Newton-Raphson Consensus for Distributed Convex Optimization

Damiano Varagnolo, *Member, IEEE*, Filippo Zanella, *Member, IEEE*, Angelo Cenedese, *Member, IEEE*, Gianluigi Pillonetto, *Member, IEEE*, and Luca Schenato, *Senior Member, IEEE*

***Abstract*—We address the problem of distributed unconstrained convex optimization under separability assumptions, i.e., the framework where each agent of a network is endowed with a local private multidimensional convex cost, is subject to communication constraints, and wants to collaborate to compute the minimizer of the sum of the local costs. We propose a design methodology that combines average consensus algorithms and separation of time-scales ideas. This strategy is proved, under suitable hypotheses, to be globally convergent to the true minimizer. Intuitively, the procedure lets the agents distributedly compute and sequentially update an approximated Newton-Raphson direction by means of suitable average consensus ratios. We show with numerical simulations that the speed of convergence of this strategy is comparable with alternative optimization strategies such as the Alternating Direction Method of Multipliers. Finally, we propose some alternative strategies which trade-off communication and computational requirements with convergence speed.**

***Index Terms*—Consensus, distributed optimization, multi-agent systems, Newton-Raphson methods, smooth functions, unconstrained convex optimization.**

I. Introduction

OPTIMIZATION is a pervasive concept underlying many aspects of modern life [1]–[5], and it also includes the management of distributed systems, i.e., artifacts composed by a multitude of interacting entities often referred to as "agents". Examples are transportation systems, where the agents are both the vehicles and the traffic management devices (traffic lights), and smart electrical grids, where the agents are the energy producers-consumers and the power transformers-transporters.

Here we consider the problem of distributed optimization, i.e., the class of algorithms suitable for networked systems and characterized by the absence of a centralized coordination unit [6]–[8]. Distributed optimization tools have received an increasing attention over the last years, concurrently with the research on networked control systems. Motivations comprise the fact that the former methods let the networks self-organize and adapt to surrounding and changing environments, and that they are necessary to manage extremely complex systems in an autonomous way with only limited human intervention. In particular we focus on unconstrained convex optimization, although there is a rich literature also on distributed constrained optimization such as Linear Programming [9].

Literature Review: The literature on distributed unconstrained convex optimization is extremely vast and a first taxonomy can be based whether the strategy uses or not the Lagrangian framework, see, e.g., [5, Ch. 5].

Among the distributed methods exploiting Lagrangian formalism, the most widely known algorithm is Alternating Direction Method of Multipliers (ADMM) [10], whose roots can be traced back to [11]. Its efficacy in several practical scenarios is undoubted, see, e.g., [12] and references therein. A notable size of the dedicated literature focuses on the analysis of its convergence performance and on the tuning of its parameters for optimal convergence speed, see, e.g., [13] for Least Squares (LS) estimation scenarios, [14] for linearly constrained convex programs, and [15] for more general ADMM algorithms. Even if proved to be an effective algorithm, ADMM suffers from requiring synchronous communication protocols, although some recent attempts for asynchronous and distributed implementations have appeared [16]–[18].

On the other hand, among the distributed methods not exploiting Lagrangian formalisms, the most popular ones are the Distributed Subgradient Methods (DSMs) [19]. Here the optimization of non-smooth cost functions is performed by means of subgradient based descent/ascent directions. These methods arise in both primal and dual formulations, since sometimes it is better to perform dual optimization. Subgradient methods have been exploited for several practical purposes, e.g., to optimally allocate resources in Wireless Sensor Networks (WSNs) [20], to maximize the convergence speeds of gossip algorithms [21], to manage optimality criteria defined in terms of ergodic limits [22]. Several works focus on the analysis of the convergence properties of the DSM basic algorithm [23]–[25] (see [26] for a unified view of many convergence results). We can also find analyses for several extensions of the original idea, e.g., directions that are computed combining information from other agents [27], [28] and stochastic errors in the evaluation of the subgradients [29]. Explicit characterizations can also show trade-offs between desired accuracy and number of iterations [30].

Manuscript received April 10, 2014; revised November 4, 2014, May 31, 2015, and June 9, 2015; accepted June 18, 2015. This work was supported by the Framework Programme for Research and Innovation Horizon 2020 under the grant agreement 636834 "DISIRE", the Swedish research council Norrbottens Forskningsråd, by the University of Padova under the Progetto di Ateneo CPDA147754/14-New statistical learning approach for multi-agents adaptive estimation and coverage control, by the Italian Ministry of Education under Grant SCN 00398, and by Smart & safe Energy-aware Assisted Living. Recommended by Associate Editor A. Papachristodoulou.

D. Varagnolo is with the Department of Computer Science, Electrical and Space Engineering, Luleå University of Tec, 971 87 Luleå, Sweden (e-mail: damiano.varagnolo@ltu.se).

F. Zanella, A. Cenedese, G. Pillonetto, and L. Schenato are with the Department of Information Engineering, Università di Padova, 35131 Padova, Italy (e-mail: fzanella@dei.unipd.it; angelo.cenedese@dei.unipd.it; giapi@dei.unipd.it; schenato@dei.unipd.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2015.2449811

These methods have the advantage of being easily distributed, to have limited computational requirements and to be inherently asynchronous as shown in [31]–[33]. However they suffer from low convergence rate since they require the update steps to decrease to zero as $1/t$ (being t the time) therefore as a consequence the rate of convergence is sub-exponential. In fact, one of the current trends is to design strategies that improve the convergence rate of DSMs. For example, a way is to accelerate the convergence of subgradient methods by means of multi-step approaches, exploiting the history of the past iterations to compute the future ones [34]. Another is to use Newton-like methods, when additional smoothness assumptions can be used. These techniques are based on estimating the Newton direction starting from the Laplacian of the communication graph. More specifically, distributed Newton techniques have been proposed in dual ascent scenarios [35]–[37]. Since the Laplacian cannot be computed exactly, the convergence rates of these schemes rely on the analysis of inexact Newton methods [38]. These Newton methods are shown to have super-linear convergence under specific assumptions, but can be applied only to specific optimization problems such as network flow problems.

Recently, several alternative approaches to ADMM and DSM have appeared. For example, in [39], [40] the authors construct contraction mappings by means of cyclic projections of the estimate of the optimum onto the constraints. A similar idea based on contraction maps is used in F-Lipschitz methods [41] but it requires additional assumptions on the cost functions. Other methods are the control-based approach [42] which exploits distributed consensus, the distributed randomized Kaczmarz method [43] for quadratic cost functions, and distributed dual sub-gradient methods [44].

Statement of Contributions: Here we propose a distributed Newton-Raphson optimization procedure, named Newton-Raphson Consensus (NRC), for the exact minimization of smooth multidimensional convex separable problems, where the global function is a sum of private local costs. With respect to the classification proposed before, the strategy exploits neither Lagrangian formalisms nor Laplacian estimation steps. More specifically, it is based on average consensus techniques [45] and on the principle of separation of time-scales [46, Ch. 11]. The main idea is that agents compute and keep updated, by means of average consensus protocols, an approximated Newton-Raphson direction that is built from suitable Taylor expansions of the local costs. Simultaneously, agents move their local guesses towards the Newton-Raphson direction. It is proved that, if the costs satisfy some smoothness assumptions and the rate of change of the local update steps is sufficiently slow to allow the consensus algorithm to converge, then the NRC algorithm exponentially converges to the global minimizer.

The main contribution of this work is to propose an algorithm that extends Newton-Raphson ideas in a distributed setting, thus being able to exploit second order information to speed up converge rate. By using singular perturbation theory we formally show that under suitable assumptions the convergence of the algorithm is exponential (linear in logspace). Differently, DSM algorithms have sublinear convergence rate even if the cost functions are smooth [39], [47], although they are easy to implement and can be employed also for non-smooth cost functions and for constrained optimization. We also show by means of numerical simulations on real-world database benchmarks that the proposed algorithm exhibits faster convergence rates (in number of communications) than standard implementations of distributed ADMM algorithms [12], probably due to the second-order information embedded into the Newton-Raphson consensus. Although we have no theoretical guarantee of the superiority of the proposed algorithmic in terms of convergence rate, these simulations suggest that it is at least a potentially competitive algorithm. Moreover, one of the promising features of the NRC is that it is essentially based on average consensus algorithms, for which there exist robust implementations that encompass asynchronous communications, time-varying network topologies [48], directed graphs [49], and packet-losses effects.

Structure of the Paper: The paper is organized as follows: Section II collects the notation used through the whole paper, while Section III formulates the considered problem and provides some ancillary results that are then used to study the convergence properties of the main algorithm. Section IV proposes the main optimization algorithm, provides convergence results and describes some strategies to trade-off communication and computational complexities with convergence speed. Section V compares, via numerical simulations, the performance of the proposed algorithm with several distributed optimization strategies available in the literature. Finally, Section VI collects some final observations and suggests future research directions. We collect all the proofs in the Appendix.

II. Notation

We model the communication network as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ whose vertices $\mathcal{N} := \{1, 2, \dots, N\}$ represent the agents and whose edges $(i, j) \in \mathcal{E}$ represent the available communication links. We assume that the graph is undirected and connected, and that the matrix $P \in \mathbb{R}^{N\times N}$ is stochastic, i.e., its elements are non-negative, it is s.t. $P\mathbb{1} = \mathbb{1}$ (where $\mathbb{1} := [1\ 1\ \cdots\ 1]^T \in \mathbb{R}^N$), symmetric, i.e., $P = P^T$ and consistent with the graph $\mathcal{G}$, in the sense that each entry p_{ij} of P is $p_{ij} > 0$ only if $(i, j) \in \mathcal{E}$. We recall that if P is stochastic, symmetric, and includes all edges (i.e., $p_{ij} > 0$ if and only if $(i, j) \in \mathcal{E}$) then $\lim_{k\to\infty} P^k = (1/N)\mathbb{1}\mathbb{1}^T$. Such P's are also often referred to as *average consensus matrices*. We will indicate with $\rho(P) := \max_{i,\lambda_i\neq 1} |\lambda_i(P)|$ the spectral radius of P, with $\sigma(P) := 1 - \rho(P)$ its spectral gap.

We use fraction bars to indicate also Hadamard divisions, e.g., if $\boldsymbol{a} = [a_1, \dots, a_N]^T$ and $\boldsymbol{b} = [b_1, \dots, b_N]^T$ then $\boldsymbol{a}/\boldsymbol{b} := [(a_1/b_1)\ \cdots\ (a_N/b_N)]^T$. Fraction bars like the previous ones may also indicate pre-multiplication with inverse matrices, i.e., if b_i is a matrix then a_i/b_i indicates $b_i^{-1}a_i$. We indicate with n the dimensionality of the domains of the cost functions, k a discrete time index, t a continuous time index. For notational simplicity we denote differentiation with ∇ operators, so that $\nabla f = \partial f/\partial x$ and $\nabla^2 f = \partial^2 f/\partial x^2$. With a little abuse of notation, we will define $\chi = (x, Z)$, where $x \in \mathbb{R}^n$ and $Z \in \mathbb{R}^{\ell\times q}$ as the vector obtained by stacking in a column both the vector x and the vectorized matrix Z. We indicate with $\|\cdot\|$ Frobenius norms. With an other abuse of notation we also define the norm of the pair $\chi = (x, Z)$ where x is a vector and Z a matrix with $\|\chi\|^2 = \|x\|^2 + \|Z\|^2$.

When using plain italic fonts with a subscript (usually i, e.g., $x_i \in \mathbb{R}^n$) we refer to the local decision variable of the specific agent i. When using bold italic fonts, e.g., $\boldsymbol{x}$, we instead refer to the collection of the decision variables of all the various agents, e.g., $\boldsymbol{x} := [x_1^T, \dots, x_N^T]^T \in \mathbb{R}^{nN}$. To indicate special variables we will instead consider the following notation:

$$\begin{aligned} \overline{x} &:= \frac{1}{N}\sum_{i=1}^{N} x_i \quad \mathbb{R}^n \\ \boldsymbol{x}^{\parallel} &:= \mathbb{1}_N \otimes \overline{x} \quad \mathbb{R}^{nN} \\ \boldsymbol{x}^{\perp} &:= \boldsymbol{x} - \boldsymbol{x}^{\parallel} \quad \mathbb{R}^{nN} \end{aligned}$$

As in [46, p. 116], we say that a function V is a *Lyapunov function* for a specific dynamics if V is continuously differentiable and satisfies $V(0)=0$, $V(x)>0$ for $x\neq 0$, and $\dot{V}(x)\leq 0$.

III. Problem Formulation and Preliminary Results

A. Structure of the Section

Our main contribution is to characterize the convergence properties of the distributed Newton-Raphson (NR) scheme proposed in Section IV. In doing so we both exploit standard singular perturbation analysis tools [46, Ch. 11] [50] and a set of ancillary results, collected for readability in this section.

The logical flow of these ancillary results is the following: Section III-C claims that, under suitable assumptions, forward-Euler discretizations of stable continuous dynamics lead to stable discrete dynamics. This basic result enables reasoning on continuous-time systems. Then, Section III-D and E respectively claim that single- and multi-agent continuous-time NR dynamics satisfy these discretization assumptions. Section III-F and G then generalize these dynamics by introducing perturbation terms that mimic the behavior of the proposed main optimization algorithm, and characterize their stability properties. Summarizing, the ancillary results characterize the stability properties of systems that are progressive approximations of the dynamics under investigation.

B. Problem Formulation

We assume that the N agents of the network are endowed with cost functions $f_i : \mathbb{R}^n \mapsto \mathbb{R}$ so that

$$\overline{f} : \mathbb{R}^n \mapsto \mathbb{R}, \qquad \overline{f}(x) := \frac{1}{N}\sum_{i=1}^{N} f_i(x) \tag{1}$$

is a well-defined global cost. We assume that the aim of the agents is to cooperate and distributedly compute the minimizer of $\overline{f}$, namely

$$x^* := \arg\min_{x\in\mathbb{R}^n} \overline{f}(x). \tag{2}$$

We now enforce the following simplifying assumptions, valid throughout the rest of the paper.

Assumption 1 (Convexity): The local costs f_i in (1) are of class $\mathcal{C}^3$. Moreover the global cost $\overline{f}$ has bounded positive definite Hessian, i.e., $0 < cI \leq \nabla^2 \overline{f}(x) \leq mI$ for some $c, m \in \mathbb{R}_+$ and $\forall x \in \mathbb{R}^n$. Moreover, w.l.o.g., we assume $\overline{f}(x^*) = 0$, $c \leq 1$ and $m \geq 1$.

The scalar c is assumed to be known by all the agents *a-priori*. Assumption 1 ensures that x^* in (2) exists and is unique. The strictly positive definite Hessian is moreover a mild sufficient condition to guarantee that the minimum x^* defined in (2) will be globally exponentially stable under the continuous and discrete Newton-Raphson dynamics described in the following Theorem 3. We also notice that, for the subsequent Theorems 2 and 3, in principle just the average function $\overline{f}$ needs to have specific properties, and thus no conditions for the single f_i's are required (that for example might be even non convex). For the convergence of the distributed NR scheme we will nonetheless enforce the more restrictive Assumptions 5 and 9, not presented now for readability issues. In the rest of this section, in order to simplify notation, we will considerer, without loss of generality, the following translated cost functions:

$$f_i'(x) = f_i(x + x^*), \quad \overline{f}'(x) = \frac{1}{N}\sum_{i=1}^{N} f_i'(x) \tag{3}$$

so that the origin becomes the minimizer of the averaged cost function $\overline{f}'(x)$, i.e., $\overline{f}'(0) = 0$.

C. Stability of Discretized Dynamics

This subsection aims to show that, under suitable assumptions, forward-Euler discretization of suitable exponentially stable continuous-time dynamics maintains the same global exponential stability properties.

Theorem 2: Let the continuous-time system

$$\dot{x} = \phi(x) \tag{4}$$

admit $x = 0 \in \mathbb{R}^n$ as an equilibrium, and let $V(x) : \mathbb{R}^n \mapsto \mathbb{R}$ be a Lyapunov function for (4) for which there exist positive scalars a_1, a_2, a_3, a_4 s.t., $\forall x \in \mathbb{R}^n$

$$\begin{cases} a_1 I \leq \nabla^2 V(x) \leq a_2 I & \text{(5a)} \\ \frac{\partial V(x)}{\partial x}\phi(x) \leq -a_3\|x\|^2 & \text{(5b)} \\ \|\phi(x)\| \leq a_4\|x\|. & \text{(5c)} \end{cases}$$

Then

a) for system (4) the origin is globally exponentially stable;
b) for the following forward-Euler discretization of system (4):

$$x(k+1) = x(k) + \varepsilon\phi\left(x(k)\right) \tag{6}$$

there exists a positive scalar $\overline{\varepsilon}$ such that for every $\varepsilon \in (0, \overline{\varepsilon})$ the origin is globally exponentially stable.

D. Stability of Single-Agent NR Dynamics

This subsection shows that the results of Section III-C apply to continuous NR dynamics, i.e., that forward-Euler discretizations maintain global exponential stability properties.[1]

Theorem 3: Let

$$\phi_{\mathrm{NR}}(x) := -\overline{\overline{h'}}(x)^{-1}\nabla\overline{f}'(x) \tag{7}$$

be defined by a generic function $\overline{\overline{h'}}(x) \in \mathbb{R}^{n\times n}$ that satisfies the positive definiteness conditions $cI \leq \overline{\overline{h'}}(x) = \overline{\overline{h'}}(x)^T \leq mI$ for all $x \in \mathbb{R}^n$ where c and m are defined in Assumption 1. Let (7) define both the dynamics

$$\dot{x} = \phi_{\mathrm{NR}}(x) \tag{8}$$
$$x(k+1) = x(k) + \varepsilon\phi_{\mathrm{NR}}\left(x(k)\right). \tag{9}$$

[1] We notice that other asymptotic properties of continuous time NR methods are available in the literature, e.g., [51], [52].

Then, under Assumption 1:

a)

$$V_{\rm NR}(x) := \overline{f}'(x) \tag{10}$$

is a Lyapunov function for (8);

b) there exist positive scalars b_1, b_2, b_3, b_4 s.t., $\forall x \in \mathbb{R}^n$

$$\begin{cases} b_1 I \le \nabla^2 V_{\rm NR}(x) \le b_2 I & \text{(11a)} \\ \frac{\partial V_{\rm NR}}{\partial x}\phi_{\rm NR}(x) \le -b_3\|x\|^2 & \text{(11b)} \\ \|\phi_{\rm NR}(x)\| \le b_4\|x\|, & \text{(11c)} \end{cases}$$

i.e., Theorem 2 applies to dynamics (8) and (9).

For suitable choices of $\overline{\overline{h}}'(x)$ the dynamics (8) corresponds to continuous versions of well known descent dynamics. Indeed, the correspondences are

$$\overline{\overline{h}}'(x) = \begin{cases} \nabla^2\overline{f}'(x) & \to \text{Newton-Raphson descent} \quad \text{(12a)} \\ \operatorname{diag}\left[\nabla^2\overline{f}'(x)\right] & \to \text{Jacobi descent} \quad \text{(12b)} \\ I & \to \text{Gradient descent} \quad \text{(12c)} \end{cases}$$

where $\operatorname{diag}[A]$ is a diagonal matrix containing the main diagonal of A. Note that for every choice of $\overline{\overline{h}}'(x)$ as in (12a)–(12c), Assumption 1 ensures the hypotheses[2] of Theorem 3, therefore by combining Theorem 3 with Theorem 2 we are guaranteed that both continuous and discrete generalized NR dynamics induced by (7) are globally exponentially stable.

Lemma 4: Under Assumption 1, the origin is a globally exponentially stable point for dynamics (8). Moreover there exists $\overline{\varepsilon} > 0$ such that the origin is a globally exponentially stable point also for dynamics (9) for all $\varepsilon < \overline{\varepsilon}$.

The previous lemma and theorems do not require $\overline{\overline{h}}'(x)$ to be differentiable. However, differentiability may be used to linearize the system dynamics and obtain explicit rates of convergence. In fact, the linearized dynamics around the origin is given by

$$F(0) := \frac{\partial \phi_{\rm NR}(0)}{\partial x} = -\overline{\overline{h}}'(0)^{-1}\nabla^2\overline{f}'(0) - \frac{\partial \overline{\overline{h}}'(0)^{-1}}{\partial x}\nabla\overline{f}'(0).$$

In particular, for the NR descent it holds that $\overline{\overline{h}}'(x) = \nabla^2\overline{f}'(x)$. Thus in this case $F(0) = -I$, since $\nabla\overline{f}'(0) = 0$, and this says that the linearized continuous time NR dynamics is $\dot{x} = -x$, independent of the cost $\overline{f}'(x)$ and whose rate of convergence is unitary and uniform along any direction.

E. Stability of Multi-Agent NR Dynamics

We now generalize (8) by considering N coupled dynamical systems that, when starting at the very same initial condition, behave like N decoupled systems (8). This novel dynamics is the core of the slow-dynamics embedded in the main algorithm presented in Section IV. In this section we also include additional assumptions to show that the generalization of (8) presented here preserves global exponential stability and some other additional properties.

[2]For the Jacobi descent, clearly $\min_{\|x\|=1} x^T \operatorname{diag}[\nabla^2\overline{f}'(x)]x = \min_{x\in\{e_1,\dots,e_n\}} x^T \operatorname{diag}[\nabla^2\overline{f}'(x)]x = \min_{x\in\{e_1,\dots,e_n\}} x^T\nabla^2\overline{f}'(x)x \ge \min_{\|x\|=1} x^T\nabla^2\overline{f}'(x)x = c$, where e_i is the n-dimensional vector with all zeros except for a one in the i-th entry.

To this aim we introduce some additional notation: let $h_i'(x) : \mathbb{R}^n \mapsto \mathbb{R}^{n\times n}$, $i = 1,\dots,N$ be defined according to one of the possible three cases

$$h_i'(x) = \begin{cases} \nabla^2 f_i'(x) & \text{(13a)} \\ \operatorname{diag}\left[\nabla^2 f_i'(x)\right] & \text{(13b)} \\ I & \text{(13c)} \end{cases}$$

so that $h_i'(x) = h_i'(x)^T$ for all x. Moreover let

$$\begin{aligned} h'(\boldsymbol{x}) &:= [h_1'(x_1),\dots,\, h_N'(x_N)]^T && \mathbb{R}^{nN} \mapsto \mathbb{R}^{nN\times n} \\ \overline{h}'(\boldsymbol{x}) &:= \frac{1}{N}\sum_{i=1}^N h_i'(x_i) && \mathbb{R}^{nN} \mapsto \mathbb{R}^{n\times n} \\ \overline{\overline{h}}'(\overline{x}) &:= \frac{1}{N}\sum_{i=1}^N h_i'(\overline{x}) && \mathbb{R}^{n} \mapsto \mathbb{R}^{n\times n} \end{aligned}$$

be additional composite functions defined starting from the h_i''s (recall that $\boldsymbol{x} := [x_1^T,\dots,x_N^T]^T \in \mathbb{R}^{nN}$ and that $\overline{x} := (1/N)\sum_{i=1}^N x_i \in \mathbb{R}^n$). Let moreover

$$g_i'(x) := h_i'(x)x - \nabla f_i'(x) \qquad \mathbb{R}^n \mapsto \mathbb{R}^n \tag{14}$$

and $g'(\boldsymbol{x})$, $\overline{g}'(\boldsymbol{x})$, $\overline{\overline{g}}'(\overline{x})$ be defined accordingly as for h_i'.

The definitions of h_i' and g_i' are instrumental to generalize the NR dynamics (8) to the distributed case. Indeed, let

$$\psi(\boldsymbol{x}) := \overline{h}'(\boldsymbol{x})^{-1}\,\overline{g}'(\boldsymbol{x}) \qquad \mathbb{R}^{nN} \mapsto \mathbb{R}^n \tag{15}$$

(with the existence of $\overline{h}'(\boldsymbol{x})^{-1}$ guaranteed by the following Assumption 5). It is easy to verify that the previous functions satisfy the following properties:

$$\begin{cases} \overline{h}'(\boldsymbol{x}^{\parallel}) = \overline{\overline{h}}'(\overline{x}) & \text{(16a)} \\ \overline{g}'(\boldsymbol{x}^{\parallel}) = \overline{\overline{g}}'(\overline{x}) = \overline{\overline{h}}'(\overline{x})\overline{x} - \nabla\overline{f}'(\overline{x}) & \text{(16b)} \\ \psi(\boldsymbol{x}^{\parallel}) = \overline{x} - \overline{\overline{h}}'(\overline{x})^{-1}\nabla\overline{f}'(\overline{x}). & \text{(16c)} \end{cases}$$

Consider then

$$\dot{\boldsymbol{x}} = \phi_{\rm PNR}(\boldsymbol{x}) := -\boldsymbol{x} + \mathbb{1}_N \otimes \psi(\boldsymbol{x}) \tag{17}$$

that can be also equivalently written as

$$\dot{x}_i = -x_i + \psi(\boldsymbol{x}), \qquad i = 1,\dots,N$$

i.e., as the combination of N independent dynamical systems that are driven by the same forcing term $\psi(\boldsymbol{x})$.

As mentioned above, this dynamics embeds the centralized generalized NR dynamics since, under identical initial conditions $x_i(0) = \overline{x}(0) \in \mathbb{R}^n$ for all i, the trajectories coincide, i.e., $x_i(t) = \overline{x}(t)$, $\forall i$, $\forall t \ge 0$. Moreover, due to (16c)

$$\begin{aligned} \dot{\overline{x}} &= -\overline{x} + \psi(\mathbb{1}_N \otimes \overline{x}) \\ &= -\overline{x} + \overline{x} - \overline{\overline{h}}'(\overline{x})^{-1}\nabla\overline{f}'(\overline{x}) = \phi_{\rm NR}(\overline{x}) \end{aligned} \tag{18}$$

i.e., we obtain dynamics (7), that is, thanks to Theorem 3 and the assumption that $\overline{h}'(\boldsymbol{x})$ is invertible, globally exponentially stable.

The question is then whether dynamics (17) is exponentially stable also in the general case where the $x_i(0)$'s may not be identical. To characterize this case we assume some additional global properties.

Assumption 5 (Global Properties): The local costs $f'_1, \dots, f'_N$ in (1) are s.t. there exist positive scalars m_g, a_g, a_h, a_ψ s.t., $\forall\, x, x' \in \mathbb{R}^n$ and $\forall\, \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{nN}$

$$\begin{cases} cI \le \overline{h'}(\boldsymbol{x}) \le mI & \text{(19a)} \\ \left\|\overline{g'}(\boldsymbol{x})\right\| \le m_g & \text{(19b)} \\ \|g'_i(x) - g'_i(x')\| \le a_g \|x - x'\| & \text{(19c)} \\ \|h'_i(x) - h'_i(x')\| \le a_h \|x - x'\| & \text{(19d)} \\ \|\psi(\boldsymbol{x}) - \psi(\boldsymbol{x}')\| \le a_\psi \|\boldsymbol{x} - \boldsymbol{x}'\| & \text{(19e)} \end{cases}$$

with c and m from Assumption 1.

Note that Assumption 5 implies

$$\begin{cases} \left\|\overline{g'}(\boldsymbol{x}) - \overline{g'}(\boldsymbol{x}')\right\| \le a_g \|\boldsymbol{x} - \boldsymbol{x}'\| & \text{(20a)} \\ \left\|\overline{h'}(\boldsymbol{x}) - \overline{h'}(\boldsymbol{x}')\right\| \le a_h \|\boldsymbol{x} - \boldsymbol{x}'\| & \text{(20b)} \\ \|g'(\boldsymbol{x}) - g'(\boldsymbol{x}')\| \le a_g \|\boldsymbol{x} - \boldsymbol{x}'\| & \text{(20c)} \\ \|h'(\boldsymbol{x}) - h'(\boldsymbol{x}')\| \le a_h \|\boldsymbol{x} - \boldsymbol{x}'\|. & \text{(20d)} \end{cases}$$

Using the previous assumptions we can now prove global stability of dynamics (17).

Theorem 6: Under Assumptions 1 and 5, and for a suitable positive scalar η,

a)

$$V_{\mathrm{PNR}}(\boldsymbol{x}) := V_{\mathrm{NR}}(\overline{x}) + \frac{1}{2}\eta\|\boldsymbol{x}^\perp\|^2 = \overline{f'}(\overline{x}) + \frac{1}{2}\eta\|\boldsymbol{x}^\perp\|^2 \quad (21)$$

is a Lyapunov function for (17);

b) there exist positive scalars b_5, b_6, b_7, b_8 s.t., $\forall \boldsymbol{x} \in \mathbb{R}^{nN}$

$$\begin{cases} b_5 I \le \nabla^2 V_{\mathrm{PNR}}(\boldsymbol{x}) \le b_6 I & \text{(22a)} \\ \frac{\partial V_{\mathrm{PNR}}}{\partial \boldsymbol{x}} \phi_{\mathrm{PNR}}(\boldsymbol{x}) \le -b_7 \|\boldsymbol{x}\|^2 & \text{(22b)} \\ \|\phi_{\mathrm{PNR}}(\boldsymbol{x})\| \le b_8 \|\boldsymbol{x}\|. & \text{(22c)} \end{cases}$$

As in Lemma 4, combining Theorem 6 with Theorem 2 it is possible to claim that (17) and its discrete-time counterpart are globally exponentially stable.

F. Multi-Agent NR Dynamics Under Vanishing Perturbations

We now aim to generalize the dynamics $\phi_{\mathrm{PNR}}(\boldsymbol{x})$ by considering some perturbation term, that will be described by the variable $\boldsymbol{\chi}$. Let then $\boldsymbol{\chi}^y := (\chi_1^y, \dots, \chi_N^y)$ where $\chi_i^y \in \mathbb{R}^n$, $\boldsymbol{\chi}^z := (\chi_1^z, \dots, \chi_N^z)$ where $\chi_i^z = (\chi_i^z)^T \in \mathbb{R}^{n\times n}$, and $\boldsymbol{\chi} := (\boldsymbol{\chi}^y, \boldsymbol{\chi}^z)$. We also define the operator $[\cdot]_c : \mathbb{R}^{nN\times n} \mapsto \mathbb{R}^{nN\times n}$, which indicates the component-wise matrix-operation

$$[\boldsymbol{z}]_c = \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix}_c := \begin{bmatrix} z'_1 \\ \vdots \\ z'_N \end{bmatrix} \qquad z'_i = \begin{cases} z_i & \text{if } z_i \ge \frac{c}{2} I \\ \frac{c}{2} I & \text{otherwise.} \end{cases} \quad (23)$$

Consider then the perturbed version of the multi-agent NR dynamics (17)

$$\dot{\boldsymbol{x}} = \phi_x(\boldsymbol{x}, \boldsymbol{\chi}) := -\boldsymbol{x} - \mathbb{1}_N \otimes x^* + \frac{\boldsymbol{\chi}^y + \mathbb{1}_N \otimes \left(\overline{g'}(\boldsymbol{x}) + \overline{h'}(\boldsymbol{x}) x^*\right)}{\left[\boldsymbol{\chi}^z + \mathbb{1}_N \otimes \overline{h'}(\boldsymbol{x})\right]_c} \quad (24)$$

where the division is a Hadamard division, as recalled in Section II. Direct inspection of dynamics (24) then shows that

$$\phi_x(\boldsymbol{x}, \boldsymbol{0}) = \phi_{\mathrm{PNR}}(\boldsymbol{x}). \quad (25)$$

The next lemma provides perturbations interconnection bounds that will be used in Theorem 12.

Lemma 7: Under Assumptions 1 and 5 there exist positive scalars a_x, a_Δ s.t., for all $\boldsymbol{x}$ and $\boldsymbol{\chi}$

$$\begin{cases} \|\phi_x(\boldsymbol{x}, \boldsymbol{\chi})\| \le a_x \left(\|\boldsymbol{x}\| + \|\boldsymbol{\chi}\|\right) & \text{(26a)} \\ \|\phi_x(\boldsymbol{x}, \boldsymbol{\chi}) - \phi_{\mathrm{PNR}}(\boldsymbol{x})\| \le a_\Delta \|\boldsymbol{\chi}\|. & \text{(26b)} \end{cases}$$

G. Multi-Agent NR Dynamics Under Non-Vanishing Perturbations

Let us now consider some additional properties of the flow (24) for some specific non-vanishing perturbation. Consider then the perturbations $\xi^y \in \mathbb{R}^n$ and $\xi^z \in \mathbb{R}^{n\times n}$, and their multi-agents versions $\boldsymbol{\xi}^y = \mathbb{1}_N \otimes \xi^y$, $\boldsymbol{\xi}^y = \mathbb{1}_N \otimes \xi^z$. Consider also the shorthand $\boldsymbol{\xi} = (\boldsymbol{\xi}^y, \boldsymbol{\xi}^z)$. The equilibrium points of the dynamics induced by $\phi_x(\boldsymbol{x}, \boldsymbol{\xi})$ are characterized by the following theorem.

Theorem 8: Let $\xi^y \in \mathbb{R}^n$, $\xi^z \in \mathbb{R}^{n\times n}$, $\xi = (\xi^y, \xi^z)$, $\boldsymbol{\xi}^y = \mathbb{1}_N \otimes \xi^y$, $\boldsymbol{\xi}^z = \mathbb{1}_N \otimes \xi^z$, $\boldsymbol{\xi} = (\boldsymbol{\xi}^y, \boldsymbol{\xi}^z)$, and consider the equation

$$\phi_x(\boldsymbol{x}, \boldsymbol{\xi}) = 0$$

defining the equilibrium points of the dynamics $\dot{\boldsymbol{x}} = \phi_x(\boldsymbol{x}, \boldsymbol{\xi})$. Then, under Assumptions 1 and 5 there exist a positive scalar $r > 0$ and a unique continuously differentiable function $\boldsymbol{x}^{\mathrm{eq}} : \mathcal{B}_r \to \mathbb{R}^{nN}$ where $\mathcal{B}_r := \{\xi \mid \|\xi\| \le r\}$ such that

$$\phi_x\left(\boldsymbol{x}^{\mathrm{eq}}(\xi), \boldsymbol{\xi}\right) = 0, \quad \boldsymbol{x}^{\mathrm{eq}}(0) = 0. \quad (27)$$

Moreover, $\boldsymbol{x}^{\mathrm{eq}}(\xi) = \mathbb{1}_N \otimes x^{\mathrm{eq}}(\xi)$, with

$$x^{\mathrm{eq}}(\xi) = \left(\overline{\overline{h'}}(x^{\mathrm{eq}}(\xi)) + \xi^z\right)^{-1} \left(\overline{\overline{g'}}\left(x^{\mathrm{eq}}(\xi)\right) + \xi^y - \xi^z x^*\right). \quad (28)$$

Theorem 8 allows to define

$$\phi'_x(\boldsymbol{x}, \xi) := \phi_x\left(\boldsymbol{x} + \mathbb{1}_N \otimes x^{\mathrm{eq}}(\xi), \boldsymbol{\xi}\right) \quad (29)$$

and the corresponding dynamics

$$\dot{\boldsymbol{x}} = \phi'_x(\boldsymbol{x}, \xi) \quad (30)$$

which corresponds to the translated version of the original perturbed system $\phi_x(\boldsymbol{x}, \boldsymbol{\xi})$, which has now the property that the origin is an equilibrium point, i.e., $\phi'_x(\boldsymbol{0}, \xi) = 0$, $\forall \|\xi\| \le r$.

To prove the global exponential stability of (30) we need the flow ϕ'_x to satisfy a global Lipschitz condition.

Assumption 9 (Global Lipschitz Perturbation): There exist positive scalars a_ξ and r such that, for all $\boldsymbol{x} \in \mathbb{R}^{nN}$ and ξ satisfying $\|\xi\| \le r$

$$\|\phi'_x(\boldsymbol{x}, \xi) - \phi'_x(\boldsymbol{x}, 0)\| \le a_\xi \|\xi\| \|\boldsymbol{x}\|.$$

With these assumptions we can prove that the origin is a globally exponentially stable equilibrium for dynamics (30).

Theorem 10: Under Assumptions 1, 5, and 9,

a) $V_{\mathrm{PNR}}(\boldsymbol{x})$ defined in (21) is a Lyapunov function for (30);

b) there exist positive scalars r, b'_7, b'_8 s.t., for all $\boldsymbol{x} \in \mathbb{R}^{nN}$ and ξ satisfying $\|\xi\| \le r$

$$\begin{cases} \frac{\partial V_{\mathrm{PNR}}}{\partial \boldsymbol{x}} \phi'_x(\boldsymbol{x}, \xi) \le -b'_7 \|\boldsymbol{x}\|^2 & \text{(31a)} \\ \|\phi'_x(\boldsymbol{x}, \xi)\| \le b'_8 \|\boldsymbol{x}\|. & \text{(31b)} \end{cases}$$

Again, as in Lemma 4, combining Theorem 10 with Theorem 2 it is possible to claim that (30) and its discrete-time counterpart are globally exponentially stable.

H. Quadratic Functions

Before presenting the main algorithm, we show that quadratic costs satisfy all the previous assumptions. In fact, let us consider then

$$f_i(x) = \frac{1}{2}(x-d_i)^T A_i (x-d_i) + e_i, \quad A_i = A_i^T.$$

Based on this definition we have the following result.

Theorem 11: Quadratic costs that satisfy

$$A := \sum_i A_i > 0$$

satisfy Assumptions 1, 5, and 9 for $h_i'(x) = \nabla^2 f_i'(x)$.

IV. Newton-Raphson Consensus

In this section we provide an algorithm to distributively compute the minimizer of the function x^* defined in (2). The algorithm will be shown to converge to x^* even if $x^* \neq 0$. The proof of convergence will be based on the results derived in the previous sections via a suitable translation of the argument of the cost functions, which basically reduces the problem to the special case $x^* = 0$.

Consider then Algorithm 1, where $g(\boldsymbol{x}(-1)) = \mathbf{0}$ and $h(\boldsymbol{x}(-1)) = \mathbf{0}$ in the initialization step should be intended as initialization of suitable registers and not as operations involving the quantity $\boldsymbol{x}(-1)$.

Algorithm 1 Fast Newton-Raphson Consensus (NRC)

(*storage allocation and constraints on the parameters*)
$x_i(k)$, $y_i(k) \in \mathbb{R}^n$ and $z_i(k) \in \mathbb{R}^{n\times n}$ for all k and $i = 1,\dots,N$; $\varepsilon \in (0,1]$, $c > 0$
(*initialization*)
$x_i(0)=0$; $y_i(0)=g_i(x_i(-1))=0$; $z_i(0)=h_i(x_i(-1))=0$
(*main algorithm*)
for $k = 1,2,\dots$ **do**
  for $i = 1,\dots,N$ **do**
    $x_i(k)=(1-\varepsilon)x_i(k-1)+\varepsilon[z_i(k-1)]_c^{-1}y_i(k-1)$
    $y_i(k) = \sum_{j=1}^N p_{ij}(y_j(k-1) + g_j(x_j(k-1)) - g_j(x_j(k-2)))$
    $z_i(k) = \sum_{j=1}^N p_{ij}(z_j(k-1) + h_j(x_j(k-1)) - h_j(x_j(k-2)))$
  end for
end for

Intuitively, the algorithm functions as follows: if the dynamics of the $x_i(k)$s is sufficiently slow w.r.t. the dynamics of the $y_i(k)$sand $z_i(k)$s, then the two latter quantities tend to reach consensus. Then, the more these quantities reach consensus, the more the products $[z_i(k)]_c^{-1}y_i(k)$ exhibit these two specific characteristics: *i*) being the same among the various agent; *ii*) representing Newton descent directions. Thus, the more the $y_i(k)$s and $z_i(k)$s in Algorithm 1 are sufficiently close, the more the various $x_i(k)$s are driven by the same forcing term, that makes them converge to the same value, equal to the optimum x^*.

We now characterize the convergence properties of Algorithm 1. Let us define

$$\xi^y := \frac{1}{N}\sum_{i=1}^N (y_i(0) - g_i(x_i(-1)))$$

$$\xi^z := \frac{1}{N}\sum_{i=1}^N (z_i(0) - h_i(x_i(-1)))$$

then we have the following theorem.

Theorem 12: Consider the dynamics defined by Algorithm 1 with possibly nonzero initial conditions. If $\xi^y = 0$ and $\xi^z = 0$, then under Assumptions 1 and 5 there exists a positive scalar $\overline{\varepsilon} > 0$ such that Theorem 2 holds, i.e., the algorithm can be considered a forward-Euler discretization of a globally exponentially stable continuous dynamics. Thus the local estimates $x_i(k)$ produced by the algorithm exponentially converge to the global minimizer, i.e.,

$$\lim_{k\to\infty} x_i(k) = x^* \quad \forall i = 1,\dots,N$$

for all $\varepsilon \in (0,\overline{\varepsilon})$ and $x_i(0) \in \mathbb{R}^n$.

Consider now that, due to finite-precision issues, the quantities ξ^y and ξ^z may be non-null. Non-null initial ξ^y and ξ^z will make the proposed algorithm converge to a point that, in general does not coincide with the global optimum x^*. Nonetheless in this case the computed solution, as a function of the initial conditions, is a smooth function and thus small errors in the initial conditions do not produce dramatic errors in the computation of the optimum.

Theorem 13: Consider the dynamics defined by Algorithm 1 with possibly nonzero initial ξ^y and ξ^z but generic $x_i(0)$'s. Under Assumptions 1, 5, and 9 there exist positive scalars a, r, $\overline{\varepsilon}$ and a continuously differentiable function $\Psi : \mathbb{R}^n \times \mathbb{R}^{n\times n} \mapsto \mathbb{R}^n$ satisfying

$$\|\Psi(\xi^y,\xi^z) - x^*\| \leq a\,(\|\xi^y\| + \|\xi^z\|)$$

s.t. the local estimates exponentially converge to it, i.e.,

$$\lim_{k\to\infty} x_i(k) = \Psi(\xi^y,\xi^z) \quad \forall i = 1,\dots,N$$

for all $\varepsilon \in (0,\overline{\varepsilon})$, initial conditions $x_i(0) \in \mathbb{R}^n$ and $(\|\xi^y\| + \|\xi^z\|) \leq r$.

We notice that Theorem 13 ensures global convergence properties w.r.t. the initial conditions $x_i(0)$'s by requiring Assumptions 1, 5, and 9, while for the same convergence properties Theorem 12 requires only Assumptions 1 and 5. The difference is that Theorem 13 considers a non-null perturbation ξ and Assumption 9 is needed to cope with this additional perturbation term.

The Assumptions 1, 5, and 9 are not needed if only local convergence is ought. In fact, local differentiability, and therefore local Lipschitzianity, of the cost functions $f_i(x)$ at the minimizer x^* is sufficient to guarantee that Assumptions 5 and 9 are locally valid. As so, the proof that the equilibrium point is a locally exponentially stable point is exactly the same, with the difference that all bounds and inequalities are local. This observation is summarized in the following theorem.

Theorem 14: Consider the dynamics defined by Algorithm 1 with possibly nonzero initial conditions. Under the assumptions that the f_i's are $\mathcal{C}^3$ and that $\nabla^2\overline{f}(x^*) \geq cI$, there exist positive scalars a, r, $\overline{\varepsilon}$ and a continuously differentiable function $\Psi : \mathbb{R}^n \times \mathbb{R}^{n\times n} \mapsto \mathbb{R}^n$ s.t.

$$\lim_{k\to\infty} x_i(k) = \Psi(\xi^y, \xi^z) \quad \forall i = 1, \ldots, N$$

and satisfying

$$\|\Psi(\xi^y, \xi^z) - x^*\| \leq a\left(\|\xi^y\| + \|\xi^z\|\right)$$

for all $\varepsilon \in (0, \overline{\varepsilon})$ and initial conditions

$$\|x_i(0) - x^*\| \leq r, \quad \left\|y_i - \overline{\overline{g}}(x^*)\right\| \leq r, \quad \left\|z_i - \overline{\overline{h}}(x^*)\right\| \leq r$$
$$\left\|g_i\left(x_i(-1)\right) - \overline{\overline{g}}(x^*)\right\| \leq r, \quad \left\|h_i\left(x_i(-1)\right) - \overline{\overline{h}}(x^*)\right\| \leq r.$$

Numerical simulations suggest that the algorithm is robust w.r.t. numerical errors and quantization noise. We also notice that Theorem 12 guarantees the existence of a critical value $\overline{\varepsilon}$ but does not provide indications on its value. This is a known issue in all the systems dealing with separation of time scales. A standard rule of thumb is then to let the rate of convergence of the fast dynamics be sufficiently faster than the one of the slow dynamics, typically 2–10 times faster. In our algorithm the fast dynamics inherits the rate of convergence of the consensus matrix P, given by its spectral gap $\sigma(P)$, i.e., its spectral radius $\rho(P) = 1 - \sigma(P)$. The rate of convergence of the slow dynamics is instead governed by (18), which is nonlinear and therefore possibly depending on the initial conditions. However, close to the equilibrium point the dynamic behavior is approximately given by $\dot{\overline{x}}(t) \approx -(\overline{x}(t) - x^*)$, thus, since $x_i(k) \approx \overline{x}(\varepsilon k)$, then the convergence rate of the algorithm approximately given by $1 - \varepsilon$.

Thus we aim to let $1 - \rho(P) \gg 1 - (1 - \varepsilon)$, which provides the rule of thumb

$$\varepsilon \ll \sigma(P) \tag{32}$$

which is suitable for generic cost functions. We then notice that, although the spectral gap $\sigma(P)$ might not be known in advance, it is possible to distributedly estimate it, see, e.g., [53]. However, such rule of thumb might be very conservative. In fact, if all the f_i's are quadratic and are, w.l.o.g. s.t. $\nabla^2 f_i \geq cI$, then one can set $\varepsilon = 1$ and neglect the thresholding $[\cdot]_c$, so that the procedure reduces to

$$\begin{aligned} \boldsymbol{x}(k+1) &= \frac{\boldsymbol{y}(k)}{\boldsymbol{z}(k)} \\ \boldsymbol{y}(k+1) &= (P \otimes I_n)\boldsymbol{y}(k) \\ \boldsymbol{z}(k+1) &= (P \otimes I_n)\boldsymbol{z}(k) \end{aligned} \tag{33}$$

where $\boldsymbol{x}(k) := [x_1^T(k), \ldots, x_N^T(k)]^T$, $\boldsymbol{y}(k) := [y_1^T(k), \ldots, y_N^T(k)]^T$, $\boldsymbol{z}(k) := [z_1(k), \ldots, z_N(k)]^T$. Thus:

Theorem 15: Consider Algorithm 1 with arbitrary initial conditions $x_i(0)$, quadratic cost functions $f_i = (1/2)(x - d_i)^T A_i (x - d_i)$ with $A_i > 0$ and $\varepsilon = 1$. Then $\|x_i(k) - x^*\| \leq \alpha(\rho(P))^k$ for all k, i and for a suitable positive α.

Thus, if the cost functions are close to be quadratic then the overall rate of convergence is limited by the rate of convergence of the embedded consensus algorithm. Moreover, the values of ε that still guarantee convergence can be much larger than those dictated by the rule of thumb (32).

TABLE I
COMPUTATIONAL, COMMUNICATION, AND MEMORY COSTS OF NRC, JC, GDC PER SINGLE UNIT AND SINGLE STEP

Choice	NRC, $h_i(x) = \nabla^2 f_i(x)$	JC, $h_i(x) = \operatorname{diag}\left[\nabla^2 f_i(x)\right]$	GDC, $h_i(x) = I$
Computational Cost	$O\left(n^3\right)$	$O\left(n\right)$	$O\left(n\right)$
Communication Cost	$O\left(n^2\right)$	$O\left(n\right)$	$O\left(n\right)$
Memory Cost	$O\left(n^2\right)$	$O\left(n\right)$	$O\left(n\right)$

A. On the Selection of the Structure of $h(x)$

As introduced in Section III-D, by selecting different structures for $h_i(x)$ one can obtain different procedures with different convergence properties and different computational/ communication requirements. Plausible choices for h_i are the ones in (13c), and the correspondences are the following:

- $h_i(x) = \nabla^2 f_i(x) \to$ Newton-Raphson Consensus (NRC): in this case it is possible to rewrite the main algorithm and show that, for sufficiently small ε, $x_i(k) \approx \overline{x}(\varepsilon k)$, where $\overline{x}(t)$ evolves according to the continuous-time Newton-Raphson dynamics
$$\dot{\overline{x}}(t) = -\left[\nabla^2\overline{f}\left(\overline{x}(t)\right)\right]^{-1}\nabla\overline{f}\left(\overline{x}(t)\right).$$
- $h_i(x) = \operatorname{diag}[\nabla^2 f_i(x)] \to$ Jocobi Consensus (JC): choice $h_i(x) = \nabla^2 f_i(x)$ requires agents to exchange information on $O(n^2)$ scalars, and this could pose problems under heavy communication bandwidth constraints and large n's. Choice $h_i(x) = \operatorname{diag}[\nabla^2 f_i(x)]$ instead reduces the amount of information to be exchanged via the underlying diagonalization process, also called Jacobi approximation.[3] In this case, for sufficiently small ε, $x_i(k) \approx \overline{x}(\varepsilon k)$, where $\overline{x}(t)$ evolves according to the continuous-time dynamics
$$\dot{\overline{x}}(t) = -\left(\operatorname{diag}\left[\nabla^2\overline{f}\left(\overline{x}(t)\right)\right]\right)^{-1}\nabla\overline{f}\left(\overline{x}(t)\right)$$
which can be shown to converge to the global optimum x^* with a convergence rate that in general is slower than the Newton-Raphson when the global cost function is skewed.
- $h_i(x) = I \to$ Gradient Descent Consensus (GDC): this choice is motivated in frameworks where the computation of the local second derivatives $(\partial^2 f_i/\partial x_m^2)|_x$ is expensive (with x_m indicating here the m-th component of x), or where the second derivatives simply might not be continuous. With this choice the main algorithm reduces to a distributed gradient-descent procedure. In fact, for sufficiently small ε, $x_i(k) \approx \overline{x}(\varepsilon k)$ with $\overline{x}(t)$ evolving according to the continuous-time dynamics
$$\dot{\overline{x}}(t) = -\nabla\overline{f}\left(\overline{x}(t)\right)$$
which one again is guaranteed to converge to the global optimum x^*.

The following Table I summarizes the various costs of the previously proposed strategies.

We remark that $\overline{\varepsilon}$ in Theorem 12 depends also on the particular choice for h_i. The list of choices for h_i given above

[3]In centralized approaches, nulling the Hessian's off-diagonal terms is a well-known procedure, see, e.g., [54]. See also [36], [55] for other Jacobi algorithms with different communication structures.

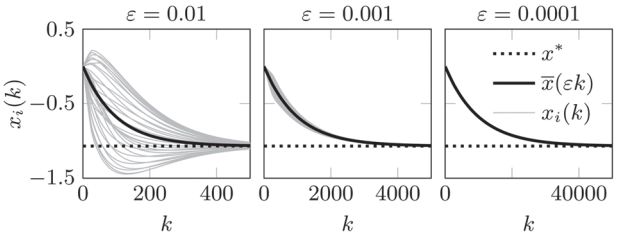

Fig. 1. Temporal evolution of system (43) for different values of ε, with $N = 30$. The black dotted line indicates x^*. The black solid line indicates the slow dynamics $\overline{x}(\varepsilon k)$ of Equation (18). As ε decreases, the difference between the time scale of the slow and fast dynamics increases, and the local states $x_i(k)$ converge to the manifold of $\overline{x}(\varepsilon k)$.

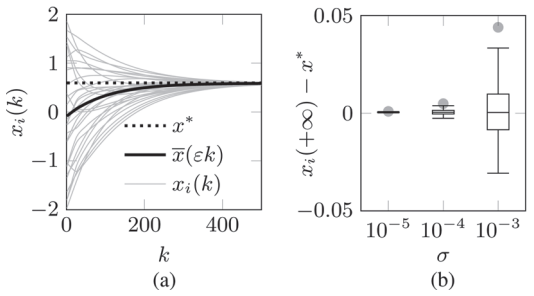

Fig. 2. Characterization of the dependency of the performance of Algorithm 1 on the initial conditions. In all the experiments $\varepsilon = 0.01$ and $N = 30$. (a) Time evolution of the local states $x_i(k)$ with $v(0) = w(0) = y(0) = z(0) = 0$ and $x_i(0) \sim \mathcal{U}[-2, 2]$. (b) Empirical distribution of the errors $x_i(+\infty) - x^*$ under artificially perturbed initial conditions $\alpha(0)$, $\beta(0) \sim \mathcal{U}[-\sigma, \sigma]$ for different values of σ.

is not exhaustive. For example, future directions are to implement distributed quasi-Newton procedures. To this regard, we recall that approximations of the Hessians that do not maintain symmetry and positive definiteness or are bad conditioned require additional modification steps, e.g., through Cholesky factorizations [56].

Finally, we notice that in scalar scenarios JC and NRC are equivalent, while GDC corresponds to algorithms requiring just the knowledge of first derivatives.

V. Numerical Examples

In Section V-A we analyze the effects of different choices of ε on the NRC on regular graphs and exponential cost functions. We then propose two machine learning problems in Section V-B, used in Section V-C and D, and numerically compare the convergence performance of the NRC, JC, GDC algorithms and other distributed convex optimization algorithms on random geometric graphs.

Notice that we will use cost functions that may not satisfy Assumptions 1, 5, and 9 to highlight the fact that the algorithm seems to have favorable numerical properties and large basins of stability even if the assumptions needed for global stability are not satisfied.

A. Effects of the Choice of ε

Consider a ring network of $S = 30$ agents that communicate only to their left and right neighbors through the consensus matrix

$$P = \begin{bmatrix} 0.5 & 0.25 & & & 0.25 \\ 0.25 & 0.5 & 0.25 & & \\ & \ddots & \ddots & \ddots & \\ & & 0.25 & 0.5 & 0.25 \\ 0.25 & & & 0.25 & 0.5 \end{bmatrix} \tag{34}$$

so that the spectral radius $\rho(P) \approx 0.99$, implying a spectral gap $\sigma(P) \approx 0.01$. Consider also scalar costs of the form $f_i(x) = c_i e^{a_i x} + d_i e^{-b_i x}$, $i = 1, \ldots, N$, with a_i, $b_i \sim \mathcal{U}[0, 0.2]$, c_i, $d_i \sim \mathcal{U}[0, 1]$ and where $\mathcal{U}$ indicates the uniform distribution.

Fig. 1 compares the evolution of the local states x_i of the continuous system (43) for different values of ε. When ε is not sufficiently small, then the trajectories of $x_i(t)$ are different even if they all start from the same initial condition $x_i(0) = 0$. As ε decreases, the difference between the two time scales becomes more evident and all the trajectories $x_i(k)$ become closer to the trajectory given by the slow NR dynamics $\overline{x}(\varepsilon k)$ given in (18) and guaranteed to converge to the global optimum x^*.

In Fig. 2 we address the robustness of the proposed algorithm w.r.t. the choice of the initial conditions. In particular, Fig. 2(a) shows that if $\alpha = \beta = 0$ then the local states $x_i(t)$ converge to the optimum x^* for arbitrary initial conditions $x_i(0)$. Fig. 2(b) considers, besides different initial conditions $x_i(0)$, also perturbed initial conditions $v(0)$, $w(0)$, $y(0)$, $z(0)$ leading to non null α's and β's. More precisely we apply Algorithm 1 to different random initial conditions s.t. α, $\beta \sim \mathcal{U}[-\sigma, \sigma]$. Fig. 2(b) shows the boxplots of the errors $x_i(+\infty) - x^*$ for different σ's based on 300 Monte Carlo runs with $\varepsilon = 0.01$ and $N = 30$.

B. Optimization Problems

The first problem considered is the distributed training of a Binomial-Deviance based classifier, to be used, e.g., for spam-nonspam classification tasks [57, Ch. 10.5]. More precisely, we consider a database of emails E, where j is the email index, $y_j = -1, 1$ denotes if the email j is considered spam or not, $\chi_j \in \mathbb{R}^{n-1}$ numerically summarizes the $n-1$ features of the j-th email (how many times the words "money", "dollars", etc., appear). If the E emails come from different users that do not want to disclose their private information, then it is meaningful to exploit the distributed optimization algorithms described in the previous sections. More specifically, letting $x = (x', x_0) \in \mathbb{R}^{n-1} \times \mathbb{R}$ represents a generic classification hyperplane, training a Binomial-Deviance based classifier corresponds to solve a distributed optimization problem where the local cost functions are given by

$$f_i(x) := \sum_{j \in E_i} \log\left(1 + \exp\left(-y_j\left(\chi_j^T x' + x_0\right)\right)\right) + \gamma \left\|x'\right\|_2^2 \tag{35}$$

where E_i is the set of emails available to agent i, $E = \cup_{i=1}^N E_i$, and γ is a global regularization parameter. In the following numerical experiments we consider $|E| = 5000$ emails from the spam-nonspam UCI repository, available at http://archive.ics.uci.edu/ml/datasets/Spambase, randomly assigned to 30 different users communicating as in graph of Fig. 4. For each email we consider 3 features (the frequency of words "make", "address", "all") so that the corresponding optimization problem is 4-dimensional.

The second problem considered is a regression problem inspired by the UCI Housing dataset available at http://archive.ics.uci.edu/ml/datasets/Housing. In this task, an example $\chi_j \in \mathbb{R}^{n-1}$ is a vector representing some features of a house (e.g.,

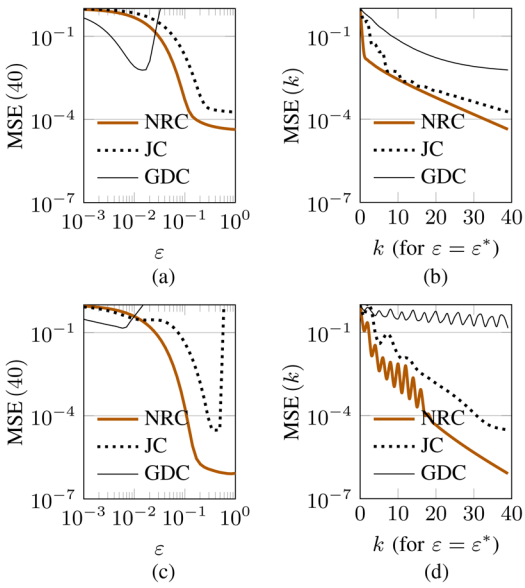

Fig. 3. Convergence properties of Algorithm 1 for the problems described in Section V-B and for different choices of $h_i(\cdot)$. Choice $h_i(x) = \nabla^2 f_i(x)$ corresponds to the NRC algorithm, $h_i(x) = \mathrm{diag}[\nabla^2 f_i(x)]$ to the JC, $h_i(x) = I$ to the GDC. (a) Relative MSE at a given time k as a function of the parameter ε for classification problem (35). (b) Relative MSE as a function of the time k, with the parameter ε chosen as the best from Fig. 3(a) for classification problem (35). (c) Relative MSE at a given time k as a function of the parameter ε for regression problem (36). (d) Relative MSE as a function of the time k, with the parameter ε chosen as the best from Fig. 3(c) for regression problem (36).

per capita crime rate by town, index of accessibility to radial highways, etc.), and $y_j \in \mathbb{R}$ denotes the corresponding median monetary value of of the house. The objective is to obtain a predictor of house value based on these data. Similarly as the previous example, if the datasets come from different users that do not want to disclose their private information, then it is meaningful to exploit the distributed optimization algorithms described in the previous sections. This problem can be formulated as a convex regression problem on the local costs

$$f_i(x) := \sum_{j \in E_i} \frac{\left(y_j - \chi_j^T x' - x_0\right)^2}{\left|y_j - \chi_j^T x' - x_0\right| + \beta} + \gamma \left\|x'\right\|_2^2 \qquad (36)$$

where $x = (x', x_0^*) \in \mathbb{R}^{n-1} \times \mathbb{R}$ is the vector of coefficient for the linear predictor $\widehat{y} = \chi^T x' + x_0$ and γ is a common regularization parameter. The loss function $(\cdot)^2/(|\cdot| + \beta)$ corresponds to a smooth $\mathcal{C}^2$ version of the Huber robust loss, a loss that is usually employed to minimize the effects of outliers. In our case β dictates for which arguments the loss is pseudo-linear or pseudo-quadratic and has been manually chosen to minimize the effects of outliers. In our experiments we used four features, $\beta = 50$, $\gamma = 1$, and $|E| = 506$ total number of examples in the dataset randomly assigned to the $N = 30$ users communicating as in the graph of Fig. 4.

In both the previous problems the optimum, in the following indicated for simplicity with x^*, has been computed with a centralized NR with the termination rule "stop when in the last 5 steps the norm of the guessed x^* changed less than $10^{-9}\%$".

C. Comparison of the NRC, JC and GDC Algorithms

In Fig. 3 we analyze the performance of the three proposed NRC, JC and GDC algorithms defined by the various choices

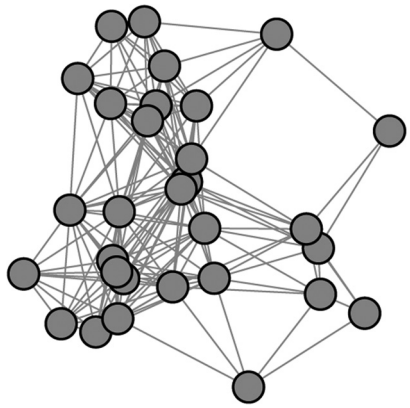

Fig. 4. Random geometric graph exploited in the simulations relative to the optimization problem (35). For this graph $\rho(P) \approx 0.9338$, with P the matrix of Metropolis weights.

for $h_i(x)$ in Algorithm 1 in terms of the relative MSE

$$\mathrm{MSE}(k) := \frac{1}{N} \sum_{i=1}^{N} \left\|x_i(k) - x^*\right\|^2 / \left\|x^*\right\|^2$$

for the classification and regression optimization problem described above. The consensus matrix P has been by selecting the Metropolis-Hastings weights which are consistent with the communication graph [58]. Panels 3(a) and 3(c) report the MSE obtained at a specific iteration $(k = 40)$ by the various algorithms, as a function of ε. These plots thus inspect the sensitivity w.r.t. the choice of the tuning parameters. Consistently with the theorems in the previous section, the GDC and JC algorithms are stable only for ε sufficiently small, while NRC exhibit much larger robustness and best performance for $\varepsilon = 1$. Panels 3(b) and 3(d) instead report the evolutions of the relative MSE as a function of the number of iterations k for the optimally tuned algorithms.

We notice that the differences between NRC and JC are evident but not resounding, due to the fact that the Jacobi approximations are in this case a good approximation of the analytical Hessians. Conversely, GDC presents a slower convergence rate which is a known drawback of gradient descent algorithms.

D. Comparisons With Other Distributed Convex Optimization Algorithms

We now compare Algorithm 1 and its accelerated version, referred as Fast Newton-Raphson Consensus (FNRC) and described in detail below in Algorithm 2), with three popular distributed convex optimization methods, namely the DSM, the Distributed Control Method (DCM) and the ADMM, described respectively in Algorithm 3, 4, and 5. The following discussion provides some details about these strategies.

Algorithm 2 Fast Newton-Raphson Consensus

storage allocation, constraints on the parameters and initialization as in Algorithm 1
for $k = 1, 2, \ldots$ **do**
  for $i = 1, \ldots, N$ **do**
    $x_i(k) = (1-\varepsilon)x_i(k-1) + \varepsilon[z_i(k-1)]_c^{-1} y_i(k-1)$
    $\widetilde{y}_i(k) = y_i(k-1) + (1/\varphi) g_i(x_i(k-1)) - g_i(x_i(k-2)) - ((1-\varphi)/\varphi) g_i(x_i(k-3))$

6: $\widetilde{z}_i(k) = z_i(k-1) + (1/\varphi)h_i(x_i(k-1)) - h_i(x_i(k-2)) - ((1-\varphi)/\varphi)h_i(x_i(k-3))$
7: $y_i(k) = \varphi \sum_{j=1}^{N}(p_{ij}\widetilde{y}_j(k)) + (1-\varphi)y_i(k-2)$
8: $z_i(k) = \varphi \sum_{j=1}^{N}(p_{ij}\widetilde{z}_j(k)) + (1-\varphi)z_i(k-2)$
9: **end for**
10: **end for**

Algorithm 3 DSM [30]

(*storage allocation and constraints onparameters*)
1: $x_i(k) \in \mathbb{R}^n$ for all i. $\varrho \in \mathbb{R}_+$
(*initialization*)
2: $x_i(0) = 0$
(*main algorithm*)
3: **for** $k = 0, 1, \ldots$ **do**
4: **for** $i = 1, \ldots, N$ **do**
5: $x_i(k+1) = \sum_{j=1}^{N} p_{ij}(x_j(k) - (\varrho/k)\nabla f_j(x_j(k)))$
6: **end for**
7: **end for**

Algorithm 4 DCM [42]

(*storage allocation and constraints onparameters*)
1: $x_i(k), z_i(k) \in \mathbb{R}^n$, for all i. $\mu, \nu \in \mathbb{R}_+$
(*initialization*)
2: $x_i(0) = z_i(0) = 0$ for all i
(*main algorithm*)
3: **for** $k = 0, 1, \ldots$ **do**
4: **for** $i = 1, \ldots, N$ **do**
5: $z_i(k+1) = z_i(k) + \mu \sum_{j \in \mathcal{N}_i}(x_i(k) - x_j(k))$
6: $x_i(k+1) = x_i(k) + \mu \sum_{j \in \mathcal{N}_i}(x_j(k) - x_i(k)) + \mu \sum_{j \in \mathcal{N}_i}(z_j(k) - z_i(k)) - \mu\nu\nabla f_i(x_i(k))$
7: **end for**
8: **end for**

Algorithm 5 ADMM [7, pp. 253–261]

(*storage allocation and constraints onparameters*)
1: $x_i(k), z_{(i,j)}(k), y_{(i,j)}(k) \in \mathbb{R}^n$, $\delta \in (0, 1)$
(*initialization*)
2: $x_i(k) = z_{(i,j)}(k) = y_{(i,j)}(k) = 0$
(*main algorithm*)
3: **for** $k = 0, 1, \ldots$ **do**
4: **for** $i = 1, \ldots, N$ **do**
5: $x_i(k+1) = \arg\min_{x_i} L_i(x_i, k)$
6: **for** $j \in \mathcal{N}_i$ **do**
7: $z_{(i,j)}(k+1) = (1/2\delta)(y_{(i,j)}(k) + y_{(j,i)}(k)) + (1/2)(x_i(k+1) + x_j(k+1))$
8: $y_{(i,j)}(k+1) = y_{(i,j)}(k) + \delta(x_i(k+1) - z_{(i,j)}(k+1))$
9: **end for**
10: **end for**
11: **end for**

- FNRC is an accelerated version of Algorithm 1 that inherits the structure of the so called *second order diffusive schedules*, see, e.g., [59], and exploits an additional level of memory to speed up the convergence properties of the consensus strategy. Here the weights multiplying the g_i's and h_i's are necessary to guarantee exact tracking of the current average, i.e., $\sum_i y_i(k) = \sum_i g_i(x(k-1))$ for all k. As suggested in [59], we set the φ that weights the gradient and the memory to $\varphi = 2/(1 + \sqrt{1 - \rho(P)^2})$. This guarantees second order diffusive schedules to be faster than first order ones (even if this does not automatically imply the FNRC to be faster than the NRC). This setting can be considered a valid heuristic to be used when $\rho(P)$ is known. For the graph in Fig. 4, $\varphi \approx 1.4730$.
- DSM, as proposed in [30], alternates consensus steps on the current estimated global minimum $x_i(k)$ with subgradient updates of each $x_i(k)$ towards the local minimum. To guarantee the convergence, the amplitude of the local subgradient steps should appropriately decrease. Algorithm 3 presents a synchronous DSM implementation, where ϱ is a tuning parameter and P is the matrix of Metropolis-Hastings weights.
- DCM, as proposed in [42], differentiates from the gradient searching because it forces the states to the global optimum by controlling the subgradient of the global cost. This approach views the subgradient as an input/output map and uses small gain theorems to guarantee the convergence property of the system. Again, each agents i locally computes and exchanges information with its neighbors, collected in the set $\mathcal{N}_i := \{j \mid (i,j) \in \mathcal{E}\}$. DCM is summarized in Algorithm 4, where μ, $\nu > 0$ are parameters to be designed to ensure the stability property of the system. Specifically, μ is chosen in the interval $0 < \mu < 2/(2\max_{i=\{1,\ldots,N\}} |\mathcal{N}_i| + 1)$ to bound the induced gain of the subgradients. Also here the parameters have been manually tuned for best convergence rates.
- ADMM, instead, requires the augmentation of the system through additional constraints that do not change the optimal solution but allow the Lagrangian formalism. There exist different implementations of ADMM in distributed contexts, see, e.g., [7], [12, pp. 253–261], [60]. For simplicity we consider the following formulation:

$$\min_{x_1,\ldots,x_N} \sum_{i=1}^{N} f_i(x_i)$$
$$\text{s.t. } z_{(i,j)} = x_i, \quad \forall i \in \mathcal{N}, \quad \forall (i,j) \in \mathcal{E}$$

where the auxiliary variables $z_{(i,j)}$ correspond to the different links in the network, and where the local Augmented Lagrangian is given by

$$L_i(x_i, k) := f_i(x_i) + \sum_{j \in \mathcal{N}_i} y_{(i,j)}\left(x_i - z_{(i,j)}\right) + \sum_{\in \mathcal{N}_i} \frac{\delta}{2} \left\| x_i - z_{(i,j)} \right\|^2$$

with δ a tuning parameter (see [61] for a discussion on how to tune it) and the $y_{(i,j)}$'s Lagrange multipliers.

The computational, communication and memory costs of these algorithms is reported in Table II. Notice that the computational and memory costs of ADMM algorithms depends on how nodes minimize the local augmented Lagrangian $L_i(x_i, k)$. E.g., in our simulations the step has been performed through a dedicated Newton-Raphson procedure with associated $O(n^3)$ computational costs and $O(n^2)$ memory costs.

TABLE II
COMPUTATIONAL, COMMUNICATION AND MEMORY COSTS OF DSM, DCM, AND ADMM PER SINGLE UNIT AND SINGLE STEP

Choice	DSM	DCM	ADMM
Computational Cost	$O(n)$	$O(n)$	from $O(n)$ to $O(n^3)$
Communication Cost	$O(n)$	$O(n)$	$O(n)$
Memory Cost	$O(n)$	$O(n)$	from $O(n)$ to $O(n^2)$

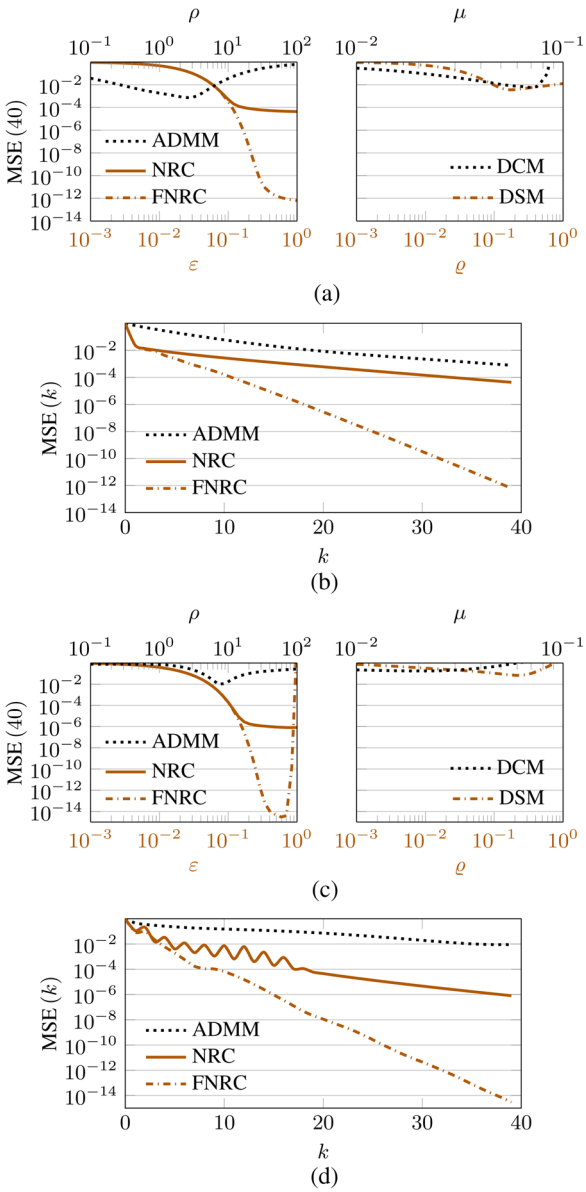

Fig. 5. Convergence properties of the various algorithms for the problems described in Section V-B. (a) Relative MSE at a given time k as a function of the algorithms parameters for problem (35). For the DCM, $\nu = 1.7$. (b) Relative MSE as a function of the time k for the three fastest algorithms for problem (35). Their parameters are chosen as the best ones from Fig. 5(a). (c) Relative MSE at a given time k as a function of the algorithms parameters for problem (36). For the DCM, $\nu = 1.7$. (d) Relative MSE as a function of the time k for the three fastest algorithms for problem (36). Their parameters are chosen as the best ones from Fig. 5(c).

Fig. 5 then compares the previously cited algorithms as did in Fig. 3. The first panel thus reports the relative MSE of the various algorithms at a given number of iterations ($k = 40$) as a function of the parameters. The second panel instead reports the temporal evolution of the relative MSE for the case of optimal tuning.

We notice that the DCM and the DSM are both much slower, in terms of communications iterations, than the NRC, FNRC and ADMM. Moreover, both the NRC and its accelerated version converge faster than the ADMM, even if not tuned at their best. These numerical examples seem to indicate that the proposed NRC might be a viable alternative to the ADMM, although further comparisons are needed to strengthen this claim. Moreover, a substantial potential advantage of NRC as compared to ADMM is that the former can be readily adapted to asynchronous and time-varying graphs, as preliminary made in [62]. Moreover, as in the case of the FNRC, the strategy can implement any improved linear consensus algorithm.

VI. CONCLUSION

We proposed a novel distributed optimization strategy suitable for convex, unconstrained, multidimensional, smooth and separable cost functions. The algorithm does not rely on Lagrangian formalisms and acts as a distributed Newton-Raphson optimization strategy by repeating the following steps: agents first locally compute and update second order Taylor expansions around the current local guesses and then they suitably combine them by means of average consensus algorithms to obtain a sort of approximated Taylor expansion of the global cost. This allows each agent to infer a local Newton direction, used to locally update the guess of the global minimum.

Importantly, the average consensus protocols and the local updates steps have different time-scales, and the whole algorithm is proved to be convergent only if the step-size is sufficiently slow. Numerical simulations based on real-world databases show that, if suitably tuned, the proposed algorithm is faster then ADMMs in terms of number of communication iterations, although no theoretical proof is provided.

The set of open research paths is extremely vast. We envisage three main avenues. The first one is to study how the agents can dynamically and locally tune the speed of the local updates w.r.t. the consensus process, namely how to tune their local step-size ε_i. In fact large values of ε gives faster convergence but might lead to instability. A second one is to let the communication protocol be asynchronous: in this regard we notice that some preliminary attempts can be found in [62]. A final branch is about the analytical characterization of the rate of convergence of the proposed strategies, a theoretical comparison with ADMMs, and the extensions to non-smooth convex functions.

APPENDIX

Proof (of Theorem 2): Proof of a): integrating (5a) twice implies

$$\frac{1}{2}a_1\|x\|^2 \leq V(x) \leq \frac{1}{2}a_2\|x\|^2$$

that, jointly with (5b), immediately guarantee global exponential stability for (4) [46, Thm. 4.10].

Proof of b): consider

$$\Delta V\left(x(k)\right) := V\left(x(k+1)\right) - V\left(x(k)\right). \qquad (37)$$

To prove the claim we show that $\Delta V(x(k)) \leq -d\|x(k)\|^2$ for some positive scalar d. To this aim, expand $V(x(k+1))$ with a

second order Taylor expansion around $x(k)$ with remainder in Lagrange form, to obtain

$$V\left(x+\varepsilon\phi(x)\right)=V(x)+\varepsilon\frac{\partial V}{\partial x}\phi(x)+\frac{1}{2}\varepsilon^2\phi^T(x)\nabla^2V(x_\varepsilon)\phi(x)$$

with $x_\varepsilon = x+\varepsilon'\phi(x)$ for $\varepsilon'\in[0,\varepsilon]$. Using inequalities (5) we then obtain

$$\begin{aligned}\Delta V\left(x(k)\right) &= V\left(x(k+1)\right)-V\left(x(k)\right)\\ &\leq -\varepsilon a_3\left\|x(k)\right\|^2+\frac{1}{2}\varepsilon^2a_2a_4^2\left\|x(k)\right\|^2\\ &= -\varepsilon\left(a_3-\varepsilon\frac{1}{2}a_2a_4^2\right)\left\|x(k)\right\|^2.\end{aligned}$$

Thus, for all $\varepsilon<\bar{\varepsilon}=2a_3/a_2a_4^2$ the origin is globally exponentially stable. ■

Proof (of Theorem 3): Proof of a): Assumption 1 guarantees that $V_{\rm NR}(0)=0$ and $V_{\rm NR}(x)>0$ for $x\neq 0$. Moreover, for $x\neq 0$

$$\begin{aligned}\frac{\partial V_{\rm NR}}{\partial x}\phi_{\rm NR}(x) &= -\left(\nabla\overline{f}'(x)\right)^T\overline{\overline{h}'}(x)^{-1}\nabla\overline{f}'(x)\\ &= -\left\|\overline{\overline{h}'}(x)^{-\frac{1}{2}}\,\nabla\overline{f}'(x)\right\|^2<0.\end{aligned}$$

Proof of b): Assumption 1 guarantees that (11a) is satisfied with $b_1=c$ and $b_2=m$. To prove (11c) we start by considering that (11a) guarantees $c\|x\|\leq\|\nabla\overline{f}'(x)\|\leq m\|x\|$. This in its turn implies

$$\begin{aligned}\|\phi_{\rm NR}(x)\| &= \left\|\overline{\overline{h}'}^{-1}(x)\nabla\overline{f}'(x)\right\|\leq\frac{1}{c}\left\|\nabla\overline{f}'(x)\right\|\leq\frac{m}{c}\|x\|\\ &= b_4\|x\|.\end{aligned}$$

To prove (11b) eventually consider then that (11c) implies

$$\begin{aligned}\frac{\partial V_{\rm NR}}{\partial x}\phi_{\rm NR}(x) &= -\left(\nabla\overline{f}'(x)\right)^T\overline{\overline{h}'}(x)^{-1}\nabla\overline{f}'(x)\\ &\leq -\frac{c^2}{m}\|x\|^2=-b_3\|x\|^2.\end{aligned}$$

■

Proof (of Theorem 6): In the interest of clarity we analyze the case where the local costs f_i' are scalar, i.e., $n=1$. The multivariable case is indeed a straightforward extension with just a more involved notation. We also recall the following equivalences:

$$\begin{aligned}\boldsymbol{x}&=\boldsymbol{x}^{\parallel}+\boldsymbol{x}^{\perp},\quad (\boldsymbol{x}^{\perp})^T\boldsymbol{x}^{\parallel}=0\\ \|\boldsymbol{x}\|^2&=\|\boldsymbol{x}^{\parallel}\|^2+\|\boldsymbol{x}^{\perp}\|^2=N|\overline{x}|^2+\|\boldsymbol{x}^{\perp}\|^2.\end{aligned}$$

Proof of a): $V_{\rm PNR}(\mathbf{0})=0$ and $V_{\rm PNR}(\boldsymbol{x})>0$ for $\boldsymbol{x}\neq\mathbf{0}$ follow immediately from the fact that $V_{\rm NR}(0)=0$ and $V_{\rm NR}(\overline{x})>0$ for $\overline{x}\neq 0$. $\dot{V}_{\rm PNR}<0$ is instead proved by proving (22b).

Proof of Inequality (22a): given (21)

$$\frac{\partial^2V_{\rm PNR}(\boldsymbol{x})}{\partial\boldsymbol{x}^2}=\frac{\partial^2\left(V_{\rm NR}(\overline{x})+\frac{1}{2}\eta\|\boldsymbol{x}^{\perp}\|^2\right)}{\partial\boldsymbol{x}^2}.$$

Since $0\leq\|\boldsymbol{x}^{\perp}\|^2\leq\|\boldsymbol{x}\|^2$ and

$$\frac{\partial^2V_{\rm NR}(\overline{x})}{\partial\boldsymbol{x}^2}=\frac{1}{N^2}\mathbb{1}\mathbb{1}^T\nabla^2V_{\rm NR}(\overline{x})$$

thanks to (11a) it follows immediately that (22a) holds with

$$b_5:=\min\left\{\frac{b_1}{N},\eta\right\},\quad b_6:=\max\left\{\frac{b_2}{N},\eta\right\}.$$

Proof of Inequality (22c): since the origin of $\overline{f}'$ is a minimum, it follows that $\nabla\overline{f}'(0)=0$, and thus $\overline{g}'(\mathbf{0})=0$ [cf. (14)]. Thus also $\psi(\mathbf{0})=0$, that in turn implies $\|\psi(\boldsymbol{x})\|\leq a_\psi\|\boldsymbol{x}\|$ by Assumption 5. Therefore

$$\|\phi_{\rm PNR}(\boldsymbol{x})\|\leq\|\boldsymbol{x}\|+N\|\psi(\boldsymbol{x})\|\leq(1+Na_\psi)\|\boldsymbol{x}\|=b_8\|\boldsymbol{x}\|.$$

Proof of Inequality (22b): since

$$\frac{\partial\overline{x}}{\partial\boldsymbol{x}}=\frac{1}{N}\mathbb{1}_N^T,\quad \frac{\partial\boldsymbol{x}^{\perp}}{\partial\boldsymbol{x}}=I-\frac{1}{N}\mathbb{1}_N\mathbb{1}_N^T=:\Pi$$

it follows that:

$$\begin{aligned}\frac{\partial V_{\rm PNR}}{\partial\boldsymbol{x}}\phi_{\rm PNR}(\boldsymbol{x}) &= \left(\frac{\partial V_{\rm PNR}}{\partial\overline{x}}\frac{\partial\overline{x}}{\partial\boldsymbol{x}}+\frac{\partial V_{\rm PNR}}{\partial\boldsymbol{x}^{\perp}}\frac{\partial\boldsymbol{x}^{\perp}}{\partial\boldsymbol{x}}\right)\phi_{\rm PNR}(\boldsymbol{x})\\ &= \left(\frac{\partial V_{\rm NR}(\overline{x})}{\partial\overline{x}}\frac{1}{N}\mathbb{1}_N^T+\eta(\boldsymbol{x}^{\perp})^T\Pi\right)\phi_{\rm PNR}(\boldsymbol{x}).\end{aligned}$$

Considering then (17), the definition of $\overline{x}$ and $\boldsymbol{x}^{\perp}$, and the fact that $\Pi\mathbb{1}_N=0$, it follows that:

$$\frac{\partial V_{\rm PNR}}{\partial\boldsymbol{x}}\phi_{\rm PNR}(\boldsymbol{x})=\frac{\partial V_{\rm NR}(\overline{x})}{\partial\overline{x}}\left(-\overline{x}+\psi(\boldsymbol{x})\right)+\eta(\boldsymbol{x}^{\perp})^T(-\boldsymbol{x}^{\perp})$$

Adding and subtracting $(\partial V_{\rm NR}(\overline{x})/\partial\overline{x})\psi(\boldsymbol{x}^{\parallel})$, and recalling definition (7) and equivalence (16c), since $(-\overline{x}+\psi(\boldsymbol{x}^{\parallel}))=\phi_{\rm NR}(\overline{x})$ it then follows that:

$$\begin{aligned}\frac{\partial V_{\rm PNR}}{\partial\boldsymbol{x}}\phi_{\rm PNR}(\boldsymbol{x}) &= \frac{\partial V_{\rm NR}(\overline{x})}{\partial\overline{x}}\phi_{\rm NR}(\overline{x})-\eta\|\boldsymbol{x}^{\perp}\|^2\\ &\quad+\frac{\partial V_{\rm NR}(\overline{x})}{\partial\overline{x}}\left(\psi(\boldsymbol{x})-\psi(\boldsymbol{x}^{\parallel})\right)\\ &\leq -b_3\overline{x}^2-\eta\|\boldsymbol{x}^{\perp}\|^2+b_2|\overline{x}|a_\psi\|\boldsymbol{x}-\boldsymbol{x}^{\parallel}\|\\ &= -b_3\overline{x}^2-\eta\|\boldsymbol{x}^{\perp}\|^2+b_2a_\psi|\overline{x}|\|\boldsymbol{x}^{\perp}\|\\ &\leq -\frac{b_3+\eta}{2}\left(|\overline{x}|^2+\|\boldsymbol{x}^{\perp}\|^2\right)\\ &\leq -\frac{b_3+\eta}{2}\left(N|\overline{x}|^2+\|\boldsymbol{x}^{\perp}\|^2\right)\\ &= -\frac{b_3+\eta}{2N}\|\boldsymbol{x}\|^2=-b_7\|\boldsymbol{x}\|^2\end{aligned}$$

where for obtaining the various inequalities we used the various assumptions and where the second inequality is valid for $\eta>b_2^2a_\psi^2/b_3$. ■

Proof (of Lemma 7): Proof of (26a): notice that $\phi_x(\boldsymbol{x},\boldsymbol{\chi})$ is globally defined since $[\cdot]_c$ ensures that the matrix inverse exists. Also note that, since $\overline{h}'(\boldsymbol{x})\geq cI>(c/2)I$ by Assumption 5, then there exists $r>0$ such that, for $\|\boldsymbol{x}\|+\|\boldsymbol{\chi}\|\leq r$

$$\phi_x(\boldsymbol{x},\boldsymbol{\chi})=-\boldsymbol{x}-\mathbb{1}_N\otimes x^*+\frac{\boldsymbol{\chi}^y+\mathbb{1}_N\otimes\left(\overline{g}'(\boldsymbol{x})+\overline{h}'(\boldsymbol{x})x^*\right)}{\boldsymbol{\chi}^z+\mathbb{1}_N\otimes\overline{h}'(\boldsymbol{x})}.$$

The differentiability of the elements defining ϕ_x, plus the fact that $[\cdot]_c$ acts as the identity in the neighborhood under consideration implies that ϕ_x is locally differentiable in $\|\boldsymbol{x}\|+\|\boldsymbol{\chi}\|\leq r$.

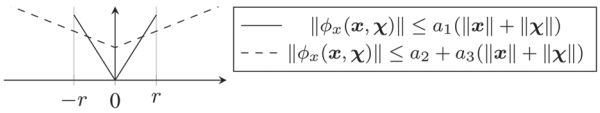

Fig. 6. Inequality (38) represents a proper cone defined in the neighborhood of radius r, while inequality (39) represents an improper cone defined in the whole domain.

In addition to this local differentiability, also observe that $\phi_x(\mathbf{0},\mathbf{0})=0$, therefore there must exist $a_1>0$ s.t.

$$\|\phi_x(\boldsymbol{x},\boldsymbol{\chi})\| \le a_1\left(\|\boldsymbol{x}\|+\|\boldsymbol{\chi}\|\right), \quad \forall\left(\|\boldsymbol{x}\|+\|\boldsymbol{\chi}\|\right)\le r. \tag{38}$$

To extend the linear inequality (38) for $(\boldsymbol{x},\boldsymbol{\chi})$ s.t. $(\|\boldsymbol{x}\|+\|\boldsymbol{\chi}\|)\ge r$ we then prove that $\phi_x(\boldsymbol{x},\boldsymbol{\chi})$ cannot grow more than linearly globally. In fact

$$\begin{aligned}\|\phi_x(\boldsymbol{x},\boldsymbol{\chi})\| &\le \|\boldsymbol{x}\|+N\|x^*\|+\frac{2}{c}\left\|\boldsymbol{\chi}^y+\mathbb{1}\otimes\left(\overline{g'}(\boldsymbol{x})+\overline{h'}(\boldsymbol{x})x^*\right)\right\|\\ &\le \|\boldsymbol{x}\|+N\|x^*\|+\frac{2}{c}\|\boldsymbol{\chi}\|\\ &\quad+\frac{2N}{c}\left(\|\overline{g'}(\boldsymbol{x})\|+\|x^*\|\|\overline{h'}(\boldsymbol{x})\|\right)\\ &\le \|\boldsymbol{x}\|+N\|x^*\|+\frac{2}{c}\|\boldsymbol{\chi}\|+\frac{2N}{c}a_g\|\boldsymbol{x}\|\\ &\quad+\frac{2N}{c}\|x^*\|\left(a_h\|\boldsymbol{x}\|+\|\overline{h'}(0)\|\right)\\ &\le a_2+a_3\left(\|\boldsymbol{x}\|+\|\boldsymbol{\chi}\|\right), \quad \forall\boldsymbol{x},\boldsymbol{\chi}\end{aligned} \tag{39}$$

where we used Assumption 5 and where a_2, a_3 are suitable positive scalars. In particular inequality (39) is valid for $(\|\boldsymbol{x}\|+\|\boldsymbol{\chi}\|)>r$. As depicted in Fig. 6, inequalities (38) and (39) define two cones, one affine (shifted by a_2) and one proper.

Therefore, combining the geometry of the two cones leads to an inequality that is defined in the whole domain. In other words, it follows that:

$$\|\phi_x(\boldsymbol{x},\boldsymbol{\chi})\| \le a_x\left(\|\boldsymbol{x}\|+\|\boldsymbol{\chi}\|\right) \quad \forall\boldsymbol{x},\boldsymbol{\chi}$$

where

$$a_x := \max\left\{a_1, \frac{a_2+a_3 r}{r}\right\}.$$

Proof of (26b): Let $\Delta(\boldsymbol{x},\boldsymbol{\chi}) := \phi_x(\boldsymbol{x},\boldsymbol{\chi})-\phi_{\mathrm{PNR}}(\boldsymbol{x})$, with ϕ_{PNR} as in (17). Then there exists a positive scalar $r>0$ such that, for all $\|\boldsymbol{\chi}\|+\|\boldsymbol{x}\|\le r$

$$\begin{aligned}\Delta(\boldsymbol{x},\boldsymbol{\chi}) &= -\mathbb{1}_N\otimes x^*+\frac{\boldsymbol{\chi}^y+\mathbb{1}_N\otimes\left(\overline{g'}(\boldsymbol{x})+\overline{h'}(\boldsymbol{x})x^*\right)}{\boldsymbol{\chi}^z+\mathbb{1}_N\otimes\overline{h'}(\boldsymbol{x})}\\ &\quad-\mathbb{1}_N\otimes\psi(\boldsymbol{x})\\ &=\frac{\boldsymbol{\chi}^y+\mathbb{1}_N\otimes\left(\overline{g'}(\boldsymbol{x})+\overline{h'}(\boldsymbol{x})x^*\right)}{\boldsymbol{\chi}^z+\mathbb{1}_N\otimes\overline{h'}(\boldsymbol{x})}\\ &\quad-\frac{\mathbb{1}_N\otimes\left(\overline{g'}(\boldsymbol{x})+\overline{h'}(\boldsymbol{x})x^*\right)}{\mathbb{1}_N\otimes\overline{h'}(\boldsymbol{x})}.\end{aligned}$$

Considerations similar to the ones that led us claim the differentiability of ϕ_x in the proof of Lemma 7 imply that $\Delta(\boldsymbol{x},\boldsymbol{\chi})$ is continuously differentiable for $\|\boldsymbol{\chi}\|+\|\boldsymbol{x}\|\le r$. Moreover, since $\Delta(\boldsymbol{x},\mathbf{0})=0$, then there exists a positive scalar $a_4>0$ s.t.

$$\|\Delta(\boldsymbol{x},\boldsymbol{\chi})\| \le a_4\|\boldsymbol{\chi}\| \qquad \|\boldsymbol{\chi}\|+\|\boldsymbol{x}\|\le r. \tag{40}$$

By using (19a) and (19b) we can then show that $\Delta(\boldsymbol{x},\boldsymbol{\chi})$ cannot grow more than linearly in the variable $\boldsymbol{\chi}$, since

$$\begin{aligned}\|\Delta(\boldsymbol{x},\boldsymbol{\chi})\| &= \left\|\frac{\boldsymbol{\chi}^y+\mathbb{1}_N\otimes\left(\overline{g'}(\boldsymbol{x})+\overline{h'}(\boldsymbol{x})x^*\right)}{\left[\boldsymbol{\chi}^z+\mathbb{1}_N\otimes\overline{h'}(\boldsymbol{x})\right]_c}\right.\\ &\qquad\left.-\mathbb{1}_N\otimes\left(x^*+\frac{\overline{g'}(\boldsymbol{x})}{\overline{h'}(\boldsymbol{x})}\right)\right\|\\ &\le\frac{2}{c}\left(\|\boldsymbol{\chi}\|+2N\|\overline{g'}(\boldsymbol{x})\|+N\|x^*\|\|\overline{h'}(\boldsymbol{x})\|\right)+N\|x^*\|\\ &\le a_5+a_6\|\boldsymbol{\chi}\|, \qquad \forall\boldsymbol{x},\boldsymbol{\chi}\end{aligned} \tag{41}$$

for suitable positive scalars a_5 and a_6. Repeating the same geometrical arguments used above we then obtain

$$\|\Delta(\boldsymbol{x},\boldsymbol{\chi})\| \le a_\Delta\|\boldsymbol{\chi}\|, \qquad \forall\boldsymbol{x},\boldsymbol{\chi}$$

with

$$a_\Delta := \max\left\{a_3, \frac{a_5+a_6 r}{r}\right\}.$$

■

Proof (of Theorem 8): For notational brevity we omit the dependence on ξ, i.e., let $\boldsymbol{x}^{\mathrm{eq}}=\boldsymbol{x}^{\mathrm{eq}}(\xi)$ and $x^{\mathrm{eq}}=x^{\mathrm{eq}}(\xi)$.

We start by assuming that there exists a $\boldsymbol{x}^{\mathrm{eq}}(\xi)$ satisfying (27) for $\|\xi\|\le r$ and prove that $\boldsymbol{x}^{\mathrm{eq}}(\xi)$ must satisfy $\boldsymbol{x}^{\mathrm{eq}}(\xi)=\mathbb{1}_N\otimes x^{\mathrm{eq}}(\xi)$ and (28). Consider then r sufficiently small. Then, since $\overline{h'}(\boldsymbol{x})>cI$ by Assumption 1

$$\left[\boldsymbol{\xi}^z+\mathbb{1}_N\otimes\overline{h'}(\boldsymbol{x})\right]_c=\boldsymbol{\xi}^z+\mathbb{1}_N\otimes\overline{h'}(\boldsymbol{x})=\mathbb{1}_N\otimes\left(\overline{h'}(\boldsymbol{x})+\xi^z\right).$$

This implies that for $\|\xi\|\le r$ we have

$$\begin{aligned}&\phi_x(\boldsymbol{x}^{\mathrm{eq}},\boldsymbol{\xi})=-\boldsymbol{x}^{\mathrm{eq}}\\ &-\mathbb{1}_N\otimes\left(x^*-\left(\xi^z+\overline{h'}(\boldsymbol{x}^{\mathrm{eq}})\right)^{-1}\left(\xi^y+\overline{g'}(\boldsymbol{x}^{\mathrm{eq}})+\overline{h'}(\boldsymbol{x}^{\mathrm{eq}})x^*\right)\right).\end{aligned}$$

Therefore $\phi_x(\boldsymbol{x}^{\mathrm{eq}},\boldsymbol{\xi})=0$ if and only if

$$x_i^{\mathrm{eq}}=-x^*+\left(\xi^z+\overline{h'}(\boldsymbol{x}^{\mathrm{eq}})\right)^{-1}\left(\xi^y+\overline{g'}(\boldsymbol{x}^{\mathrm{eq}})+\overline{h'}(\boldsymbol{x}^{\mathrm{eq}})x^*\right).$$

Since the right-hand-side is independent of i, this implies both that the $\boldsymbol{x}^{\mathrm{eq}}(\xi)$ satisfying (27) must satisfy $\boldsymbol{x}^{\mathrm{eq}}=\mathbb{1}\otimes x^{\mathrm{eq}}$, and that its expression is given by (28) (indeed (28) can be retrieved immediately from the equivalence above since $-x^*=(\xi^z+\overline{h'}(\boldsymbol{x}^{\mathrm{eq}}))^{-1}(-\xi^z x^*-\overline{h'}(\boldsymbol{x}^{\mathrm{eq}})x^*)$).

We now prove (27) by exploiting the Implicit Function Theorem [63]. If we indeed substitute the necessary condition $\boldsymbol{x}^{\mathrm{eq}}=\mathbb{1}_N\otimes x^{\mathrm{eq}}$ into the definition of $\phi_x(\boldsymbol{x}^{\mathrm{eq}},\xi)$, we obtain the parallelization of N equivalent equations of the form

$$x^{\mathrm{eq}}+x^*=\left(\overline{\overline{h'}}(x^{\mathrm{eq}})+\xi^z\right)^{-1}\left(\overline{\overline{g}}(x^{\mathrm{eq}})+\xi^y+\overline{\overline{h'}}(x^{\mathrm{eq}})x^*\right)$$

where we used properties (16a) and (16b) that lead to $\overline{h'}(\mathbb{1}_N\otimes x)=\overline{\overline{h'}}(x)$ and $\overline{g'}(\mathbb{1}_N\otimes x)=\overline{\overline{g'}}(x)$.

Moreover, Assumption 5 ensures that $\overline{\overline{h'}}(x^*)\ge cI$. Thus, for the continuity assumptions in Assumption 1, there exists a sufficiently small $r>0$ s.t. if $\|\xi^z\|\le\|\xi\|\le r$ then $\overline{\overline{h'}}(x^*)+\xi^z$ is still invertible. Therefore

$$\overline{\overline{g'}}(x^{\mathrm{eq}})+\xi^y+\overline{\overline{h'}}(x^{\mathrm{eq}})x^*=\overline{\overline{h'}}(x^{\mathrm{eq}})(x^{\mathrm{eq}}+x^*)+\xi^z(x^{\mathrm{eq}}+x^*).$$

Exploiting now the equivalence $\overline{\overline{g}}(x^{\rm eq}) = \overline{\overline{h}}(x^{\rm eq})x^{\rm eq} - \nabla\overline{f}(x^{\rm eq})$, it follows that $x^{\rm eq}$ must satisfy the following condition:

$$\nabla\overline{f}(x^{\rm eq}) - \xi^y + \xi^z(x^{\rm eq} + x^*) = 0.$$

Given Assumption 1, the left-hand side of the previous inequality is a continuously differentiable function, since

$$\frac{\partial\left(\nabla\overline{f}(x^{\rm eq}) - \xi^y + \xi^z(x^{\rm eq} + x^*)\right)}{\partial x^{\rm eq}} = \nabla^2\overline{f}(x^{\rm eq}) + \xi^z.$$

Notice moreover that if r is sufficiently small (i.e., $\|\xi^z\|$ is sufficiently small) then the differentiation is an invertible matrix, since once again $\nabla^2\overline{f}(x^*) \geq cI$ by assumption. Therefore, by the Implicit Function Theorem, $x^{\rm eq}(\xi)$ exists, is unique and continuously differentiable. ■

Proof (of Theorem 10): Proof of a): $V_{\rm PNR}(\mathbf{0}) = 0$ and $V_{\rm PNR}(\boldsymbol{x}) > 0$ for $\boldsymbol{x} \neq \mathbf{0}$ have been proved before. $\dot{V}_{\rm PNR} < 0$ is instead proved by proving (31a).

Proof of b): as for (31a), consider that, $\forall \boldsymbol{x} \in \mathbb{R}^{nN}$

$$\begin{aligned}\frac{\partial V_{\rm PNR}}{\partial \boldsymbol{x}}\phi'_x(\boldsymbol{x},\xi) &= \frac{\partial V_{\rm PNR}}{\partial \boldsymbol{x}}\phi'_x(\boldsymbol{x},0) + \frac{\partial V_{\rm PNR}}{\partial \boldsymbol{x}} \\ &\quad\times(\phi'_x(\boldsymbol{x},\xi) - \phi'_x(\boldsymbol{x},0)) \\ &\leq \frac{\partial V_{\rm PNR}}{\partial \boldsymbol{x}}\phi_{\rm PNR}(\boldsymbol{x}) + \left\|\frac{\partial V_{\rm PNR}}{\partial \boldsymbol{x}}\right\| \\ &\quad\times\|\phi'_x(\boldsymbol{x},\xi) - \phi'_x(\boldsymbol{x},0)\| \\ &\leq -b_7\|\boldsymbol{x}\|^2 + b_6\|\boldsymbol{x}\|a_\xi\|\xi\|\|\boldsymbol{x}\| \\ &\leq -(b_7 - b_6a_\xi r)\|\boldsymbol{x}\|^2 \leq -b'_7\|\boldsymbol{x}\|^2.\end{aligned}$$

Notice that this inequality is meaningful for $r < (b_7/b_6a_\xi)$.

As for (31b), consider that, $\forall \boldsymbol{x} \in \mathbb{R}^{nN}$

$$\begin{aligned}\|\phi'_x(\boldsymbol{x},\xi)\| &\leq \|\phi'_x(\boldsymbol{x},0)\| + \|\phi'_x(\boldsymbol{x},\xi) - \phi'_x(\boldsymbol{x},0)\| \\ &\leq (b_8 + a_\xi r)\|\boldsymbol{x}\| \leq b'_8\|\boldsymbol{x}\|.\end{aligned}$$

■

Proof (of Theorem 11): The miminizer of the global cost function is easily seen to be $x^* = (\sum_i A_i)^{-1}(\sum_i A_i d_i)$ from which it follows that $\overline{f}'(x) = (1/N)x^T Ax$. Clearly $\overline{f}(x)$ satisfies Assumption 1 since $\nabla^2\overline{f}(x) = (1/N)A > 0$ is independent of x. Considering then $h'_i(x) = \nabla^2 f'_i(x) = A_i$ it follows after some suitable simplifications that:

$$\begin{aligned}\overline{h'}(\boldsymbol{x}) &= \frac{1}{N}A \\ g'_i(x) &= A_i x - A_i(x + x^* - d_i) = A_i(d_i - x^*) \\ g'(\boldsymbol{x}) - g'(\boldsymbol{x}') &= 0 \\ \overline{g'}(\boldsymbol{x}) &= \frac{1}{N}\left(\sum_i A_i d_i - \sum_i A_i x^*\right) = 0 \\ h'(\boldsymbol{x}) - h'(\boldsymbol{x}') &= 0 \\ \psi(\boldsymbol{x}) &= \overline{h}^{-1}(\boldsymbol{x})\overline{g}(\boldsymbol{x}) = 0 \\ x^{\rm eq}(\xi) &= \left(\frac{1}{N}A + \xi^z\right)^{-1}(\xi^y - \xi^z x^*) \\ \phi'_x(\boldsymbol{x},\xi) &= \phi'_x(\boldsymbol{x},0) = -\boldsymbol{x}\end{aligned}$$

where in the last equivalence we exploited definition (28). Thus also the other assumptions are satisfied. ■

Proof (of Theorem 12): The proof considers the system as an autonomous singularly perturbed system, and proceeds as follows: *a*) show that x^* is an equilibrium; *b*) perform a change of variables; *c*) construct a Lyapunov function for the boundary layer system; *d*) construct a Lyapunov function for the reduced system; *e*) join the two Lyapunov functions into one, and show (by cascading the previously introduced Lemmas and Theorems) that the complete system (43) converges to x^* while satisfying the hypotheses of Theorem 2. By doing so it follows that (42), i.e., Algorithm 1, is exponentially stable.

For notational simplicity we let $\boldsymbol{x}^* := \mathbb{1}_N \otimes x^*$. We also use all the notation collected in Section II.

- *Discrete to continuous dynamics)* The dynamics of Algorithm 1 can be written in state space as

$$\begin{cases}\boldsymbol{v}(k) = g\left(\boldsymbol{x}(k-1)\right) \\ \boldsymbol{w}(k) = h\left(\boldsymbol{x}(k-1)\right) \\ \boldsymbol{y}(k) = P\left[\boldsymbol{y}(k-1) + g\left(\boldsymbol{x}(k-1)\right) - \boldsymbol{v}(k-1)\right] \\ \boldsymbol{z}(k) = P\left[\boldsymbol{z}(k-1) + h\left(\boldsymbol{x}(k-1)\right) - \boldsymbol{w}(k-1)\right] \\ \boldsymbol{x}(k) = (1-\varepsilon)\boldsymbol{x}(k-1) + \varepsilon\frac{\boldsymbol{y}(k-1)}{[\boldsymbol{z}(k-1)]_c}\end{cases} \tag{42}$$

with suitable initial conditions. (42) can then be interpreted as the forward-Euler discretization of

$$\begin{cases}\varepsilon\dot{\boldsymbol{v}}(t) = -\boldsymbol{v}(t) + g\left(\boldsymbol{x}(t)\right) \\ \varepsilon\dot{\boldsymbol{w}}(t) = -\boldsymbol{w}(t) + h\left(\boldsymbol{x}(t)\right) \\ \varepsilon\dot{\boldsymbol{y}}(t) = -K\boldsymbol{y}(t) + (I-K)\left[g\left(\boldsymbol{x}(t)\right) - \boldsymbol{v}(t)\right] \\ \varepsilon\dot{\boldsymbol{z}}(t) = -K\boldsymbol{z}(t) + (I-K)\left[h\left(\boldsymbol{x}(t)\right) - \boldsymbol{w}(t)\right] \\ \dot{\boldsymbol{x}}(t) = -\boldsymbol{x}(t) + \frac{\boldsymbol{y}(t)}{[\boldsymbol{z}(t)]_c}\end{cases} \tag{43}$$

with null initial conditions, where ε is the discretization time interval and $K := I - P$. Notice that, as for P, if n is the dimension of the local costs then $P = P' \otimes I_n$ with P' a doubly-stochastic average consensus matrix. Nonetheless for brevity we will omit the superscripts $'$.

- *b*) let

$$\begin{aligned}\boldsymbol{v}' &:= \boldsymbol{v} - g(\boldsymbol{x}) \\ \boldsymbol{w}' &:= \boldsymbol{w} - h(\boldsymbol{x}) \\ \boldsymbol{y}' &:= \boldsymbol{y} - \boldsymbol{v}' - \mathbb{1}_N \otimes \overline{g}(\boldsymbol{x}) \\ \boldsymbol{z}' &:= \boldsymbol{z} - \boldsymbol{w}' - \mathbb{1}_N \otimes \overline{h}(\boldsymbol{x}) \\ \boldsymbol{x}' &:= \boldsymbol{x} - \boldsymbol{x}^* \\ \phi_g(\boldsymbol{x}') &:= \frac{\partial g}{\partial \boldsymbol{x}'} - \mathbb{1}_N \otimes \frac{\partial \overline{g}}{\partial \boldsymbol{x}'} \\ \phi_h(\boldsymbol{x}') &:= \frac{\partial h}{\partial \boldsymbol{x}'} - \mathbb{1}_N \otimes \frac{\partial \overline{h}}{\partial \boldsymbol{x}'} \\ \phi_x(\boldsymbol{x}',\chi) &:= -\boldsymbol{x}'(t) - \boldsymbol{x}^* + \\ &\quad + \frac{\boldsymbol{y}'(t) + \boldsymbol{v}'(t) + \mathbb{1}_N \otimes \overline{g}\left(\boldsymbol{x}'(t) + \boldsymbol{x}^*\right)}{\left[\boldsymbol{z}'(t) + \boldsymbol{w}'(t) + \mathbb{1}_N \otimes \overline{h}\left(\boldsymbol{x}'(t) + \boldsymbol{x}^*\right)\right]_c}\end{aligned}$$

with $\boldsymbol{\chi} := (\boldsymbol{v}', \boldsymbol{w}', \boldsymbol{y}', \boldsymbol{z}')$, so that (43) becomes

$$\begin{cases}\varepsilon\dot{\boldsymbol{v}}'(t) = -\boldsymbol{v}'(t) - \varepsilon\frac{\partial g}{\partial \boldsymbol{x}'}\dot{\boldsymbol{x}}'(t) \\ \varepsilon\dot{\boldsymbol{w}}'(t) = -\boldsymbol{w}'(t) - \varepsilon\frac{\partial h}{\partial \boldsymbol{x}'}\dot{\boldsymbol{x}}'(t) \\ \varepsilon\dot{\boldsymbol{y}}'(t) = -K\boldsymbol{y}'(t) + \varepsilon\phi_g(\boldsymbol{x}')\dot{\boldsymbol{x}}'(t) \\ \varepsilon\dot{\boldsymbol{z}}'(t) = -K\boldsymbol{z}'(t) + \varepsilon\phi_h(\boldsymbol{x}')\dot{\boldsymbol{x}}'(t) \\ \dot{\boldsymbol{x}}'(t) = \phi_x(\boldsymbol{x}', \boldsymbol{\chi}')\end{cases} \tag{44}$$

with initial conditions

$$\begin{cases} \boldsymbol{v}'(0) = \boldsymbol{v}(0) - g(\boldsymbol{x}(0)) \\ \boldsymbol{w}'(0) = \boldsymbol{w}(0) - h(\boldsymbol{x}(0)) \\ \boldsymbol{y}'(0) = \boldsymbol{y}(0) - \boldsymbol{v}(0) + g^{\perp}(\boldsymbol{x}(0)) \\ \boldsymbol{z}'(0) = \boldsymbol{z}(0) - \boldsymbol{w}(0) + h^{\perp}(\boldsymbol{x}(0)) \\ \boldsymbol{x}'(0) = \boldsymbol{x}(0) - \boldsymbol{x}^* \end{cases}$$

where $g^{\perp}(\boldsymbol{x}) := g(\boldsymbol{x}) - \mathbb{1}_N \otimes \overline{g}(\boldsymbol{x})$ (equivalent definition for $h^{\perp}$). Notice that (44) has the origin as an equilibrium point. Moreover this dynamics exploits the function ϕ_x defined in (24), with $\boldsymbol{\chi}^y = \boldsymbol{y}' + \boldsymbol{v}'$, and $\boldsymbol{\chi}^z = \boldsymbol{z}' + \boldsymbol{w}'$.

The next step is to exploit the structure of K (more precisely, the fact that it contains an average consensus matrix) to reduce the dynamics, i.e., to eliminate the dynamics of the average since the latter does not change in time. To this aim, we analyze the behavior of the average of the y_i's, i.e., the behavior of $(\mathbb{1}_N^T \otimes I_n)\dot{\boldsymbol{y}}'$. To this point, consider the third equation in (44). Recalling that $(A \otimes B)(C \otimes D) = AB \otimes CD$, and exploiting the fact that $\mathbb{1}_N^T P' = 0$, we notice that $(\mathbb{1}_N^T \otimes I_n)\ K = 0$. Moreover, from the definitions of g and $\overline{g}$

$$\left(\mathbb{1}_N^T \otimes I_n\right) \frac{\partial g(\boldsymbol{x}')}{\partial \boldsymbol{x}'} = N \frac{\partial \overline{g}(\boldsymbol{x}')}{\partial \boldsymbol{x}'}.$$

Since $N = \mathbb{1}_N^T \mathbb{1}_N$, it follows also that:

$$\left(\mathbb{1}_N^T \otimes I_n\right) \phi_g(\boldsymbol{x}') = 0$$

for all $t \geq 0$, i.e., $\mathbb{1}^T \boldsymbol{y}'(t) = \mathbb{1}^T \boldsymbol{y}'(0) \equiv 0$. Similarly it is possible to show that $\boldsymbol{z}'(t) \equiv 0$. This eventually implies that

$$\boldsymbol{y}'^{\parallel}(t) = 0 \qquad \boldsymbol{z}'^{\parallel}(t) = 0 \qquad \forall t$$

that means, recalling that $\boldsymbol{y}' = \boldsymbol{y}'^{\parallel} + \boldsymbol{y}'^{\perp}$ and $\boldsymbol{z}' = \boldsymbol{z}'^{\parallel} + \boldsymbol{z}'^{\perp}$, that (44) can be equivalently rewritten as

$$\begin{cases} \varepsilon \dot{\boldsymbol{v}}'(t) = -\boldsymbol{v}'(t) - \varepsilon \frac{\partial g}{\partial \boldsymbol{x}'} \phi_x(\boldsymbol{x}', \boldsymbol{\chi}') \\ \varepsilon \dot{\boldsymbol{w}}'(t) = -\boldsymbol{w}'(t) - \varepsilon \frac{\partial h}{\partial \boldsymbol{x}'} \phi_x(\boldsymbol{x}', \boldsymbol{\chi}') \\ \varepsilon \dot{\boldsymbol{y}}'^{\perp}(t) = -K\boldsymbol{y}'^{\perp}(t) + \varepsilon \phi_g(\boldsymbol{x}') \phi_x(\boldsymbol{x}', \boldsymbol{\chi}') \\ \varepsilon \dot{\boldsymbol{z}}'^{\perp}(t) = -K\boldsymbol{z}'^{\perp}(t) + \varepsilon \phi_h(\boldsymbol{x}') \phi_x(\boldsymbol{x}', \boldsymbol{\chi}') \\ \dot{\boldsymbol{x}}'(t) = \phi_x(\boldsymbol{x}', \boldsymbol{\chi}') \end{cases} \tag{45}$$

where now $\boldsymbol{\chi}' := (\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{y}'^{\perp}, \boldsymbol{z}'^{\perp})$ and where the novel initial conditions for the changed variables are

$$\begin{cases} \boldsymbol{y}'^{\perp}(0) = \boldsymbol{y}^{\perp}(0) - \boldsymbol{v}^{\perp}(0) + g^{\perp}(\boldsymbol{x}(0)) \\ \boldsymbol{z}'^{\perp}(0) = \boldsymbol{z}^{\perp}(0) - \boldsymbol{w}^{\perp}(0) + h^{\perp}(\boldsymbol{x}(0)) \end{cases}$$

- *c*) the boundary layer of (45) is computed by setting $\boldsymbol{x}'(t) = \boldsymbol{x}'$. Since a constant $\boldsymbol{x}'$ implies $\dot{\boldsymbol{x}}' = \phi_x = 0$, this boundary layer reduces to a linear system globally exponentially converging to the origin. Notice that this implies that, in the original coordinates system

$$v = g(\boldsymbol{x}),\ w = h(\boldsymbol{x}),\ y = \mathbb{1}_N \otimes \overline{g}(\boldsymbol{x}),\ z = \mathbb{1}_N \otimes \overline{h}(\boldsymbol{x}).$$

  In the novel coordinates system we thus consider, as a Lyapunov function, $(1/2)\|\boldsymbol{\chi}'\|^2$.
- *d*) the reduced system of (45) is computed by plugging $\boldsymbol{\chi}' = \mathbf{0}$ into the equations (i.e., by setting $\boldsymbol{v}'(t) = \mathbf{0}$, $\boldsymbol{w}'(t) = \mathbf{0}$, $\boldsymbol{y}'^{\perp}(t) = \mathbf{0}$, $\boldsymbol{z}'^{\perp}(t) = \mathbf{0}$). Defining then

$$f_i'(\boldsymbol{x}') := f_i(\boldsymbol{x}' + \boldsymbol{x}^*), \qquad h_i'(\boldsymbol{x}') := h_i(\boldsymbol{x}' + \boldsymbol{x}^*)$$

we obtain

$$\begin{aligned} \dot{\boldsymbol{x}}'(t) &= -\boldsymbol{x}'(t) - \boldsymbol{x}^* + \mathbb{1}_N \otimes \frac{\overline{g'}(\boldsymbol{x}'(t))}{\overline{h'}(\boldsymbol{x}'(t))} \\ &= -\boldsymbol{x}'(t) - \boldsymbol{x}^* + \mathbb{1}_N \\ &\quad \otimes \frac{\overline{h'}(\boldsymbol{x}'(t))(\boldsymbol{x}'(t) + \boldsymbol{x}^*) - \nabla f'(\boldsymbol{x}'(t))}{\overline{h'}(\boldsymbol{x}'(t))} \\ &= -\boldsymbol{x}'(t) + \mathbb{1}_N \otimes \frac{\overline{h'}(\boldsymbol{x}'(t))\boldsymbol{x}'(t) - \nabla f'(\boldsymbol{x}'(t))}{\overline{h'}(\boldsymbol{x}'(t))} \\ &= -\boldsymbol{x}'(t) + \mathbb{1}_N \otimes \psi(\boldsymbol{x}'(t)) \\ &= \phi_{\mathrm{PNR}}(\boldsymbol{x}') \end{aligned}$$

  where ψ and ϕ_{PNR} are the functions defined in (15) and (17), respectively. Thus the reduced system, thanks to Theorem 6, admits $\boldsymbol{x}^*$ as a global exponentially stable equilibrium, and admits V_{PNR} in (21) as a Lyapunov function.
- *e*) we now notice that the interconnection of the boundary layer and reduced systems maintains the global stability, since their Lyapunov functions are quadratic type. Thus (see [46, pp. 453]) the global system is asymptotically globally stable. To check that forward-Euler discretizations of the system preserve these stability properties we then consider as a global Lyapunov function the function

$$V(\boldsymbol{x}', \boldsymbol{\chi}') = (1-d)V_{\mathrm{PNR}}(\boldsymbol{x}') + d\frac{1}{2}\|\boldsymbol{\chi}'\|^2$$

  that is clearly positive definite for every $d \in (0,1)$, and prove that inequalities (5) of Theorem 2 are satisfied.

Proof That (5a) *Holds*: from (22a) and the structure of V it follows immediately that:

$$((1-d)b_5 + d)\, I \leq \nabla^2 V(\boldsymbol{x}', \boldsymbol{\chi}') \leq ((1-d)b_6 + d)\, I.$$

Proof That (5c) *Holds*: applying (20) and (26a) to (45) it follows that (5c) holds with

$$a_4 = a_V := \max\{1 + 2\varepsilon a_g a_x,\ 1 + 2\varepsilon a_h a_x,\ a_x\}.$$

Proof That (5b) *Holds*: the part relative to the slow dynamics is already characterized by (31a). For the part relative to the fast dynamics, since $(\partial(1/2)\|\boldsymbol{\chi}\|^2)/\partial\boldsymbol{\chi} = \boldsymbol{\chi}^T$ to check that (5b) corresponds to check the negativity of the terms

$$\begin{aligned} &-\boldsymbol{v}'^T\boldsymbol{v}' - \varepsilon \boldsymbol{v}'^T \frac{\partial g}{\partial \boldsymbol{x}'} \phi_x(\boldsymbol{x}', \boldsymbol{\chi}') \\ &-\boldsymbol{w}'^T\boldsymbol{w}' - \varepsilon \boldsymbol{w}'^T \frac{\partial h}{\partial \boldsymbol{x}'} \phi_x(\boldsymbol{x}', \boldsymbol{\chi}') \\ &-(\boldsymbol{y}'^{\perp})^T K\boldsymbol{y}' \perp +\varepsilon \boldsymbol{y}'^{\perp T} \phi_g(\boldsymbol{x}') \phi_x(\boldsymbol{x}', \boldsymbol{\chi}') \\ &-(\boldsymbol{z}'^{\perp})^T K\boldsymbol{z}' \perp +\varepsilon \boldsymbol{z}'^{\perp T} \phi_h(\boldsymbol{x}') \phi_x(\boldsymbol{x}', \boldsymbol{\chi}'). \end{aligned}$$

These terms can then be majorized using (20) and (26a). E.g., the third term can be majorized with

$$-\sigma(P)\|\boldsymbol{y}'^{\perp}\|^2 + 2\varepsilon a_g a_x \|\boldsymbol{y}'^{\perp}\| (\|\boldsymbol{x}\| + \|\boldsymbol{\chi}\|)$$

where $\sigma(P)$ is the spectral gap of P. Applying similar concepts also to the other terms it follows that (5b) holds with

$$a_3 = \min\{\sigma(P) - 2\varepsilon a_g a_x,\ \sigma(P) - 2\varepsilon a_h a_x\}.$$

■

Proof (of Theorem 13): The proof is identical to the one of Theorem 12 with the exception that the substitution is now $\boldsymbol{x}'' = \boldsymbol{x} - \boldsymbol{x}^* - \mathbb{1}_N \otimes \Psi(\xi^y, \xi^z)$. Indeed one can prove the stability of the novel system using the same Lyapunov function of Theorem 12. Notice that we are ensured that there exists a sufficiently small neighborhood of the origin for which the function Ψ exists due to the smoothness conditions in Assumption 1. ■

Proof (of Theorem 14): The proof is the local version of the one in Theorem 13. Indeed the local versions of Assumptions 1, 5, and 9 always hold, i.e., they hold when considering $\boldsymbol{x}$ s.t. $\|\boldsymbol{x}\| \leq r'$, and one can thus repeat that reasonings using local perspectives. ■

Proof (of Theorem 15): Consider for simplicity the scalar case. Let $y^* := (1/N)\sum_i A_i d_i$ and $z^* := (1/N)\sum_i A_i$, so that $x^* = y^*/z^*$. Since $\boldsymbol{y}(k+1) = P\boldsymbol{y}(k)$ and $\boldsymbol{z}(k+1) = P\boldsymbol{z}(k)$, given the assumptions on P, there exist positive α_y, α_z independent of $\boldsymbol{x}(0)$ s.t. $|y_i(k) - y^*| \leq \alpha_y(\rho(P))^k$ and $|z_i(k) - z^*| \leq \alpha_z(\rho(P))^k$. The claim thus follows considering that $x_i(k) = y_i(k)/[z_i(k)]_c$ and that, since the elements of P are non negative, all the $z_i(k)$ are non smaller than c for all $k \geq 0$ (i.e., the operator $[\cdot]_c$ is always performing as the identity operator). ■

REFERENCES

[1] F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-Raphson consensus for distributed convex optimization," in *Proc. IEEE Conf. Decision Control & Eur. Control Conf.*, Dec. 2011, pp. 5917–5922.

[2] F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato, "Multidimensional Newton-Raphson consensus for distributed convex optimization," in *Proc. Amer. Control Conf.*, 2012, pp. 1079–1084.

[3] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions*. New York: Springer-Verlag, 1985.

[4] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Belmont, MA: Athena Scientific, 2003.

[5] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[6] J. N. Tsitsiklis, "Problems in Decentralized Decision Making and Computation," Ph.D. dissertation, MIT, Cambridge, MA, 1984.

[7] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods.*. Belmont, MA: Athena Scientific, 1997.

[8] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Belmont, MA, USA: Athena Scientific, 1998.

[9] M. Bürger, G. Notarstefano, F. Bullo, and F. Allgöwer, "A distributed simplex algorithm for degenerate linear programs and multi-agent assignments," *Automatica*, vol. 48, no. 9, pp. 2298–2304, 2012.

[10] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods.*. Boston, MA: Academic Press, 1982.

[11] M. R. Hestenes, "Multiplier and gradient methods," *J. Optim. Theory Appl.*, vol. 4, no. 5, pp. 303–320, 1969.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," Statistics Dept., Stanford Univ., Stanford, CA, Tech. Rep., 2010.

[13] T. Erseghe, D. Zennaro, E. Dall'Anese, and L. Vangelista, "Fast consensus by the alternating direction multipliers method," *IEEE Trans. Signal Processing*, vol. 59, no. 11, pp. 5523–5537, Nov. 2011.

[14] B. He and X. Yuan, "On the O(1/t) convergence rate of alternating direction method," *SIAM J. Numer. Anal.*, 2011. [Online]. Available: http://www.optimization-online.org/DB_FILE/2011/09/3157.pdf

[15] W. Deng and W. Yin, "On the Global and Linear Convergence of the Generalized Alternating Direction Method of Multipliers," DTIC Document, Tech. Rep., 2012.

[16] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *Proc. IEEE Conf. Decision Control*, 2012, pp. 5445–5450.

[17] J. a. Mota, J. Xavier, P. Aguiar, and M. Püschel, "Distributed ADMM for model predictive control and congestion control," in *Proc. IEEE Conf. Decision Control*, 2012, pp. 5110–5115.

[18] D. Jakovetić, J. a. Xavier, and J. M. F. Moura, "Cooperative convex optimization in networked systems: Augmented lagrangian algorithms with directed gossip communication," *IEEE Trans. Signal Processing*, vol. 59, no. 8, pp. 3889–3902, Aug. 2011.

[19] V. F. Dem'yanov and L. V. Vasil'ev, *Nondifferentiable Optimization*. New York: Springer-Verlag, 1985.

[20] B. Johansson, "On Distributed Optimization in Networked Systems," Ph.D. dissertation, KTH Royal Institute of Technology, Stockholm, Sweden, 2008.

[21] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inform. Theory/ACM Trans. Netw.*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[22] A. Ribeiro, "Ergodic stochastic optimization algorithms for wireless communication and networking," *IEEE Trans. Signal Processing*, vol. 58, no. 12, pp. 6369–6386, Dec. 2010.

[23] A. Nedić and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM J. Optim.*, vol. 12, no. 1, pp. 109–138, 2001.

[24] A. Nedić, D. Bertsekas, and V. Borkar, "Distributed asynchronous incremental subgradient methods," *Studies Computat. Math.*, vol. 8, pp. 381–407, 2001.

[25] A. Nedić and A. Ozdaglar, "Approximate primal solutions and rate analysis for dual subgradient methods," *SIAM J. Optim.*, vol. 19, no. 4, pp. 1757–1780, 2008.

[26] K. C. Kiwiel, "Convergence of approximate and incremental subgradient methods for convex optimization," *SIAM J. Optim.*, vol. 14, no. 3, pp. 807–840, 2004.

[27] D. Blatt, A. Hero, and H. Gauchman, "A convergent incremental gradient method with a constant step size," *SIAM J. Optim.*, vol. 18, no. 1, pp. 29–51, 2007.

[28] L. Xiao and S. Boyd, "Optimal scaling of a gradient method for distributed resource allocation," *J. Optim. Theory Appl.*, vol. 129, no. 3, pp. 469–488, 2006.

[29] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Incremental stochastic subgradient algorithms for convex optimzation," *SIAM J. Optim.*, vol. 20, no. 2, pp. 691–717, 2009.

[30] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, 2009.

[31] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM J. Optim.*, vol. 20, no. 3, pp. 1157–1170, 2009.

[32] I. Lobel, A. Ozdaglar, and D. Feijer, "Distributed multi-agent optimization with state-dependent communication," *Math. Programming*, vol. 129, no. 2, pp. 255–284, 2011.

[33] A. Nedić, "Asynchronous broadcast-based convex optimization over a network," *IEEE Trans. Autom. Control*, vol. 56, no. 6, pp. 1337–1351, Jun. 2010.

[34] E. Ghadimi, I. Shames, and M. Johansson, "Accelerated gradient methods for networked optimization," in *Proc. Amer. Control Conf.*, 2011, pp. 1668–1673.

[35] A. Jadbabaie, A. Ozdaglar, and M. Zargham, "A distributed newton method for network optimization," in *Proc. IEEE Conf. Decision Control*, 2009, pp. 2736–2741.

[36] M. Zargham, A. Ribeiro, A. Ozdaglar, and A. Jadbabaie, "Accelerated dual descent for network optimization," *IEEE Trans. Autom. Control*, vol. 59, no. 4, pp. 905–920, 2013.

[37] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A distributed newton method for network utility maximization," in *Proc. IEEE Conf. Decision Control*, 2010, pp. 1816–1821.

[38] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, "Inexact newton methods," *SIAM J. Numer.*, vol. 19, no. 2, pp. 400–408, 1982.

[39] A. Nedić, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.

[40] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 151–164, 2012.

[41] C. Fischione, "F-lipschitz optimization with wireless sensor networks applications," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2319–2331, 2011.

[42] J. Wang and N. Elia, "Control approach to distributed optimization," in *Proc. 48th Annu. Allerton Conf.*, Allerton, IL, USA, Sep. 2010, vol. 1, no. 1, pp. 557–561, IEEE.

[43] N. Freris and A. Zouzias, "Fast distributed smoothing for network clock synchronization," in *Proc. IEEE Conf. Decision Control*, 2012, pp. 1411–1416.

[44] I. Necoara and V. Nedelcu, "Distributed dual gradient methods and error bound conditions," 2014. [Online]. Available: http://arxiv.org/abs/1401.4398

[45] F. Garin and L. Schenato, *A Survey on Distributed Estimation and Control Applications Using Linear Consensus Algorithms*, vol. 406. New York: Springer, 2011, pp. 75–107.

[46] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 2001.

[47] A. Nedic and A. Olshevsky, "Distributed optimization over time-varying directed graphs," in *Proc. IEEE 52nd Annu. Conf. Decision Control (CDC)*, 2013, pp. 6855–6860.

[48] F. Fagnani and S. Zampieri, "Randomized consensus algorithms over large scale networks," *IEEE J. Selected Areas Commun.*, vol. 26, no. 4, pp. 634–649, May 2008.

[49] A. D. Domínguez-García, C. N. Hadjicostis, and N. H. Vaidya, "Distributed Algorithms for Consensus and Coordination in the Presence of Packet-Dropping Communication Links Part I: Statistical Moments Analysis Approach," Coordinated Sciences Laboratory, University of Illinois at Urbana-Champaign, Tech. Rep., 2011.

[50] P. Kokotović, H. K. Khalil, and J. O'Reilly, *Singular Perturbation Methods in Control: Analysis and Design*. Philadelphia, PA: SIAM, 1999, ser. Classics in applied mathematics.

[51] K. Tanabe, "Global analysis of continuous analogues of the Levenberg-Marquardt and Newton-Raphson methods for solving nonlinear equations," *Annals Inst. Stat. Math.*, vol. 37, no. 1, pp. 189–203, 1985.

[52] R. Hauser and J. Nedić, "The continuous newton-raphson method can look ahead," *SIAM J. Optim.*, vol. 15, pp. 915–925, 2005.

[53] T. Sahai, A. Speranzon, and A. Banaszuk, "Hearing the clusters of a graph: A distributed algorithm," *Automatica*, vol. 48, no. 1, pp. 15–24, Jan. 2012.

[54] S. Becker and Y. Le Cun, "Improving the Convergence of Back-Propagation Learning With Second Order Methods," Univ. Toronto, Tech. Rep., Sep. 1988.

[55] S. Athuraliya and S. H. Low, "Optimization flow control with newton like algorithm," *Telecommun. Syst.*, vol. 15, pp. 345–358, 2000.

[56] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: John Hopkins Univ. Press, 1996.

[57] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, Prediction*, 2nd ed. New York: Springer, 2001.

[58] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distrib. Comp.*, vol. 67, no. 1, pp. 33–46, Jan. 2007.

[59] S. Muthukrishnan, B. Ghosh, and M. H. Schultz, "First and second order diffusive methods for rapid, coarse, distributed load balancing," *Theory Comp. Syst.*, vol. 31, no. 4, pp. 331–354, 1998.

[60] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in Ad Hoc WSNs with noisy links—part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Processing*, vol. 56, pp. 350–364, Jan. 2008.

[61] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "On the optimal step-size selection for the alternating direction method of multipliers," *Necsys*, 2012. [Online]. Available: https://people.kth.se/~andretei/papers/gtsj_NECSYS12.pdf

[62] F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato, "Asynchronous newton-raphson consensus for distributed convex optimization," *Necsys*, 2012. [Online]. Available: http://automatica.dei.unipd.it/tl_files/utenti/lucaschenato/Papers/Conference/Necsys12_Asynchronous_Newton_Raphson_Consensus.pdf

[63] L. Kudryavtsev, *Encyclopedia of Mathematics*. New York: Springer, 2001.

Damiano Varagnolo (M'11) received the M.S. degree in automation engineering and the Ph.D. degree in information engineering from the University of Padova, Italy, in 2005 and 2011, respectively.

He was a Research Engineer at Tecnogamma S.p.A., Treviso, Italy, from 2006 to 2007 and visited UC Berkeley as a Scholar Researcher in 2010. From March 2012 to December 2013 he worked as a Post-Doctoral Scholar at the KTH, Royal Institute of Technology, Stockholm, Sweden. Currently, he is Associate Senior Lecturer at LTU, Luleå University of Technology. His interests include distributed optimization, distributed estimation, identification and control of HVAC systems.

Filippo Zanella (M'06) was born in Treviso, Italy, in 1983. He received the B.S. and M.S. degrees in automation engineering and the Ph.D. degree in information engineering from the University of Padova, Padova, Italy, in 2005, 2008, and 2013, respectively.

He has been a Visiting Student Researcher at UC Berkeley in 2011 and at UC Santa Barbara in 2012. His research interests are in wireless cameras/sensors networks and mobile networks with emphasis on distributed control, estimation and optimization.

Dr. Zanella was a Staff Member of the IEEE Student Branch of the University of Padova from 2006 to 2008.

Angelo Cenedese (M'12) received the M.S. and the Ph.D. degrees from the University of Padova, Padova, Italy, in 1999 and 2004, respectively.

He is currently an Assistant Professor with the Department of Information Engineering and member of the Human Inspired Tecnologies Research Center and the Research Center on Fusion. He has been and he is currently involved in several projects on distributed systems (sensor and actor networks, camera networks), control of complex systems (adaptive optics systems, fusion devices), funded by European and National government institutions and industries, with different roles of participant and/or principal investigator. He coauthored more than 90 papers and holds three patents in the area of sensor/actor networks and videosurveillance. His research interests include system modeling, control theory and its applications, sensor and actuator networks, home automation systems.

Gianluigi Pillonetto (M'03) was born in Montebelluna (TV), Italy on January 21, 1975. He received the Doctoral degree in computer science engineering (with highest honors) from the University of Padova, Padova, italy, in 1998 and the PhD degree in bioengineering from the Polytechnic of Milan, Milan, Italy, in 2002.

In 2000 and 2002, he was Visiting Scholar and Visiting Scientist, respectively, at the Applied Physics Laboratory, University of Washington, Seattle. In 2005, he became Assistant Professor of Control and Dynamic Systems at the Department of Information Engineering, University of Padova where he currently serves as an Associate Professor. He is an Associate Editor of *Automatica* and *Systems and Control Letters*. His research interests are in the field of system identification and machine learning.

Luca Schenato (SM'15) received the Dr. Eng. degree in electrical engineering from the University of Padova, Padova, Italy, in 1999 and the Ph.D. degree in electrical engineering and computer sciences from UC Berkeley, in 2003.

He held a post-doctoral position in 2004 and a Visiting Professor position from 2013 to 2014 at U.C. Berkeley. Currently, he is Associate Professor at the Information Engineering Department, University of Padova. His interests include networked control systems, multi-agent systems, wireless sensor networks, smart grids and cooperative robotics.

Dr. Schenato received the 2004 Researchers Mobility Fellowship from the Italian Ministry of Education, University and Research (MIUR), and the 2006 Eli Jury Award in U.C. Berkeley and the EUCA European Control Award in 2014. He served as Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL from 2010 to 2014.