The convergence rate of Newton-Raphson consensus optimization for quadratic cost functions

Filippo Zanella, Damiano Varagnolo, Angelo Cenedese, Gianluigi Pillonetto and Luca Schenato

***Abstract*— We consider the convergence rates of two convex optimization strategies in the context of multi agent systems, namely the Newton-Raphson consensus optimization and a distributed Gradient-Descent opportunely derived from the first. To allow analytical derivations, the convergence analyses are performed under the simplificative assumption of quadratic local cost functions. In this framework we derive sufficient conditions which guarantee the convergence of the algorithms. From these conditions we then obtain closed form expressions that can be used to tune the parameters for maximizing the rate of convergence. Despite these formulae have been derived under quadratic local cost functions assumptions, they can be used as rules-of-thumb for tuning the parameters of the algorithms in general situations.**

***Index Terms*— distributed optimization, convex optimization, consensus algorithms, multi-agent systems, Newton-Raphson methods, rate of convergence**

I. Introduction

Optimization, intended as the search for the best choice in a set of plausible alternatives, is probably the most widely pervasive concept in the whole set of all the computational sciences. This happens because many practical problems are usually translatable into well-posed optimization problems, e.g., as for learning [1] and decision theory [2].

In this paper we aim to contribute to this trend analyzing the stability and the rate of convergence of two recently proposed distributed optimization strategies [3], [4] for smooth convex cost functions. More precisely, we consider study these algorithms under the assumption of quadratic cost functions. Although this restriction is substantial, nonetheless it allows for analytic characterization and stability conditions and optimization of the rate of convergence, which are in general not possible for general convex functions. The intent is thus to derive general rule-of-thumbs based on these analytical results that could be useful for the design and tuning of the algorithm in the context of general convex cost functions.

Here we specifically consider nonlinear unconstrained optimization techniques, and send the reader that is interested in distributed Linear Programming techniques back to [5].

All the authors are with the Department of Information Engineering, University of Padova, via Gradenigo 6/b, 35131 Padova, Italy. Emails: { bb } @dei.unipd.it.

The research leading to these results has received funding from the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement n°257462 HYCON2 Network of excellence and n°223866 FeedNetBack, by Progetto di Ateneo CPDA090135/09 funded by the University of Padova, and by the Italian PRIN Project "New Methods and Algorithms for Identification and Adaptive Control of Technological Systems".

The contributions on nonlinear distributed optimization can then possibly be divided in three main categories: primal decompositions based methods, dual decompositions based methods, and heuristic methods.

The primal decomposition methods class contains all the algorithms that follow from manipulation of the primal problem. An important class is the one of the so-called subgradient methods, see, e.g., [6] and references therein. They are characterized by a wide applicability, easy implementability and mild requirements on the objective functions. Unfortunately they may be rather slow: see, e.g., [7, Chap. 6] for results on real distributed systems. This approach has been deeply explored including different features such as randomization and asynchronous implementations [8], [9], [10], [11].

The dual decomposition methods class contains all the algorithms obtained from the manipulation of the dual problem. Usually the dual problem is split into simpler sub-tasks, and this requires the addition of local variables that need to be exchanged among agents. The idea is to constrain these additional variables to agree upon a common value, thus forcing the algorithm to converge to the global optimum. An important algorithm in this class of methods is the so-called Alternating Direction Method of Multipliers (ADMM), developed in [12, pp. 253-261] and extended in various distributed contexts like, e.g., [13], [14].

Other approaches may be tailored for particular optimization problems and be thus extremely fast at the cost of restricting the class of possible cost functions, like the so-called Fast-Lipschitz methods [15], [16].

Since the performance properties of the strategies presented above are difficult to be characterized analytically, in this paper we focus only on a novel primal-based strategy called distributed Newton-Raphson optimization consensus, named because of its reminiscence with the classical Newton-Raphson optimization method. The choice of analyzing this strategy is motivated by the fact that it is as easily implementable and as flexible as subgradient-based techniques (i.e., it requires neither synchronous communications schemes nor fixed communication graphs, thus it can be used as it is, e.g., in swarm robotics frameworks), and it has convergence speeds comparable to the ones of ADMM schemes [3], [17].

In particular, we analytically characterize the rate of convergence of two distributed convex optimization techniques in a opportune simplificative framework. In particular, we consider connected networks where: a) the local cost functions are generic quadratic costs; b) communications

are synchronous; c) the communication matrix P is an irreducible symmetric stochastic matrix, i.e., s.t. $P\mathbb{1} = \mathbb{1}$, $P = P^T$, $P_{ij} \geq 0$ where $\mathbb{1} := [1, \ldots, 1]$ and P_{ij} is the generic element of P.

The distributed optimization techniques that are characterized in the following are the Newton-Raphson consensus optimization approach (Algorithm 1, derived from [3]) and a novel gradient-descent implementation that is obtained from a simplification of the previous Newton-Raphson approach (Algorithm 2, derived from [4]).

II. Problem formulation and notation

We consider an undirected, connected and static network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ composed by N agents, each endowed with a local scalar quadratic cost function

$$\psi_i : \mathbb{R} \mapsto \mathbb{R} \qquad \psi_i(x) = \frac{1}{2} a_i (x - b_i)^2$$

with $a_i > 0$ (this implies ψ_i to be strictly convex). The global cost function

$$\overline{\psi} : \mathbb{R} \mapsto \mathbb{R} \qquad \overline{\psi}(x) := \frac{1}{N} \sum_{i=1}^{N} \psi_i(x)$$

is thus still a quadratic cost. The goal of the agents is to collaborate in order to compute the minimizer x^* of the global cost function $\overline{\psi}$, which is given by:

$$x^* := \arg\min_x \overline{\psi}(x) = \frac{\sum_{i=1}^N a_i b_i}{\sum_{i=1}^N a_i} = \frac{\frac{1}{N}\sum_{i=1}^N a_i b_i}{\frac{1}{N}\sum_{i=1}^N a_i} . \quad (1)$$

The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents the communication network as follows: the agents are represented by the vertexes $\mathcal{V} = \{1, \ldots, N\}$ while the available communication links correspond to the edges $(i, j) \in \mathcal{E}$. The time-invariant communication matrix $P \in \mathbb{R}^{N \times N}$ is assumed to be a *symmetric consensus matrix*, i.e., to have non-negative elements, to be s.t. $P\mathbb{1} = \mathbb{1}$, $P = P^T$, and compatible with the communication edges $\mathcal{E}$ (i.e., $P_{ij} > 0$ only if $(i, j) \in \mathcal{E}$). Under these hypotheses, the Perron-Frobenius theorem guarantees that $\lim_{k \to +\infty} P^k = \frac{1}{N}\mathbb{1}\mathbb{1}^T$. We assume that the spectrum of P, $\mathrm{eig}(P) = \{\lambda_1 = 1, \lambda_2, \ldots, \lambda_N\}$, is known, and that the eigenvalues are sorted in decreasing order. To compact the notation, we let $\Lambda := \mathrm{diag}[\lambda_1, \ldots, \lambda_N]$. We also consider the *essential spectral radius* defined as

$$\sigma := \max_{\lambda_i,\, i=2,\ldots,N} |\lambda_i|$$

which under the connectivity hypothesis of the communication graph is s.t. $\sigma < 1$. We also assume that no communication or quantization errors occur (i.e., information can be exchanged with infinite precision).

In the following sections we will use the following shorthands:

$$\begin{aligned}
g_i(x_i(k)) &:= \psi_i''(x_i(k))\, x_i(k) - \psi_i'(x_i(k)) \\
\widetilde{g}_i(x_i(k)) &:= x_i(k) - \psi_i'(x_i(k)) \\
h_i(x_i(k)) &:= \psi_i''(x_i(k)) \\
\boldsymbol{x}(k) &:= [x_1(k) \ \cdots \ x_N(k)]^T \\
\boldsymbol{g}(\boldsymbol{x}(k)) &:= [g_1(x_1(k)) \ \cdots \ g_N(x_N(k))]^T \\
\widetilde{\boldsymbol{g}}(\boldsymbol{x}(k)) &:= [\widetilde{g}_1(x_1(k)) \ \cdots \ \widetilde{g}_N(x_N(k))]^T \\
\boldsymbol{h}(\boldsymbol{x}(k)) &:= [h_1(x_1(k)) \ \cdots \ h_N(x_N(k))]^T \\
\boldsymbol{a} &:= [a_1 \ \cdots \ a_N]^T \\
\boldsymbol{b} &:= [b_1 \ \cdots \ b_N]^T
\end{aligned}$$

where $\psi' := \frac{d\psi}{dx}$ and $\psi'' := \frac{d^2\psi}{dx^2}$. Plain upper case letters generally indicate matrices (sometimes scalar parameters), bold lower case letters indicate vectors, and plain lower case letters indicate scalars. The notation $\mathrm{diag}[\boldsymbol{v}]$, where $\boldsymbol{v} = [v_1 \ \cdots \ v_N]$ is a generic vector, denotes a diagonal matrix with $v_1, \ldots, v_N$ on its diagonal. $I := \mathrm{diag}[\mathbb{1}]$. We use the usual symbol $\odot$ to indicate the component-wise Hadamard product, and the fraction bar to indicate also the Hadamard division, i.e., the component-wise division of vectors:

$$\frac{\boldsymbol{g}(\boldsymbol{x}(k))}{\boldsymbol{h}(\boldsymbol{x}(k))} := \left[\frac{g_1(x_1(k))}{h_1(x_1(k))}, \ \ldots, \ \frac{g_N(x_N(k))}{h_N(x_N(k))}\right]^T .$$

III. Distributed Newton-Raphson

In this section we analyze the synchronous version of the Newton-Raphson consensus algorithm proposed in [17], reported in Alg. 1.

Algorithm 1 Newton-Raphson Consensus [3], [17]

(storage allocation and constraints on parameters)

$\boldsymbol{x}(k), \boldsymbol{y}(k,m), \boldsymbol{z}(k,m) \in \mathbb{R}^N$ for $m = 0, \ldots, M$ and $k = 0, 1, \ldots$

$P \in \mathbb{R}^{N \times N}$, positive and doubly stochastic

$\varepsilon \in (0, 1)$

(initialization)

set: $\boldsymbol{g}\big(\boldsymbol{x}(-1)\big) = \boldsymbol{h}\big(\boldsymbol{x}(-1)\big) = \boldsymbol{0}$

$\boldsymbol{y}(0, M) = \boldsymbol{z}(0, M) = \boldsymbol{0}$

$\boldsymbol{x}(0) = \boldsymbol{0}$

(main algorithm)

for $k = 1, 2, \ldots$ **do**

  $\boldsymbol{y}(k, 0) = \boldsymbol{y}(k-1, M) + \boldsymbol{g}\big(\boldsymbol{x}(k-1)\big) - \boldsymbol{g}\big(\boldsymbol{x}(k-2)\big)$

  $\boldsymbol{z}(k, 0) = \boldsymbol{z}(k-1, M) + \boldsymbol{h}\big(\boldsymbol{x}(k-1)\big) - \boldsymbol{h}\big(\boldsymbol{x}(k-2)\big)$

  for $m = 1, \ldots, M$ **do**

    $\boldsymbol{y}(k, m) = P\boldsymbol{y}(k, m-1)$

    $\boldsymbol{z}(k, m) = P\boldsymbol{z}(k, m-1)$

  $\boldsymbol{x}(k) = (1 - \varepsilon)\boldsymbol{x}(k-1) + \varepsilon \dfrac{\boldsymbol{y}(k, M)}{\boldsymbol{z}(k, M)}$

Assuming $M = 1$ and introducing the variables $\boldsymbol{v}(k)$ and $\boldsymbol{w}(k)$ to account respectively for $\boldsymbol{g}\big(\boldsymbol{x}(k-1)\big)$ and

$\boldsymbol{h}\big(\boldsymbol{x}(k-1)\big)$, Alg. 1 can be rewritten as

$$\begin{cases} \boldsymbol{v}(k) & = \boldsymbol{a} \odot \boldsymbol{b} \\ \boldsymbol{w}(k) & = \boldsymbol{a} \\ \boldsymbol{y}(k) & = P\big(\boldsymbol{y}(k-1) + \boldsymbol{a} \odot \boldsymbol{b} - \boldsymbol{v}(k-1)\big) \\ \boldsymbol{z}(k) & = P\big(\boldsymbol{z}(k-1) + \boldsymbol{a} - \boldsymbol{w}(k-1)\big) \\ \boldsymbol{x}(k) & = (1-\varepsilon)\boldsymbol{x}(k-1) + \varepsilon \dfrac{\boldsymbol{y}(k)}{\boldsymbol{z}(k)} \end{cases}$$

with initial conditions $\boldsymbol{v}(0) = \boldsymbol{w}(0) = \boldsymbol{y}(0) = \boldsymbol{z}(0) = \boldsymbol{x}(0) = \boldsymbol{0}$. From

$$\begin{cases} \boldsymbol{v}(k) & = \boldsymbol{a} \odot \boldsymbol{b} \\ \boldsymbol{w}(k) & = \boldsymbol{a} \\ \boldsymbol{y}(k) & = P^k\,(\boldsymbol{a} \odot \boldsymbol{b}) \\ \boldsymbol{z}(k) & = P^k \boldsymbol{a} \end{cases}$$

and

$$\boldsymbol{p}(k) := \frac{P^{k+1}\,(\boldsymbol{a} \odot \boldsymbol{b})}{P^{k+1}\boldsymbol{a}}\;; \quad \boldsymbol{p}^* := \frac{\frac{1}{N}\sum_{i=1}^N a_i b_i}{\frac{1}{N}\sum_{i=1}^N a_i}\mathbb{1}$$

we obtain the simplified system

$$\boldsymbol{x}(k+1) = (1-\varepsilon)\boldsymbol{x}(k) + \varepsilon \boldsymbol{p}(k)\;, \qquad \boldsymbol{x}(0) = \boldsymbol{0}\;.$$

We notice that proposition 2 in [3] assures the existence of an $\bar{\varepsilon} \in \mathbb{R}_+$ s.t. if $\varepsilon < \bar{\varepsilon}$ then Algorithm 1 distributedly and asymptotically computes the global optimum x^*, i.e., $\lim_{k\to+\infty} \boldsymbol{x}(k) = x^*\mathbb{1}$.

In order to compute the rate of convergence, we want to express the systems dynamics in terms of the local error of each agent with respect to the global minimum, i.e., $|x^* - x_i(k)|$, therefore we consider the error vector $\boldsymbol{\xi}(k) := \boldsymbol{x}(k) - \boldsymbol{p}^*$ whose dynamics can be written as

$$\boldsymbol{\xi}(k+1) = (1-\varepsilon)\boldsymbol{\xi}(k) + \varepsilon\big(\boldsymbol{p}^* - \boldsymbol{p}(k)\big),$$

or equivalently as

$$\boldsymbol{\xi}(k) = \sum_{\ell=1}^{k} \varepsilon(1-\varepsilon)^{k-\ell}\big(\boldsymbol{p}^* - \boldsymbol{p}(\ell-1)\big)\;.$$

Consider now that, from consensus theory, it holds that $\boldsymbol{y}(k) \xrightarrow{\sigma^k} \frac{1}{N}\sum_{i=1}^N a_i b_i \mathbb{1}$ and $\boldsymbol{z}(k) \xrightarrow{\sigma^k} \frac{1}{N}\sum_{i=1}^N a_i \mathbb{1}$. Recall then that $\boldsymbol{p}(k) = \dfrac{\boldsymbol{y}(k)}{\boldsymbol{z}(k)}$, and that the Hadamard division operator is continuous and differentiable around the point $\boldsymbol{p}^*$. This implies then that there exist a positive constant $c \in \mathbb{R}_+$ which might depend on the initial condition $\boldsymbol{\xi}(0)$ s.t.

$$\|\boldsymbol{p}^* - \boldsymbol{p}(\ell)\|_2 \le c\sigma^\ell, \quad \forall \ell$$

where σ is the essential spectral radius of P. Thus, we have that

$$\begin{aligned} \|\boldsymbol{\xi}(k)\|_2 &\le c\varepsilon\frac{(1-\varepsilon)^k}{\sigma}\sum_{\ell=1}^{k}\left(\frac{\lambda_2}{1-\varepsilon}\right)^\ell \\ &= c\varepsilon\frac{(1-\varepsilon)^k - \sigma^k}{1-\varepsilon-\sigma}\;. \end{aligned}$$

Considering then that

$$\|\boldsymbol{\xi}(k)\|_2 \le \left|\frac{c\varepsilon}{1-\varepsilon-\sigma}\right|(1-\varepsilon)^k + \left|\frac{c\varepsilon}{1-\varepsilon-\sigma}\right|\sigma^k$$

it follows that the convergence rate is dominated by the biggest between $(1-\varepsilon)$ and σ. The previous thus states that it is possible, by setting $\varepsilon = 1$, to directly obtain $\|\boldsymbol{\xi}(k)\|_2 \le c\sigma^k$, i.e., for the quadratic case the unique factor limiting the convergence rate of the algorithm is given by the speed of the consensus algorithm induced by P.

We can summarize the results obtained above in the following:

Theorem 1 Under the assumption of local quadratic cost functions, Algorithm 1 is ensured to converge for all $\varepsilon \in (0,2)$ for any positive vector $\boldsymbol{a}$. Moreover the fastest rate of convergence of the algorithm is given by the essential spectral radius of P, namely σ, and it is achieved for any $|\varepsilon| \le 1-\sigma$.

IV. Distributed Gradient Descent

In this section we consider a modified version of Alg. 1, with the advantage of requiring a smaller number of local variables and therefore a reduced communication load. As we will see, this trades off with a more restricted interval of ε's guaranteeing the convergence to the global optimum, and thus eventually with a slower convergence rate. The algorithm, initially proposed in [4] and here reported in Alg. 2, is reminiscent of a distributed gradient descent strategy based on a consensus algorithm.

Algorithm 2 Distributed Gradient Descent [4]

(storage allocation and constraints on parameters)
1: $\boldsymbol{x}(k), \boldsymbol{y}(k,m) \in \mathbb{R}^N$ for $m = 0,\dots,M$ and $k = 0,1,\dots$
2: $P \in \mathbb{R}^{N\times N}$, positive and doubly stochastic
3: $\varepsilon \in (0,1)$

(initialization)
set: $\widetilde{\boldsymbol{g}}\,(\boldsymbol{x}(-1)) = \boldsymbol{0}$
4: $\boldsymbol{y}(0,M) = \boldsymbol{0}$
$\boldsymbol{x}(0) = \boldsymbol{0}$

(main algorithm)
5: **for** $k = 1,2,\dots$ **do**
6: $\boldsymbol{y}(k,0) = \boldsymbol{y}(k-1,M) + \widetilde{\boldsymbol{g}}\big(\boldsymbol{x}(k-1)\big) - \widetilde{\boldsymbol{g}}\big(\boldsymbol{x}(k-2)\big)$
7: **for** $m = 1,\dots,M$ **do**
8: $\boldsymbol{y}(k,m) = P\boldsymbol{y}(k,m-1)$
9: $\boldsymbol{x}(k) = (1-\varepsilon)\boldsymbol{x}(k-1) + \varepsilon\boldsymbol{y}(k,M)$

The analytical derivations of the stability and convergence rate of Algorithm 2 are more involved that those of the previous algorithm and rely on two main steps: 1) the transformation of the algorithm into a Linear Time Invariant (LTI) system, characterized by an additional parameter; 2) the adoption of small-gain theory to derive analytical rules to compute the optimal ε and the convergence rate based on σ and the vector $\boldsymbol{a}$.

A. Transformation of Algorithm 2 into an LTI system

We start by setting $M = 1$ and defining the new variable $\boldsymbol{v}(k) = \widetilde{\boldsymbol{g}}\big(\boldsymbol{x}(k-1)\big)$, so that Alg. 2 can be rewritten as

$$\begin{cases} \boldsymbol{v}(k) = \mathrm{diag}\,[\mathbb{1} - \boldsymbol{a}]\,\boldsymbol{x}(k-1) + \mathrm{diag}\,[\boldsymbol{a}]\,\boldsymbol{b} & (2) \\ \boldsymbol{y}(k) = P\big(\boldsymbol{y}(k-1) + \boldsymbol{v}(k) - \boldsymbol{v}(k-1)\big) & (3) \\ \boldsymbol{x}(k) = (1-\varepsilon)\boldsymbol{x}(k-1) + \varepsilon \boldsymbol{y}(k) & (4) \end{cases}$$

with initial conditions $\boldsymbol{v}(0) = \boldsymbol{y}(0) = \boldsymbol{x}(0) = \mathbf{0}$. Substituting (2) into (3) we obtain

$$\boldsymbol{y}(k) = P\boldsymbol{y}(k-1) + P\mathrm{diag}\,[\mathbb{1} - \boldsymbol{a}]\,\big(\boldsymbol{x}(k-1) - \boldsymbol{x}(k-2)\big)$$

that, substituted into (4), gives

$$\begin{aligned} \boldsymbol{x}(k) = \Big((1-\varepsilon)I + \varepsilon P\mathrm{diag}\,[\mathbb{1} - \boldsymbol{a}]\Big)\boldsymbol{x}(k-1) + \varepsilon P\boldsymbol{y}(k-1) + \\ -\varepsilon P\mathrm{diag}\,[\mathbb{1} - \boldsymbol{a}]\,\boldsymbol{x}(k-2) \end{aligned} \tag{5}$$

Rearranging the update rule (4) we obtain[1]:

$$\boldsymbol{y}(k-1) = \frac{1}{\varepsilon}\boldsymbol{x}(k-1) - \frac{1-\varepsilon}{\varepsilon}\boldsymbol{x}(k-2)\,. \tag{6}$$

Then by substituting (6) into (5) we eventually rewrite (4) as

$$\begin{aligned} \boldsymbol{x}(k) \;=\; & \Big((1+\varepsilon)P + (1-\varepsilon)I - \varepsilon P\mathrm{diag}\,[\boldsymbol{a}]\Big)\boldsymbol{x}(k-1) \\ & + \Big(-P + \varepsilon P\mathrm{diag}\,[\boldsymbol{a}]\Big)\boldsymbol{x}(k-2) \end{aligned} \tag{7}$$

with initial conditions $\boldsymbol{x}(-1) = \boldsymbol{x}(0) = \mathbf{0}$.

Let us new define the following diagonal matrix $\Delta := \mathrm{diag}\,[1-a_1, 1-a_2, \ldots, 1-a_N]$, summarizing the deviations from the ideal condition where alla parabolic cost functions are identical and with unitary curvature. Clearly

$$\delta := \max_i |1 - a_i| \quad \Rightarrow \quad \|\Delta\|_2 = \delta.$$

Let us now define the new state vector $\boldsymbol{\chi}(k) := \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{x}(k-1) \end{bmatrix}$ and the following matrices:

$$A := \begin{bmatrix} (1+\varepsilon)P + (1-\varepsilon)I - \varepsilon P & \varepsilon P - P \\ I & \mathbf{0} \end{bmatrix} \tag{8}$$

$$B := \begin{bmatrix} \varepsilon P \\ \mathbf{0} \end{bmatrix} \qquad C := \begin{bmatrix} -I & I \end{bmatrix}. \tag{9}$$

With these we can transform (7) into

$$\boldsymbol{\chi}(k+1) = (A + B\Delta C)\boldsymbol{\chi}(k), \tag{10}$$

i.e., into the LTI feedback system

$$\begin{cases} \boldsymbol{\chi}(k+1) &= A\boldsymbol{\chi}(k) + B\boldsymbol{u}(k) \\ \boldsymbol{\nu}(k) &= C\boldsymbol{\chi}(k) \\ \boldsymbol{u}(k) &= \Delta\boldsymbol{\nu}(k) \end{cases} \tag{11}$$

Therefore $\boldsymbol{x}(k)$ converges to $x^*\mathbb{1}$ if and only if $\boldsymbol{\chi}(k)$ converges to $x^*[\mathbb{1}^T\ \mathbb{1}^T]^T$.

[1]We notice that there is a causal connection between $\boldsymbol{y}(k)$ and $\boldsymbol{x}(k)$ since (4) can be computed only after the computation of (3). Nonetheless we can exploit (6) being it a relation between quantities that, at time k, are all known.

Let us consider the unitary matrix U that diagonalizes the communication matrix P, i.e., $U^T P U = \Lambda$, and let us introduce $\overline{\boldsymbol{\chi}} := \begin{bmatrix} U & 0 \\ 0 & U \end{bmatrix}\boldsymbol{\chi}$. With these we can obtain the equivalent LTI system

$$\overline{\boldsymbol{\chi}}(k+1) = (\overline{A} + \overline{B}\,\overline{\Delta}\,\overline{C})\overline{\boldsymbol{\chi}}(k) \tag{12}$$

where

$$\overline{A} = \begin{bmatrix} \Lambda + (1-\varepsilon)I & (\varepsilon - 1)\Lambda \\ I & \mathbf{0} \end{bmatrix} \qquad \overline{B} = \begin{bmatrix} \varepsilon\Lambda \\ \mathbf{0} \end{bmatrix} \tag{13}$$

$$\overline{C} = C \qquad\qquad \overline{\Delta} = U^T\Delta U\,. \tag{14}$$

The previous system can be rewritten in block-diagonal form by adopting an opportune change of variables $\widetilde{\boldsymbol{\chi}} := V\overline{\boldsymbol{\chi}}$ where V is a simple permutation matrix. More precisely, let $\overline{\boldsymbol{\chi}}$ be $\overline{\boldsymbol{\chi}} = [\overline{\chi}_1'\,\overline{\chi}_2'\ldots\overline{\chi}_N'\,\overline{\chi}_1''\,\overline{\chi}_2''\ldots\overline{\chi}_N'']^T$. Then V can be chosen s.t. $\widetilde{\boldsymbol{\chi}} = [\overline{\chi}_1'\,\overline{\chi}_1''\overline{\chi}_2'\,\overline{\chi}_2''\ldots\overline{\chi}_N'\,\overline{\chi}_N'']^T$. In this way we obtain

$$\widetilde{\boldsymbol{\chi}}(k+1) = (\widetilde{A} + \widetilde{B}\,\widetilde{\Delta}\,\widetilde{C})\widetilde{\boldsymbol{\chi}}(k) \tag{15}$$

where

$$\widetilde{A} = \begin{bmatrix} A_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & A_N \end{bmatrix} \qquad \widetilde{B} = \begin{bmatrix} B_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_N \end{bmatrix} \tag{16}$$

$$\widetilde{C} = \begin{bmatrix} C_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_N \end{bmatrix} \qquad \widetilde{\Delta} = V^T U^T \Delta U V \tag{17}$$

and where $C_i := [-1, 1]$,

$$A_i := \begin{bmatrix} \lambda_i + (1-\varepsilon) & (\varepsilon - 1)\lambda_i \\ 1 & 0 \end{bmatrix} \qquad B_i := \begin{bmatrix} \varepsilon\lambda_i \\ 0 \end{bmatrix}. \tag{18}$$

From the previous equation it can be seen that the global dynamics can be decomposed into N parallel subsystem of dimension 2, which are coupled by the uncertainty matrix $\widetilde{\Delta}$. Notice that the dynamics of the global system is thus affine in the uncertainty matrix $\widetilde{\Delta}$, the latter thus amenable for the stability and convergence properties of the whole algorithm.

To analyze these stability properties we now exploit classical small-gain theory results [18, Chap. 5], that guarantee the stability of the global system if the following N perturbed subsystems (with an abuse of notation on $\boldsymbol{x}$, $\boldsymbol{u}$ and $\boldsymbol{y}$)

$$\begin{cases} \boldsymbol{x}(k+1) &= A_i\boldsymbol{x}(k) + B_i\boldsymbol{u}(k) \\ y(k) &= C_i\boldsymbol{x}(k) \\ \boldsymbol{u}(k) &= \Delta_i y(k), \end{cases} \tag{19}$$

with Δ_i s.t. $\|\Delta_i\|_2 \le \|\widetilde{\Delta}\|_2 = \|\Delta\|_2 = \delta$, are stable. We remark that this kind of results generally provide conservative bounds, since they do not take into account the structure of $\widetilde{\Delta}$, but only the knowledge of its 2-norm.

Small-gain theory can also be used to analyze the rate of convergence by recasting the computation of the rate of convergence as a stability problem. In fact, by considering the transformation $\overline{\boldsymbol{x}}(k) = \rho^k\boldsymbol{x}(k)$, it is immediate to check

that the convergence rate of (19) is at least as ρ^{-k}, $\rho > 1$ if and only if all the systems

$$\begin{cases} \overline{\boldsymbol{x}}(k+1) &= \rho A_i \overline{\boldsymbol{x}}(k) + \rho B_i \boldsymbol{u}(k) \\ y(k) &= C_i \overline{\boldsymbol{x}}(k) \\ \boldsymbol{u}(k) &= \Delta_i y(k) \end{cases} \tag{20}$$

are asymptotically stable.

To this regard, the transfer functions of the input-output systems (20) are given by

$$f_i(z) = \begin{cases} \dfrac{-\varepsilon\rho}{z - \rho(1-\varepsilon)} & \text{if } i = 1 \\ \dfrac{-\varepsilon\lambda_i\rho(z-\rho)}{(z-\rho\lambda_i)(z-\rho(1-\varepsilon))} & \text{if } i = 2, \dots, N \end{cases} \tag{21}$$

i.e., ρ modulates the position of the natural poles λ_i and $(1-\varepsilon)$. Note that the transfer function relative to the average component, i.e., the transfer function relative to the subsystem $i = 1$, has order one due a zero-pole cancelation relative to the eigenvalue $z = \rho$. This is to be expected since it corresponds to the unitary eigenvalue (multiplied by ρ) of the global dynamics (10) which guarantees that the consensus $\boldsymbol{\chi} = \mathbb{1}$ is an equilibrium point of the global system. Such eigenvalue and its relative eigenspace $\boldsymbol{\chi} = \mathbb{1}$ should be excluded from the stability analysis. This is indeed the case since this eigenvector is independent of Δ, therefore it does not appear in the transfer functions above.

B. Stability analysis

Based on small-gain theory, see, e.g., [18, Chap. 5], the various systems (20) are ensured to be asymptotically stable if the products of the $\mathcal{L}^2$ gains of the direct chains and the feedbacks are strictly smaller than 1, i.e., if

$$\max_{\omega \in [0, 2\pi)} \|f_i(\exp(j\omega))\|_2 \, \|\Delta_i\|_2 < 1, \qquad i = 1, \dots, N. \tag{22}$$

To check whether (22) holds we exploit the following:

Lemma 2

$$\|f_1\|_2 = \mathcal{F}(\rho, \varepsilon) = \begin{cases} |f_1(0)| = \dfrac{\varepsilon\rho}{1-\rho(1-\varepsilon)} \\ \qquad \text{if } \rho(1-\varepsilon) \geq 0 \\ |f_1(\pi)| = \dfrac{\varepsilon\rho}{1+\rho(1-\varepsilon)} \\ \qquad \text{if } \rho(1-\varepsilon) < 0 \end{cases} \tag{23}$$

$$\|f_i(\exp(j\omega)\|_2 \leq \mathcal{G}(\rho, \varepsilon), \quad i = 2, \dots, N \tag{24}$$

where $\mathcal{G}(\rho, \varepsilon)$ is defined as

$$\begin{cases} \dfrac{\varepsilon\sigma\rho}{(1-\rho\sigma)} \dfrac{(\rho+1)}{(1+\rho(1-\varepsilon))} & \text{if } 1 \leq \rho \leq \dfrac{1}{\sqrt{1-\varepsilon}} \\ \dfrac{\varepsilon\sigma\rho}{(1-\rho\sigma)} \dfrac{(\rho-1)}{(1-\rho(1-\varepsilon))} & \text{if } \dfrac{1}{\sqrt{1-\varepsilon}} \leq \rho \leq \dfrac{1}{1-\varepsilon}. \end{cases}$$

With lemma 2 and condition (22) we then bound the stability region of the proposed algorithm (in terms of ε) and upper and lower bound its rate of convergence. We start with the smallest ε guaranteeing stability, that can be defined via the following optimization problem:

$$\begin{array}{rll} \varepsilon_c(\sigma, \delta) := & \sup_\varepsilon & \varepsilon \\ & \text{s.t.} & \mathcal{F}(1, \varepsilon)\delta < 1 \\ & & \mathcal{G}(1, \varepsilon)\delta < 1. \end{array} \tag{25}$$

The smallest ε guaranteeing stability is then described by the following:

Theorem 3 Let

$$\varepsilon_1 := \frac{2}{1+\delta}, \quad \varepsilon_2 := \frac{2(1-\sigma)}{1-\sigma+2\delta\sigma}, \quad \varepsilon_c := \min\{\varepsilon_1, \varepsilon_2\}. \tag{26}$$

If $\varepsilon \in (0, \varepsilon_c)$ then Alg. 2 converges to the global optimum.

It is also easy to verify that if $\sigma > \frac{1}{2}$ then $\varepsilon_2 < \varepsilon_1$ and thus also $\varepsilon_c = \varepsilon_2$. In general, large networks are s.t. $\sigma \approx 1$, thus in large networks the typical limiting factor is ε_2 (recall that ε_1 is associated to the dynamics of the average component). To complete the caracterization in large networks we also notice that, in the same situation, if $\delta\sigma \gg 1 - \sigma$ then $\varepsilon_c \approx \dfrac{1-\sigma}{\delta}$. I.e., in this case the critical ε_c is directly proportional to the spectral gap and inversely proportional to the deviation from the ideal condition where all the costs are unitarily curved parabolas.

Convergence rate analysis

Letting $\eta := 1/\rho$, we can bound the rate of convergence by means of the optimization problem

$$\begin{array}{rll} (\varepsilon^*, \eta^*) := & \arg\inf_{\varepsilon,\eta} & \eta \\ & \text{s.t.} & \mathcal{F}(1/\eta, \varepsilon)\delta < 1 \\ & & \mathcal{G}(1/\eta, \varepsilon)\delta < 1, \end{array} \tag{27}$$

To solve (27) we divide it in two subproblems that are then analyzed separately. In particular, defining

$$\begin{array}{rll} \eta_1(\varepsilon) := & \inf_\eta & \eta \\ & \text{s.t.} & \mathcal{F}(1/\eta, \varepsilon)\delta < 1 \end{array} \tag{28}$$

$$\begin{array}{rll} \eta_2(\varepsilon) := & \inf_\eta & \eta \\ & \text{s.t.} & \mathcal{G}(1/\eta, \varepsilon)\delta < 1 \end{array} \tag{29}$$

$$\eta(\varepsilon) := \max\{\eta_1(\varepsilon), \eta_2(\varepsilon)\} \tag{30}$$

It follows that the solution of (27) can be rewritten as

$$\varepsilon^* = \arg\inf_\varepsilon \eta(\varepsilon), \quad \eta^* = \eta(\varepsilon^*).$$

The solutions of (28) and (29) are then given by:

Theorem 4 Let

$$\kappa_1(\varepsilon) := \frac{\sigma+1-\varepsilon(1+\sigma\delta)+\sqrt{\big(\sigma+1-\varepsilon(1+\sigma\delta)\big)^2-4\big(\sigma-\varepsilon\sigma(1+\delta)\big)}}{2}$$

$$\kappa_2(\varepsilon) := \frac{\sigma-1+\varepsilon(1+\sigma\delta)-\sqrt{\big(\sigma-1+\varepsilon(1+\sigma\delta)\big)^2+4\big(\sigma-\varepsilon\sigma(1-\delta)\big)}}{2}$$

$$\overline{\varepsilon} := \frac{-\sigma^2+2\sigma\delta+2-\sigma\sqrt{\sigma^2+4\sigma\delta+4\sigma^2\delta^2}}{2\big(\sigma\delta+1\big)^2}$$

$$\eta_2(\overline{\varepsilon}) := \frac{\sigma+\sqrt{\sigma^2+4\sigma\delta(1+\sigma\delta)}}{2(1+\sigma\delta)}.$$

Then

$$\eta_1(\varepsilon) = \begin{cases} 1-\varepsilon(1-\delta) & \text{if } 0 < \varepsilon \leq 1 \\ -1+\varepsilon(1+\delta) & \text{if } 1 < \varepsilon \leq \varepsilon_1 \end{cases} \tag{31}$$

$$\eta_2(\varepsilon) = \begin{cases} \kappa_1(\varepsilon) & \text{if } 0 < \varepsilon \leq \overline{\varepsilon} \\ \kappa_2(\varepsilon) & \text{if } \overline{\varepsilon} < \varepsilon \leq \varepsilon_2. \end{cases} \tag{32}$$

A graphical representation of the functions and variables defined above is given in Figure 1 for a specific choice of the parameters δ and σ.

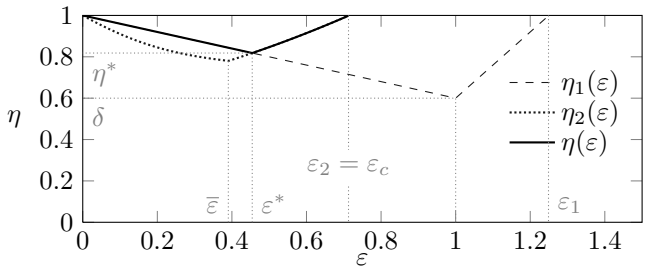

Fig. 1. Numerical evaluation of function η_1, η_2, η as given in Theorem 4 and problem (30) for $\sigma = \delta = 0.6$.

The optimal rate of converge η^* as a function of the parameters σ and δ cannot be given in closed form, however can be readily computed based on the previous theorem as shown in Figure 2. This figure clearly shows that rate of convergence decreases as either δ or σ are close to one.

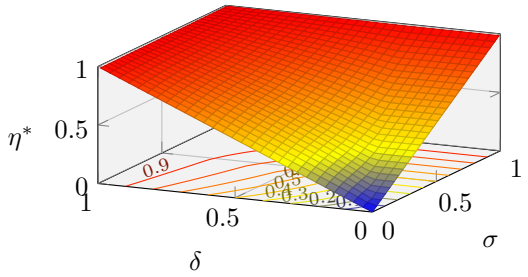

Fig. 2. Optimal rate of convergence η^* as a function of the parameters σ and δ.

As a completion of the remark proposed after Thm. 3, Lemma 4 implies that, in large networks and for $\delta \neq 0$,

$$\overline{\varepsilon} = \frac{2(1-\sigma)}{1+2\delta} + o(1-\sigma), \quad \eta_2(\overline{\varepsilon}) = 1 - \frac{1-\sigma}{1+2\delta} + o(1-\sigma),$$

i.e., as expected, the convergence rate η is lower with small spectral gaps $1-\sigma$ and higher with large deviations of the curvature of the local costs given by a_i.

V. Conclusions and Future Works

The rates of convergence of the two consensus-based optimization algorithms, namely the Newton-Raphson consensus and the Gradient Descent consensus, have been studied under some simplifying assumptions, i.e. quadratic cost functions and synchronous communications, with the aim of building the path for characterizations valid in general frameworks.

The results have shown that convergence properties heavily rely on the amount of coordination required to the agents. Especially for the distributed gradient descent, the degree of diversity of the local cost functions, i.e. their curvature, impacts on the rate of convergence: the intuition is that the optimum can be reached more easily if agents have similar curvatures. In a certain sense, similar costs reflect to similar behaviors, and similar behaviors require less coordination to reach consensus.

This intuition introduces one of the main research directions to be addressed in the future, since an absence of synchronization can be intended as milder coordination requirements. Natural questions are thus how the rate of convergence of the algorithms is affected by time-varying consensus protocols and and non-quadratic costs fucntions.

References

[1] B. Schölkopf and A. J. Smola, *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. The MIT Press, 2002.

[2] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis*, 2nd ed. Springer - Verlag, 1985.

[3] F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-Raphson consensus for distributed convex optimization," in *IEEE Conference on Decision and Control and European Control Conference*, Dec. 2011, pp. 5917–5922.

[4] ——, "Multidimensional Newton-Raphson consensus for distributed convex optimization," in *American Control Conference*, 2012.

[5] 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.

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

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

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

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

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

[11] A. Nedić, 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, Apr. 2010.

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

[13] 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, pp. 350–364, Jan. 2008.

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

[15] C. Fischione, "F-Lipschitz Optimization with Wireless Sensor Networks Applications," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2319 – 2331, 2011.

[16] C. Fischione and U. Jönsson, "Fast-Lipschitz Optimization with Wireless Sensor Networks Applications," in *International symposium on Information processing in sensor networks*, 2011.

[17] F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato, "Asynchronous Newton-Raphson Consensus for Distributed Convex Optimization," in *Necsys 2012*, 2012.

[18] C. A. Desoer and M. Vidyasagar, *Feedback Systems*. SIAM, 2009.