Analysis of Newton-Raphson Consensus for multi-agent convex optimization under asynchronous and lossy communications

Ruggero Carli Giuseppe Notarstefano Luca Schenato Damiano Varagnolo

***Abstract*— We extend a multi-agent convex-optimization algorithm named Newton-Raphson consensus to a network scenario that involves directed, asynchronous and lossy communications. We theoretically analyze the stability and performance of the algorithm and, in particular, provide sufficient conditions that guarantee local exponential convergence of the node-states to the global centralized minimizer even in presence of packet losses. Finally, we complement the theoretical analysis with numerical simulations that compare the performance of the Newton-Raphson consensus against asynchronous implementations of distributed subgradient methods on real datasets extracted from open-source databases.**

I. Introduction

Distributed optimization algorithms are important building blocks in several estimation and control problems arising in peer-to-peer networks. To cope with real-world requirements, these algorithms need to be designed to work under asynchronous, directed, faulty and time-varying communications. Unfortunately, despite being the literature on distributed optimization already rich, most of the existing contributions have been proved to work in networks whose communication schemes follow synchronous, undirected, and often time-invariant information exchange mechanisms.

Early references on distributed optimization algorithms involve primal subgradient iterations [1]. Sub-gradient based algorithms have the advantage of being simple to implement and suitable for non-differentiable cost functions. Moreover, they recently have been extended to directed and time-varying communication [2], [3]. However, these algorithms exhibit sub-linear convergence rates.

More recently, primal subgradient strategies have been proposed with guaranteed convergence in directed communication graphs [4] and in time-varying event-triggered communication schemes [5]. However, these schemes require weight-balanced graphs, an assumption that is difficult to be satisfied in the presence of lossy communication.

A second set of contributions is based on dual decomposition schemes. The related literature is very large and we refer to [6] for a comprehensive tutorial on network optimization via dual decomposition. A very popular dual distributed optimization algorithm that have improved robustness in the computation and convergence rate in the case of non-strictly convex functions is the so called Alternating Direction Method of Multipliers (ADMM). A first distributed ADMM implementation was initially proposed in [7], and since then several works have appeared as accounted by the survey [8]. Recently, contributions have been dedicated to increase the convergence speed of this technique by means of accelerated consensus schemes [9], [10]. All these algorithms have been proved to converge to the global optimum under the assumption of fixed and undirected topologies.

Another class of distributed optimization algorithms exploits the exchange of active constraints among the network nodes. A constraints consensus algorithm has been proposed in [11] to solve linear, convex and general abstract programs. These were the first distributed optimization algorithms working under asynchronous and direct communication. Recently the constraint exchange idea has been combined with dual decomposition and cutting-plane methods to solve distributed robust convex optimization problems via polyhedral approximations [12]. Although well-suited for asynchronous and directed communications, these algorithms mainly solve constrained optimization problems in which the number of constraints is much smaller than the number of decision variables (or vice-versa).

Other optimization methods include algorithms that try to exploit second-order derivatives, i.e., the Hessians of the cost functions as in [13], [14], where the distributed optimization is applied to general time-varying directed graphs. Another approach, based on Newton-Raphson directions combined with consensus algorithms, has been proposed in [15]: this technique works under synchronous communication, and has recently been extended to asynchronous symmetric gossip frameworks [16].

Importantly, all the works mentioned above require reliable communication; and, to the best of our knowledge, there is no distributed optimization algorithm that has been proved to be guaranteed to converge in the presence of lossy communication. Aiming at filling this gap, we here extend the aforementioned Newton-Raphson consensus approach in [15], [16] to an asynchronous, directed and unreliable network set-up. Specifically, we design a distributed algorithm which works under an asynchronous broadcast protocol over a directed graph and that is robust with respect to packet losses.

The first main contributions of this paper is to endow the Newton-Raphson algorithm in [15] with two additional

This work is supported by the Framework Programme for Research and Innovation Horizon 2020 under grant agreement n. 636834 (DISIRE), and the Swedish research council Norrbottens Forskningsråd.

R. Carli and L. Schenato are with the Department of Information Engineering, University of Padova, Via Gradenigo 6/a, 35131 Padova, Italy { carlirug | schenato }@dei.unipd.it.

Giuseppe Notarstefano is with the Department of Engineering, Università del Salento, Via per Monteroni, 73100 Lecce, Italy giuseppe.notarstefano@unisalento.it.

D. Varagnolo is with the Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Forskargatan 1, 97187 Luleå, Sweden damiano.varagnolo@ltu.se.

strategies: first, a push-sum consensus method, proposed in [17] to achieve average consensus in directed networks; second, a robust consensus method, proposed in [18] to achieve average consensus in presence of packet losses through keeping memory of the total mass of the internal states of the algorithm, so that nodes can recognize if they missed some information at a certain point, and reconstruct it.

The rationale under the combination of the push-sum and robustification protocols with the Newton-Raphson consensus is the following. In the Newton-Raphson consensus, nodes continuously update estimates of a Newton descent direction by means of an average consensus, that forces the nodes to share a common descent direction. Thus, if this averaging property is maintained under asynchronous, directed and lossy communication, the convergence properties of the descent updates can be preserved.

The second main contribution of this paper is to show that, under suitable assumptions on the initial conditions and on the step-size parameter, the Newton-Raphson consensus is locally exponentially stable around the global optimum as soon as the local costs are $\mathcal{C}^2$ and strongly convex with second derivative bounded from below. The exponential convergence is achieved even in the presence of lossy and broadcast communication, as long as the communication graph is strongly connected and the number of consecutive packet losses is bounded. The proof relies on a time-scale separation of the Newton descent dynamics and the average consensus one. This result thus extends the findings of [19], where the convergence was proved for the quadratic local costs case.

The third main contribution of this paper is to complement the theoretical results with numerical simulations based on real datasets extracted from an open-source database. Findings then confirm the local exponential stability and the exponential rate of convergence on a problem where the local cost functions are smooth and convex.

The paper is organized as follows: Section II formulates our problem and working assumptions. Section III then introduces the proposed algorithm and gives some intuitions on the convergence properties of the scheme, which are then summarized in Section V. Finally, Section VI collects some numerical experiments corroborating the theoretical results.

II. Problem Formulation and Assumptions

Problem formulation: we consider the separable optimization problem

$$x^* := \min_x \sum_{i=1}^N f_i(x) \tag{1}$$

under the assumptions that each f_i is known only to node i and is $\mathcal{C}^2$, and strongly convex with second derivative bounded from below, i.e., $f_i''(x) > c$ for all x (so that f_i is coercive). For notational convenience and w.l.o.g. we deal with the scalar case, i.e., $x \in \mathbb{R}$.

We then aim at designing an algorithm solving (1) with the following features:

(i) *being distributed*: each node has limited computational and memory resources and it is allowed to communicate directly only with its in- and out-neighbors;

(ii) *being asynchronous*: nodes do not share a common reference time, but rather perform actions according to local clocks independent of each other;

(iii) *being robust w.r.t. packet losses*: packets broadcast by a node may sometimes be not received by its out-neighbors due to, e.g., collisions or fading effects.

Assumptions: formally, we consider a network representable through a given, fixed, directed and strongly connected graph $\mathcal{G} = (V, \mathcal{E})$ with nodes $V = \{1, \ldots, N\}$ and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ so that $(i, j) \in \mathcal{E}$ iff node j can directly receive information from node i. With $\mathcal{N}_i^{\text{out}}$ we denote the set of *out-neighbors* of node i, i.e., $\mathcal{N}_i^{\text{out}} := \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$ is the set of nodes receiving messages from i. Similarly, with $\mathcal{N}_i^{\text{in}}$ we denote the set of *in-neighbors* of i, i.e., $\mathcal{N}_i^{\text{in}} := \{j \in \mathcal{V} \mid (j, i) \in \mathcal{E}\}$.

As for the concept of time, we assume that each node has its own clock that locally and independently triggers when to transmit. With $\sigma(t) \in \{1, \ldots, N\}$, $t = 1, 2, \ldots$ be the sequence identifying the generic triggered node at time t, i.e., $\sigma(1)$ is the first triggered node, $\sigma(2)$ the second, etc., so that $\sigma(t)$ is a process on the alphabet $\{1, \ldots, N\}$. When a node is triggered, it performs some local computation and then broadcasts some information to its out-neighbors. Due to unreliable communication links, this information can be potentially lost.

We assume that to solve (1) each node i stores in its memory a local copy, say x_i (also called *local estimate* or *local decision variable*), of the global decision variable x. With this new notation (1) reads as

$$\min_{x_1,\ldots,x_N} \sum_{i=1}^N f_i(x_i) \quad \text{s.t. } x_i = x_j \text{ for all } (i,j) \in \mathcal{E}. \tag{2}$$

Notice that the strong connectivity of graph $\mathcal{G}$ ensures then that the optimal solution of (2) is given by $x_1 = \ldots = x_N = x^*$, i.e., ensures that problems (1) and (2) are equivalent.

III. The Robust Asynchronous Newton-Raphson Consensus Algorithm

We now introduce an algorithm suitable for solving problem (1) under the asynchronous and lossy communication assumptions posed in Section II. The procedure, called robust asynchronous Newton-Raphson Consensus (ra-NRC) and reported in Algorithm 1, has been initially presented in [19] but is reported here for completeness and ease of reference. In the pseudo-code we assume w.l.o.g. $\sigma(t) = i$, i.e., that the node that triggers at iteration t is the node i.

We assume that every node i stores in its memory the variables x_i, g_i, h_i, y_i, z_i, $b_{i,y}$, $b_{i,z}$, and $r_{i,y}^{(j)}$, $r_{i,z}^{(j)}$ for every $j \in \mathcal{N}_i^{\text{in}}$, with the following meanings:

- x_i represents the current local estimate at node i of the global minimizer x^*;
- g_i and h_i represent some specific function of the first and second derivatives of the local cost $f_i(x_i)$ computed

at the current value of x_i. g_i^{old} and h_i^{old} represent the old values of g_i and h_i at the previous local step;
- y_i and z_i represent respectively the local estimate at node i of the global sums $\sum_i g_i$ and $\sum_i h_i$;
- $b_{i,y}$ and $b_{i,z}$ represent respectively quantities that are used by node i to locally keep track of the total mass of the internal states y_i and z_i. Notice that $b_{i,y}$ and $b_{i,z}$ are the only local variables that are broadcast by node i to its out-neighbors;
- $r_{j,y}^{(i)}$ and $r_{j,z}^{(i)}$ represent respectively quantities that are used by node j to locally keep track of the total mass of the internal states y_i and z_i of i, that are in general inaccessible by j. In other words, with $r_{j,y}^{(i)}$ and $r_{j,z}^{(i)}$ node j tracks the status of node i: when the communication link from i to j does not fail, then node j updates $r_{j,y}^{(i)}$ and $r_{j,z}^{(i)}$ with the received $b_{i,y}$ and $b_{i,z}$. Otherwise, when the communication link from i to j fails, then $r_{j,y}^{(i)}$ and $r_{j,z}^{(i)}$ remain equal to the previous total mass received.

Thus the ra-NRC algorithm builds on top of broadcast-like average consensus protocols [17] (i.e., the structure of the updates of the variables y_i and z_i) and of strategies for handling packets losses in consensus schemes [18] (i.e., the way of using the variables $b_{i,y}$, $b_{i,z}$, $r_{j,y}^{(i)}$ and $r_{j,z}^{(i)}$ to prevent information losses through mass-tracking robust strategies).

We also notice that the algorithm exploits the thresholding operator

$$[z]_c := \begin{cases} z & \text{if } z \geq c \\ c & \text{otherwise.} \end{cases}$$

where c is a positive scalar to be properly chosen to avoid division-by-zero in the algorithm.

Initialization of the ra-NRC algorithm: we assume that every agents perform the following initialization step of the local variables: let x^o be a common initial estimate of the global minimizer (may be chosen equal to zero for convenience). Then

$$\begin{aligned} x_i &= x^o \\ y_i &= g_i^{\text{old}} = g_i = f_i''(x^o)x^o - f_i'(x^o) =: y_i^o \\ z_i &= h_i^{\text{old}} = h_i = f_i''(x^o) =: z_i^o. \end{aligned}$$

IV. Informal description of the convergence properties of the algorithm

We now provide an intuitive verbal description of the main features and intuitions behind the proposed algorithm, before presenting a mathematical characterization in the following Section V.

We start by noticing that the only free parameter of the algorithm is given by the scalar $\varepsilon \in (0, 1]$. This parameter is fundamental since it regulates the trade-off between the stability of the algorithm and its speed of convergence. Indeed the algorithm is characterizable through two distinct dynamics: a fast one, which distributedly computes averages of the y_i's and z_i's based on a robust consensus algorithm, and a slow dynamics, that estimates the minimizer of the global cost function using the ratio of the averaged y_i's and z_i's as a Newton direction. More specifically, the variables x_i are associated to the slow dynamics, while all the other variables $y_i, z_i, g_i, h_i, b_{i,z}, b_{i,y}, r_{i,y}^{(i)}, r_{i,z}^{(i)}$ are associated to the fast dynamics.

The parameter ε regulates then the separation of these two time scales: the smaller ε is, the larger this separation is, so that small ε's imply slow distributed averaging of the y_i's and z_i's. On the other hand, the rate of convergence of the slow dynamics, i.e., of the Newton-Raphson on the x_i's, can be shown to be locally given by $(1-\varepsilon)$; therefore small ε's imply also slower convergence towards the global optimum.

In the following we use the symbol $\to$ to indicate the behavior of a certain variable as the number of iterations of Algorithm 1 goes to infinity, while we reserve $\leftarrow$ for denoting values assignment operations (e.g., $x_i \leftarrow x^o$ reads as "variable x_i assumes the value x^o").

Algorithm 1 robust asynchronous Newton-Raphson Consensus (ra-NRC)

1: on wake-up, and before transmission, node i updates its local variables as

$$\begin{aligned} y_i &\leftarrow \frac{1}{|\mathcal{N}_i^{\text{out}}|+1}\left[y_i + g_i - g_i^{\text{old}}\right] \\ z_i &\leftarrow \frac{1}{|\mathcal{N}_i^{\text{out}}|+1}\left[z_i + h_i - h_i^{\text{old}}\right] \\ g_i^{\text{old}} &\leftarrow g_i \\ h_i^{\text{old}} &\leftarrow h_i \\ x_i &\leftarrow (1-\varepsilon)x_i + \varepsilon\frac{y_i}{[z_i]_c} \\ g_i &\leftarrow f_i''(x_i)x_i - f_i'(x_i) \\ h_i &\leftarrow f_i''(x_i) \\ b_{i,y} &\leftarrow b_{i,y} + y_i \\ b_{i,z} &\leftarrow b_{i,z} + z_i \end{aligned}$$

2: node i then broadcasts $b_{i,y}$ and $b_{i,z}$ to its neighbors;

3: every out-neighbor $j \in \mathcal{N}_i^{\text{out}}$ updates (if receiving the packet, otherwise it does nothing) its local variables as

$$\begin{aligned} y_j &\leftarrow b_{i,y} - r_{j,y}^{(i)} + y_j + g_j - g_j^{\text{old}} \\ z_j &\leftarrow b_{i,z} - r_{j,z}^{(i)} + z_j + h_j - h_j^{\text{old}} \\ g_j^{\text{old}} &\leftarrow g_j \\ h_j^{\text{old}} &\leftarrow h_j \\ x_j &\leftarrow (1-\varepsilon)x_j + \varepsilon\frac{y_j}{[z_j]_c} \\ g_j &\leftarrow f_i''(x_j)x_i - f_i'(x_j) \\ h_j &\leftarrow f_i''(x_j) \\ r_{j,y}^{(i)} &\leftarrow b_{i,y} \\ r_{j,z}^{(i)} &\leftarrow b_{i,z} \end{aligned}$$

A. Intuitions behind the fast dynamics: the case $\varepsilon = 0$

As ε approaches zeros, x_i changes very little from one iteration to the other, i.e., $x_i \approx \text{cost.}$. Indeed if we assume $\varepsilon = 0$, then the local estimate update rule becomes $x_i \leftarrow x_i$, so that $g_i \leftarrow g_i^{\text{old}}$ and $h_i \leftarrow h_i^{\text{old}}$, i.e., constant values. Therefore in this case the dynamics of y_i only depends on its initial value $f_i''(x_i)x_i - f_i'(x_i)$ and on the communication sequence. Similar considerations hold for z_i's. Thus in this case the variables y_i and z_i evolve as the robust ratio consensus described in [18], i.e.,

$$y_i \to \rho_i \left(\frac{1}{N} \sum_{i=1}^{N} \left(f_i''(x_i)x_i - f_i'(x_i) \right) \right)$$

$$z_i \to \rho_i \left(\frac{1}{N} \sum_{i=1}^{N} f_i''(x_i) \right)$$

where $0 < \rho_i \leq 1$ is some scalar that depends on the packet loss sequence. Thus, regardless of the specific communications and packet losses sequence,

$$\frac{y_i}{z_i} \to \frac{\sum_i f_i''(x_i)x_i - f_i'(x_i)}{\sum_i f_i''(x_i)} =: \phi(x_1, \ldots, x_N)$$

i.e., all the local ratios $\dfrac{y_i}{z_i}$ converge to the same value ϕ.

B. Intuitions behind the slow dynamics: the case $\dfrac{y_i}{z_i} = \phi(x_1, \ldots, x_N)$

The slow dynamics can be obtained by assuming that the fast dynamics has converged to steady-state value considering $\varepsilon = 0$. The idea is that if $\varepsilon \approx 0$, then also $\dfrac{y_i(k)}{z_i(k)} \approx \phi(x_1, \ldots, x_N)$. In this scenario, the dynamics of each local variable x_i can then be written as

$$x_i \leftarrow (1-\varepsilon)x_i + \varepsilon\phi(x_1, \ldots, x_N), \quad i = 1, \ldots, N.$$

This implies that all the various agents update the local values with the same identical rule; thus nodes behave in this case as N identical systems that are driven by the same forcing term. This implies that any difference in the initial value of x_i will vanish, eventually leading to

$$x_i \to x, \quad \forall i = 1, \ldots, N.$$

In this case, moreover,

$$\phi(x_1, \ldots, x_N) \to \frac{\sum_i f_i''(x)x - f_i'(x)}{\sum_i f_i''(x)} = x - \frac{\overline{f}'(x)}{\overline{f}''(x)}$$

where $\overline{f}(x) := \sum_i f_i(x)$. Thus the dynamics of the local variables are of the form

$$x \leftarrow (1-\varepsilon)x + \varepsilon \left(x - \frac{\overline{f}'(x)}{\overline{f}''(x)} \right) = x - \varepsilon \frac{\overline{f}'(x)}{\overline{f}''(x)},$$

i.e., a Newton-Raphson algorithm that, under the posed smoothness assumptions on the local f_i's, converges to the solution of (1). Thus,

$$x_i \to x^* \qquad \forall i = 1, \ldots, N, \quad \forall x^o \in \mathbb{R}.$$

C. Intuitions behind the local rate of convergence $1 - \varepsilon$

The previous analysis allows to estimate the rate of convergence around the global minimum x^*. In fact, if we assume a sufficiently large separation of time scales (i.e., the average consensus on the y_i's and z_i's to be much faster than the Newton-Raphson dynamics), then the rate of convergence of the whole algorithm is dominated by the slow dynamics. If then one further assumes the $f_i(x)$ to be $\mathcal{C}^3$ then the Newton-Raphson dynamics can be linearized so to obtain

$$\frac{d}{dx} \frac{\overline{f}'(x)}{\overline{f}''(x)} \bigg|_{x=x^*} = \frac{\overline{f}''(x)}{\overline{f}''(x)} - \frac{\overline{f}'(x)\overline{f}'''(x)}{(\overline{f}''(x))^2} \bigg|_{x=x^*} = 1$$

where we used the fact that $\overline{f}'(x^*) = 0$. Therefore, the dynamics of the Newton-Raphson component of the algorithm around the equilibrium point x^* can be written as

$$x^+ \approx x - \varepsilon(x - x^*) \Rightarrow (x - x^*)^+ \approx (1-\varepsilon)(x - x^*),$$

which clearly shows that locally the rate of convergence is exponential with a rate given by $(1 - \varepsilon)$. This confirms the previous intuition that smaller ε's lead to slower convergence rates.

D. Intuitions behind the stability properties of the ra-NRC algorithm

As discussed above, ε dictates the relative speed of the fast dynamics (driving the variables y_i and z_i to a consensus), and the slow dynamics for the Newton-Raphson-like evolution of the local estimates x_i. The parameter ε, moreover, dictates how much each node i trusts $\dfrac{y_i}{z_i}$ as a valid Newton direction. During the transient, indeed, this ratio is not the Newton direction of neither the local nor the global cost computed at the current x_i.

Clearly, if the consensus on the y_i's and z_i's is much faster than the evolution of the x_i's (i.e., if ε is "small enough") then one can expect that the aforementioned separation of time scales holds, so that all the quantities converge to their equilibria and the overall algorithm converges. But if ε is not sufficiently small then the stability of the overall system is not guaranteed: indeed, in the following section we prove that there always exists a suitable critical value ε_c such that for all $0 < \varepsilon < \varepsilon_c$ the algorithm is locally exponential stable, while nothing can be said for $\varepsilon > \varepsilon_c$.

Notice that estimating (even offline) such ε_c is a very difficult task, and that explicit bounds are often very conservative. Unfortunately, moreover, the difficulty of finding conservative bounds on ε_c conflicts with the practical necessity of having high ε's (the higher ε, the faster the algorithm converges – if converging – to the optimum).

V. Theoretical analysis of the robust asynchronous Newton-Raphson Consensus

We now provide a theoretical analysis of the proposed algorithm under asynchronous and lossy communication scenarios. In particular we provide some sufficient conditions that guarantee local exponential stability under the assumptions posed in Section II. We thus extend our previous work [19], dedicated to the quadratic local costs case, to more generic local convex costs.

Informally, we assume that each node updates its local variables and communicates with its neighbors infinitely often, and that the number of consecutive packet losses is bounded. Formally, we assume that:

Assumption V.1 (Communications are persistent) *For any iteration $t \in \mathbb{N}$ there exists a positive integer number τ such that each node performs at least one broadcast transmission within the interval $[t, t+\tau]$, i.e., for each $i \in \{1, \ldots, N\}$ there exists $t_i \in [t, t+\tau]$ such that $\sigma(t_i) = i$.*

Assumption V.2 (Packet losses are bounded) *There exists a positive integer L such that the number of consecutive communication failures over every directed edge in the communication graph is smaller than L.*

The following result summarizes our characterization of the convergence properties of the ra-NRC algorithm:

Theorem V.3 *Under Assumptions V.1, V.2 and the assumptions posed in Section II there exist some positive scalars ε_c and δ s.t. if the initial conditions $x^o \in \mathbb{R}$ satisfy $|x^o - x^*| < \delta$ and if ε satisfies $0 < \varepsilon < \varepsilon_c$ then the local variables x_i in Algorithm 1 are exponentially stable w.r.t. the global minimizer x^*.*

Proof: The proof of this theorem is quite involved and relies on many intermediate results. In the interest of space we refer the interested reader to a longer version of this work, [20], including all the technical details in a dedicated Appendix. ■

Introducing the notation $x_i(t)$ to indicate the value x_i after the t-th broadcast event in the whole network, Theorem V.3 reads as follows: if the hypotheses are satisfied then there exist positive scalars C and $\lambda < 1$, possibly function of δ and ε, s.t.

$$|x_i(t) - x^*| \leq C\lambda^t, \qquad t = 1, 2, \ldots$$

Remark V.4 Algorithm 1 assumes the initial conditions of the local variable x_i to be all identical to x^o. Although being not a very stringent requirement, this assumption can be relaxed. I.e., slightly modified versions of Theorem V.3 would hold even in the case $x_i = x_i^o$ as soon as all the initial conditions are sufficiently close to the global minimizer x^*, i.e., as soon as $|x_i^o - x^*| < \delta$ for all $i = 1, \ldots, N$.

Remark V.5 The initial conditions on the local variables $y_i = g_i^{\text{old}} = g_i = f_i''(x^o)x^o - f_i'(x^o)$ and $z_i = h_i^{\text{old}} = h_i = f_i''(x^o)$ are instead more critical for the convergence of the local variables x_i to the true minimizer x^*. As shown in [15], any small perturbation of these initial conditions can affect the equilibrium point of the algorithm, even if it does not affect the stability of the algorithm. In other words, if these perturbations are small then $x_i \to \overline{x}$ with $\overline{x} \approx x^*$. This implies that possible small numerical errors due to the computation and data quantization do not disrupt the convergence properties of the algorithm.

Remark V.6 Although the previous theorem guarantees only local exponential convergence, numerical simulations on real datasets seem to indicate that the basin of attraction is rather large and stability is mostly dictated by the choice of the parameter ε.

VI. Numerical Experiments

First, we empirically study the sensitivity of the convergence speed of the proposed ra-NRC algorithm on ε and on the packet loss probability in Sections VI-A and VI-B, respectively. Then, we compare in Section VI-C the convergence speed of the ra-NRC against the speed of asynchronous subgradient schemes.

We consider the network depicted in Figure 1 and apply our algorithm in the context of robust regression using real-world data. Specifically we consider a database $\mathcal{D}$ containing financial information on various houses. To each house j there is associated an output variable $y_j \in \mathbb{R}$, which indicates its monetary value, and a vector $\chi_j \in \mathbb{R}^n$, which represents n numerical attributes of the j-th house (e.g., per capita crime rate by town, index of accessibility to radial highways, etc.). The database is distributed, i.e., the set $\mathcal{D}$ comes from N different sellers that do not want to disclose their private information. More specifically, each seller i owns a subset $\mathcal{D}_i$ of the global dataset $\mathcal{D}$ so that $\cup_i \mathcal{D}_i = \mathcal{D}$. Nonetheless sellers want to collectively build an estimator of the prices of new houses that is based on all the information possessed by the peers. An approach to solve this distributed regression problem is to solve an optimization problem where the local costs are given by the smooth Huber costs

$$f_i(x) := \sum_{j \in \mathcal{D}_i} \frac{\left(y_j - \chi_j^T x - x_0\right)^2}{\left|y_j - \chi_j^T x - x_0\right| + \beta} + \gamma \left\|x\right\|_2^2 \qquad (3)$$

where γ is a global regularization parameter that is, for our purposes, considered to be known to all agents. We then consider a dataset $\mathcal{D}$ with $|\mathcal{D}| = 500$ elements from the Housing UCI repository[1], randomly assigned to $N = 15$ different users communicating as in graph of Figure 1. For each element we consider $n = 9$ features (the first 9 ones in the database), so that the corresponding optimization problem is 10-dimensional. The centralized optimum x^* for this problem has been computed using a centralized Newton-Raphson (NR) scheme with Newton step chosen with back-tracking, and terminating when the Newton decrement was $< 10^{-9}$.

A. Empirical analysis of the effects of ε on the convergence speed of the ra-NRC algorithm

We consider a probability of packet losses fixed to 0.1, and a ε that ranges in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, and compare in

[1] `http://archive.ics.uci.edu/ml/datasets/Housing`

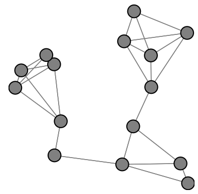

Fig. 1. A random geometric graph with connectivity radius 0.35.

Figure 2 the evolution of the average errors for different values of ε. We notice how the results agree with the intuitions developed in the previous sections, and that, importantly, $\varepsilon = 10^{-1}$ leads to non converging behaviors.

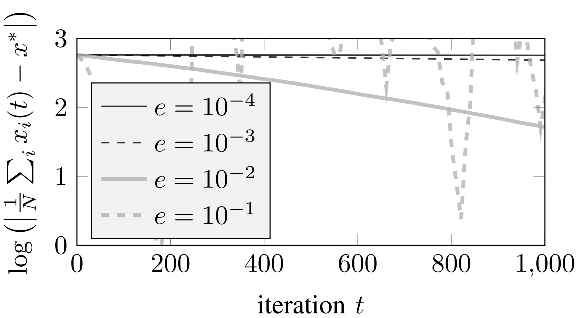

Fig. 2. Comparison of the evolutions of the trajectories of the average errors for different values of ε and a packets loss probability $p = 0.1$.

B. Empirical analysis of the effects of packet losses on the convergence speed of the ra-NRC algorithm

We consider a parameter ε fixed to 0.01, and a probability of packet losses that ranges in $\{0, 0.2, 0.4, 0.6\}$, so to compare in Figure 3 the evolution of the average errors for different packets unreliability levels. We notice that, as expected, the severity of the packet losses negatively affects the convergence speed. Nonetheless the overall slowing effect is not disruptive, in the sense that even severe packet loss probabilities (namely, 0.6) do not lead to meaningless estimates.

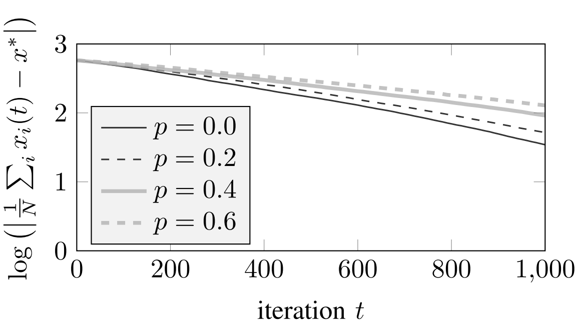

Fig. 3. Comparison of the evolutions of the trajectories of the average errors for the ra-NRC algorithm for different values of the packet loss probabilities and $\varepsilon = 0.01$.

C. Convergence speeds comparisons

We consider the asynchronous subgradient scheme reported in Algorithm 2, and numerically compare its convergence properties against the proposed ra-NRC scheme under a packet losses probability equal to 0.1.

Algorithm 2 Distributed Subgradient

1: on initialization, each node i initializes x_i as x_i^o and t_i (the local counter of the number of updates) to 1;

2: on wake up, node i broadcasts x_i and $f_i(x_i)$ to all its neighbors;

3: every out-neighbor $j \in \mathcal{N}_i^{\text{out}}$ updates (if receiving the packet, otherwise it does nothing) its local variables as

$$x_j \leftarrow \frac{1}{2}(x_i + x_j) + \frac{\alpha}{t_j}\big(f_i(x_i) + f_j(x_j)\big)$$
$$t_j \leftarrow t_j + 1$$

For both algorithms we compute, through gridding, that parameter (ε for the ra-NRC, α for the subgradient) that leads to the best performance in terms of convergence speed of the average guess over the various agents. We then report the evolution of the average guess over time in Figure 4, and notice how the higher order information used by the ra-NRC scheme over the subgradient one positively affects the asymptotic convergence speed of the procedure.

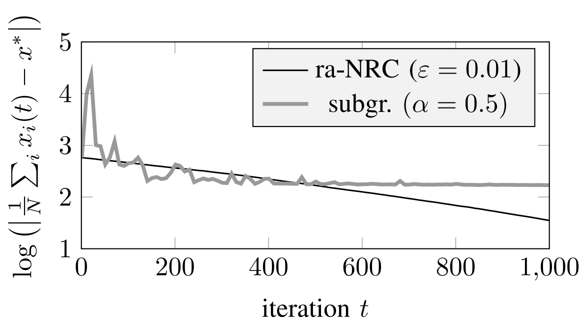

Fig. 4. Comparison of the evolutions of the trajectories of the average errors for the algorithms tuned with their best parameters and a packet loss probability $p = 0.1$.

VII. Conclusions

Implementations of distributed optimization methods in real-world scenarios require strategies that are both able to cope with real-world problematics (like unreliable, asynchronous and directed communications), and converge sufficiently fast so to produce usable results in meaningful times. Here we worked towards this direction, and improved an already existing distributed optimization strategy, previously shown to have fast convergence properties, so to make it tolerate the previously mentioned real-world problematics.

More specifically, we considered a robustified version of the Newton-Raphson consensus algorithm originally proposed in [15] and proved its convergence properties under some general mild assumptions on the local costs. From technical perspectives we shown that under suitable assumptions on the initial conditions, on the step-size parameter, on the connectivity of the communication graph and on the boundedness of the number of consecutive packet losses, the considered optimization strategy is locally exponentially stable around the global optimum as soon as the local costs are $\mathcal{C}^2$ and strongly convex with second derivative bounded from below.

We also shown how the strategy can be applied to real world scenarios and datasets, and be used to successfully compute optima in a distributed way.

We then notice that the results offered in this manuscript do not deplete the set of open questions and plausible extensions of the Newton Raphson consensus strategy. We indeed devise that the algorithm is potentially usable as a building block for distributed interior point methods, but that some lacking features prevent this development. Indeed it is still not clear how to tune the parameter ε online so that the convergence speed is dynamically adjusted (and maximized), how to account for equality constraints of the form $Ax = b$, and how to update the local variables x_i using partition-based approaches so that each agent keeps and updates only a subset of the components of x.

REFERENCES

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

[2] P. Lin and W. Ren, "Distributed subgradient projection algorithm for multi-agent optimization with nonidentical constraints and switching topologies," in *IEEE Conference on Decision and Control*. IEEE, 2012, pp. 6813–6818.

[3] A. Nedic and A. Olshevsky, "Distributed optimization over time-varying directed graphs," in *IEEE Conference on Decision and Control*, Dec 2013, pp. 6855–6860.

[4] B. Gharesifard and J. Cortes, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 781–786, 2014.

[5] S. S. Kia, J. Cortes, and S. Martinez, "Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication," in *arXiv*, 2014.

[6] B. Yang and M. Johansson, "Distributed optimization and games: A tutorial overview," *Networked Control Systems*, pp. 109–148, 2011.

[7] 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 Transactions on Signal Processing*, vol. 56, no. 1, pp. 350 – 364, 2008.

[8] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[9] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the Alternating Direction Method of Multipliers (ADMM): quadratic problems," *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, 2014.

[10] A. Teixeira, E. Ghadimi, I. Shames, H. Sandberg, and M. Johansson, "Optimal scaling of the ADMM algorithm for distributed quadratic programming," in *IEEE Conference on Decision and Control*. IEEE, 2013, pp. 6868–6873.

[11] G. Notarstefano and F. Bullo, "Distributed abstract optimization via constraints consensus: Theory and applications," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2247–2261, October 2011.

[12] M. Bürger, G. Notarstefano, and F. Allgöwer, "A polyhedral approximation framework for convex and robust distributed optimization," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 384–395, Feb 2014.

[13] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A Distributed Newton Method for Network Utility Maximization - I: Algorithm," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2162–2175, 2013.

[14] ——, "A Distributed Newton Method for Network Utility Maximization - Part II: Convergence," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2176 – 2188, 2013.

[15] F. Zanella, D. Varagnolo, A. Cenedese, P. Gianluigi, and L. Scenato, "Newton-Raphson Consensus for Distributed Convex Optimization," in *Proc. 50th IEEE Conf. on Decision and Control*, Orlando, Florida, December 2011, pp. 5917 – 5922.

[16] F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato, "Asynchronous Newton-Raphson Consensus for Distributed Convex Optimization," in *3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 2012.

[17] F. Bénézit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli, "Weighted gossip: Distributed averaging using non-doubly stochastic matrices," in *IEEE International Symposium on Information Theory Proceedings (ISIT)*. IEEE, 2010, pp. 1753–1757.

[18] M. A. D. Dominguez-Garcis, 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," *arXiv:1109.6391v1 [cs.SY] 29 Sep 2011*, 2011.

[19] R. Carli, G. Notarstefano, L. Schenato, and D. Varagnolo, "Distributed quadratic programming under Asynchronous and Lossy Communications via Newton-Raphson Consensus," in *European Control Conference*, 2015.

[20] ——, "Analysis of Newton-Raphson Consensus for multi-agent convex optimization under asynchronous and lossy communications," in *IEEE Conference on Decision and Control*, 2015, [Online] Available at http://automatica.dei.unipd.it/people/schenato/publications.html.