result has a broader scope of application. However, as a result, the error bound we obtain on the optimizers is conservative. Instead, the next result focuses on a much more specific class of optimization problems (quadratic with affine constraints) and establishes a Lipschitz bound on the distance between optimizers by making use of the affine nature of constraints.

Proposition 5.2: (Perturbation to quadratic programming with affine constraints: Lipschitz upper bound between optimizers): Consider the following optimization problems for $x \in \mathbb{R}^n$,

$$\min\{\|x - x_0\|^2 \mid A_1 x = b_1\}, \tag{14a}$$

$$\min\{\|x - x_0\|^2 \mid A_2 x = b_2\}. \tag{14b}$$

where $x_0 \in \mathbb{R}^n$, $A_1, A_2 \in \mathbb{R}^{m\times n}$, and $b_1, b_2 \in \mathbb{R}^m$. Let A_1 and A_2 have full row-rank. Denote $x_1^*, x_2^* \in \mathbb{R}^n$ to be the optimizers of (14a) and (14b), respectively. Then,

$$\|x_1^* - x_2^*\| \leq \alpha\|A_1 - A_2\| + \beta\|b_1 - b_2\|,$$

where

$$\begin{aligned}
&\alpha = (\|x_0\| + \|b_2\|)\tilde{\alpha},\\
&\tilde{\alpha} = \|A_1^\top(A_1A_1^\top)^{-1}\| + \|A_2\|\Big(\|(A_1A_1^\top)^{-1}\|\\
&\qquad + \|A_2\|\|(A_2A_2^\top)^{-1}\|\|(A_1A_1^\top)^{-1}\|(\|A_1\| + \|A_2\|)\Big),\\
&\beta = \|A_1^\top(A_1A_1^\top)^{-1}\|.
\end{aligned}$$

Proof: Consider (14a). The Lagrangian is

$$L_1(x,\lambda) = \|x - x_0\|^2 + \lambda^\top(A_1x - b_1),$$

where $\lambda \in \mathbb{R}^m$ is the Lagrange multiplier. Since the constraint is linear, the refined Slater condition is satisfied for this problem and so a primal-dual optimizer (x_1^*, λ_1^*) of (14a) satisfies the following Karush-Kuhn-Tucker (KKT) conditions [15]

$$2(x_1^* - x_0) + A_1^\top\lambda_1^* = 0, \tag{15a}$$

$$A_1x_1^* = b_1. \tag{15b}$$

Solving the above set of equations for (x_1^*, λ_1^*) yields

$$x_1^* = x_0 - A_1^\top(A_1A_1^\top)^{-1}(A_1x_0 - b_1). \tag{16}$$

Using the same reasoning for (14b), we have

$$x_2^* = x_0 - A_2^\top(A_2A_2^\top)^{-1}(A_2x_0 - b_2). \tag{17}$$

From (16) and (17), we obtain

$$\begin{aligned}
\|x_1^* - x_2^*\| &= \|A_2^\top(A_2A_2^\top)^{-1}(A_2x_0 - b_2)\\
&\qquad - A_1^\top(A_1A_1^\top)^{-1}(A_1x_0 - b_1)\|\\
&\overset{(a)}{\leq} \|A_2^\top(A_2A_2^\top)^{-1}A_2x_0 - A_1^\top(A_1A_1^\top)^{-1}A_1x_0\|\\
&\quad + \|A_1^\top(A_1A_1^\top)^{-1}b_1 - A_2^\top(A_2A_2^\top)^{-1}b_2\|\\
&\overset{(b)}{=} \tilde{\alpha}\|A_1 - A_2\|\|x_0\|\\
&\quad + \|A_1^\top(A_1A_1^\top)^{-1}b_1 - A_1^\top(A_1A_1^\top)^{-1}b_2\\
&\quad + A_1^\top(A_1A_1^\top)^{-1}b_2 - A_2^\top(A_2A_2^\top)^{-1}b_2\|\\
&\overset{(c)}{\leq} \tilde{\alpha}\|A_1 - A_2\|\|x_0\| + \|A_1^\top(A_1A_1^\top)^{-1}\|\|b_1 - b_2\|\\
&\quad + \|b_2\|\|A_1^\top(A_1A_1^\top)^{-1} - A_2^\top(A_2A_2^\top)^{-1}\|\\
&\overset{(d)}{\leq} \tilde{\alpha}(\|x_0\| + \|b_2\|)\|A_1 - A_2\|\\
&\quad + \|A_1^\top(A_1A_1^\top)^{-1}\|\|b_1 - b_2\|,
\end{aligned}$$

where in (a) and (c), we have used the triangle inequality; in (b) and (d) we have used the bound from Lemma A.3. This completes the proof. ■

Comparing with the original and the perturbed network optimization problems, the constraint data (A_1, b_1) and (A_2, b_2) in the above result represent (A, b) and $(A + A_p, b + b_p)$ from (1b) and (7b), respectively. The above result has limitations in the sense that the cost has a specific structure and there are no inequality constraints. Nonetheless, we conjecture that one can drop these limitations and generalize the result by using the second-order strong convexity bounds.

VI. Conclusions

We have considered a constrained network optimization problem where the objective function is the summation of individual agents' objective and the agents' decisions are coupled through affine global equality constraints. We have explored two approaches that make this optimization problem amenable to the design of distributed algorithms. In the first approach, we have introduced additional variables to give two exact reformulations that only have local constraints in the decision variables of the agents. We have also discussed the scalability properties with the network size of these reformulations. In the second approach, we have analyzed the effect of perturbing the feasibility set of the problem on its optimizer. Our results help determine bounds on the distance between the optimizers of the original and perturbed problems when we eliminate some entries of the global affine constraints to improve sparsity. Future work will extend our perturbation analysis to general class of convex functions with affine constraints, analyze the effect of perturbation from a geometric perspective to identify entries of the affine constraint matrix that deviate the optimizer the least, and develop distributed algorithms for optimal sparsification. We also plan to apply our results to find efficient distributed algorithms for power system optimization problems.

Acknowledgments

This work was partially supported by NSF award ECCS-1307176 and ARPA-e Cooperative Agreement DE-

AR0000695.

References

[1] K. Arrow, L. Hurwitz, and H. Uzawa, *Studies in Linear and Non-Linear Programming*. Stanford, California: Stanford University Press, 1958.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[3] E. Dall’Anese, H. Zhu, and G. B. Giannakis, “Distributed optimal power flow for smart microgirds,” *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1464–1475, 2013.

[4] D. P. Palomar and M. Chiang, “Alternative distributed algorithms for network utility maximization: Framework and applications,” *IEEE Transactions on Automatic Control*, vol. 52, no. 12, pp. 2254–2269, 2007.

[5] Q. Ba, K. Savla, and G. Como, “Distributed optimal equilibrium selection for traffic flow over networks,” in *IEEE Conf. on Decision and Control*, (Osaka, Japan), pp. 6942–6947, 2015.

[6] A. Cherukuri and J. Cortés, “Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment,” *Automatica*, vol. 74, pp. 183–193, 2016.

[7] K. E. V. Horn, A. D. Domínguez-García, and P. W. Sauer, “Measurement-based real-time security-constrained economic dispatch,” *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3548–3560, 2016.

[8] A. Nedic, A. Ozdaglar, and P. A. Parrilo, “Constrained consensus and optimization in multi-agent networks,” *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[9] M. Zhu and S. Martínez, “On distributed convex optimization under inequality and equality constraints,” *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2012.

[10] J. Wang and N. Elia, “A control perspective for centralized and distributed convex optimization,” in *IEEE Conf. on Decision and Control*, (Orlando, Florida), pp. 3800–3805, 2011.

[11] D. Richert and J. Cortés, “Robust distributed linear programming,” *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2567–2582, 2015.

[12] J. F. Bonnans and A. Shapiro, “Optimization problems with perturbations: a guided tour,” *SIAM Review*, vol. 40, no. 2, pp. 228–264, 1998.

[13] R. Janin and J. Gauvin, “Lipschitz-type stability in nonsmooth convex programs,” *SIAM Journal on Optimization*, vol. 38, no. 1, pp. 124–137, 1999.

[14] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*. Applied Mathematics Series, Princeton University Press, 2009. Electronically available at `http://coordinationbook.info`.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009.

[16] J. Machowski, J. W. Bialek, and J. R. Bumby, *Power System Dynamics: Stability and Control*. Chichester, England: Wiley, 2008.

Appendix

The following results support the proof of Proposition 5.2.

Lemma A.1: Let $B_1, B_2 \in \mathbb{R}^{n\times n}$ be symmetric nonsingular matrices. Then,

$$\|B_1^{-1} - B_2^{-1}\| \leq \|B_1^{-1}\|\|B_2^{-1}\|\|B_1 - B_2\|,$$

Proof: The reasoning follows as

$$\begin{aligned}\|B_1^{-1} - B_2^{-1}\| &= \|B_1^{-1}B_1(B_1^{-1} - B_2^{-1})B_2B_2^{-1}\| \\ &= \|B_1^{-1}(B_2 - B_1)B_2^{-1}\| \\ &\leq \|B_1^{-1}\|\|B_2^{-1}\|\|B_1 - B_2\|.\end{aligned}$$

■

Lemma A.2: Let $A_1, A_2 \in \mathbb{R}^{m\times n}$ be full row-rank matrices. Then,

$$\|(A_2A_2^\top)^{-1} - (A_1A_1^\top)^{-1}\| \leq a\|A_1 - A_2\|,$$

where $a = \|(A_2A_2^\top)^{-1}\|\|(A_1A_1^\top)^{-1}\|\Big(\|A_1\| + \|A_2\|\Big)$.

Proof: From Lemma A.1, we obtain

$$\begin{aligned}\|(A_2A_2^\top)^{-1} - (A_1A_1^\top)^{-1}\| \leq \|(A_2A_2^\top)^{-1}\|\|(A_1A_1^\top)^{-1}\| \\ \|A_2A_2^\top - A_1A_1^\top\|.\end{aligned}$$

Further,

$$\begin{aligned}\|A_2A_2^\top - A_1A_1^\top\| &= \|A_2A_2^\top - A_2A_1^\top + A_2A_1^\top - A_1A_1^\top\| \\ &\leq (\|A_2\| + \|A_1\|)\|A_1 - A_2\|,\end{aligned}$$

where we have used the triangle inequality in the above expression. Combining the above sets of inequalities yields the result. ■

The next result uses the previous two.

Lemma A.3: Let $A_1, A_2 \in \mathbb{R}^{m\times n}$ be full row-rank matrices. Then,

$$\|A_1^\top(A_1A_1^\top)^{-1}A_1 - A_2^\top(A_2A_2^\top)^{-1}A_2\| \leq \tilde{\alpha}\|A_1 - A_2\|,$$

where

$$\begin{aligned}\tilde{\alpha} = \|A_1^\top(A_1A_1^\top)^{-1}\| + \|A_2\|\Big(\|(A_1A_1^\top)^{-1}\| \\ + \|A_2\|\|(A_2A_2^\top)^{-1}\|\|(A_1A_1^\top)^{-1}\|(\|A_1\| + \|A_2\|)\Big)\end{aligned}$$

Proof: Note that

$$\begin{aligned}&\|A_1^\top(A_1A_1^\top)^{-1}A_1 - A_2^\top(A_2A_2^\top)^{-1}A_2\| \\ &= \|A_1^\top(A_1A_1^\top)^{-1}A_1 - A_1^\top(A_1A_1^\top)^{-1}A_2 \\ &\quad + A_1^\top(A_1A_1^\top)^{-1}A_2 - A_2^\top(A_2A_2^\top)^{-1}A_2\| \\ &\leq \|A_1^\top(A_1A_1^\top)^{-1}\|\|A_1 - A_2\| \\ &\quad + \|A_2\|\|A_1^\top(A_1A_1^\top)^{-1} - A_2^\top(A_2A_2^\top)^{-1}\|. \qquad \text{(A.18)}\end{aligned}$$

For the second term in the last expression, we write

$$\begin{aligned}&\|A_1^\top(A_1A_1^\top)^{-1} - A_2^\top(A_2A_2^\top)^{-1}\| \\ &\quad = \|A_1^\top(A_1A_1^\top)^{-1} - A_2^\top(A_1A_1^\top) \\ &\qquad + A_2^\top(A_1A_1^\top)^{-1} - A_2^\top(A_2A_2^\top)^{-1}\| \\ &\leq \|(A_1A_1^\top)^{-1}\|\|A_1 - A_2\| \\ &\qquad + \|A_2^\top\|\|(A_1A_1^\top)^{-1} - (A_2A_2^\top)^{-1}\| \\ &\leq \Big(\|(A_1A_1^\top)^{-1}\| + \|A_2\|\|(A_2A_2^\top)^{-1}\|\|(A_1A_1^\top)^{-1}\| \\ &\qquad (\|A_1\| + \|A_2\|)\Big)\|A_1 - A_2\|,\end{aligned}$$

where the last inequality follows from Lemma A.2. The above inequality along with (A.18) completes the proof. ■