2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Newton-Raphson consensus for distributed convex optimization

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

***Abstract*— We study the problem of unconstrained distributed optimization in the context of multi-agents systems subject to limited communication connectivity. In particular we focus on the minimization of a sum of convex cost functions, where each component of the global function is available only to a specific agent and can thus be seen as a private local cost. The agents need to cooperate to compute the minimizer of the sum of all costs. We propose a consensus-like strategy to estimate a Newton-Raphson descending update for the local estimates of the global minimizer at each agent. In particular, the algorithm is based on the separation of time-scales principle and it is proved to converge to the global minimizer if a specific parameter that tunes the rate of convergence is chosen sufficiently small. We also provide numerical simulations and compare them with alternative distributed optimization strategies like the Alternating Direction Method of Multipliers and the Distributed Subgradient Method.**

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

I. Introduction

The ability of distributedly and autonomously solve large scale optimization problems is becoming nowadays of paramount importance to build effectively smart networks of agents performing auxiliary and automatic operations such as in wireless sensor networks [1] and in the next-generation electrical-power smart grids [2].

In distributed scenarios it is mandatory to provide the agents with the ability to jointly and autonomously solve optimization problems without relying on a central processing units while requiring minimal coordination effort and possibly small computational and memory requirements.

We focus on the problem of distributed unconstrained minimization of a sum of convex functions, where each component of the global function is available only to a specific agent and can thus be seen as a private local cost. Practical examples of this scenario arise in home automation contexts, where smart electrical devices need to agree on the total amount of energy consumption that maximizes an overall utility-function that is given by the sum of the utility functions of the devices. Other examples include distributed computation of M-estimators in robust statistics [3] and distributed statistical learning [4].

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".

A. Previous work

Since the seminal work [5], both centralized and distributed optimization have been a major research topic for the decades in the area of control and system theory [6], [7], [8], [9]. Distributed optimization algorithms can be roughly divided into three main categories: methods based on primal decompositions, methods based on dual decomposition, and heuristic methods.

In primal decomposition methods, there is a direct manipulation of the primal variables, often through subgradient methods as shown in the survey [10] and in the references therein. Even if they were originally proposed to boost the convergence speed of centralized optimization schemes, they are widely applicable, easy to implement and require mild assumptions on the objective functions where convexity is the most crucial one. Despite these nice properties, they may be rather slow and may not progress at each iteration, as shown in [11, Chap. 6] in the context of real wireless sensor networks. There are several possible implementations mostly based on incremental gradients methods [12] which can be deterministic [13] or randomized [14], [15]. Important extensions include the use of projections in order to take into account possible different local constraints [16], and the analysis of the convergence rate and error bounds [17], [18]. These algorithms have also been compared with more traditional linear consensus algorithms [19].

In dual decomposition methods, not only the primal variables are manipulated, but also the dual ones are. Usually the original problem is split into several sub-problems whose solution is simpler although some form of coordination is required since the order of the execution of the sub-problems is critical. Despite originally intended for CPU-saving purposes, they have been successfully employed for distributed optimization purposes. In dual based methods every agent owns a local copy of the variables that are locally updated by the same agent. Convergence to the global optimum is then ensured constraining the convergence of the local variables to a common value [20], [21]. In the class of dual decomposition methods, a particularly popular strategy is the alternating direction method of multipliers developed in [8, pp. 253-261] which has been recently proposed in a distributed context [22], [4].

Other approaches, e.g. the so-called Fast-Lipschitz strategies [23], [24], exploit particular structures of the objective functions and constraints to increase the convergence speed at the cost of being suitable only for particular optimization problems. Finally, alternative

distributed optimization approaches can be based on heuristics, like swarm optimization [25], or genetic algorithms [26], however their convergence and performance properties are difficult to be studied analytically.

B. Contribution

In this work we propose a distributed algorithm for the exact computation of optimal solution that approximatively operates as a Newton-Raphson minimization procedure. The algorithm is based on inter-agents communication schemes that are used in classical *average-consensus* algorithms [27], [28]. The main idea is to compute a Newton-Raphson approximation for the minimizer of the global cost function via an average consensus algorithm and to move towards such minimum sufficiently slowly to allow the consensus algorithm to converge. The use of consensus algorithms has also been adopted in other distributed strategies such as the Distributed Subgradient Methods (DSMs) [29]. These methods typically have a convergence rate of $\frac{1}{k}$ where k is the number of performed steps, but are not always proven to converge to the global optimum under non-smoothness hypotheses (see e.g. [30, Prop. 3]). Differently, our algorithm is proved to converge to the optimal solution for appropriate choices of the algorithm parameters based on the principle of separation of time-scales. Another very important feature of the algorithm is that it inherits the properties of consensus algorithms like their simplicity, their potential implementation with asynchronous communication schemes, and their ability to adapt to time-varying network topologies. This differentiates our algorithm from the strategies based on the Alternating Direction Method of Multipliers (ADMMs) whose asynchronous implementation is more involved and cannot easily handle time-varying topologies [22], [4]. Finally, despite an existing literature on second-order based distributed optimization techniques, see e.g. [31], [32], at the best of our knowledge the proposed strategy is the unique method (among the primal decomposition methods) s.t. the local estimates evolve as driven by a Newton-Raphson procedure.

In the following we present our algorithm under a number of simplificative assumptions. In particular we consider smooth convex scalar cost functions and synchronous communication schemes. Finally we complement the analytical results with some numerical simulations and comparison with DSM and ADMM optimization schemes on a ring communication graph. In the interest of space all the proofs of the offered propositions are included in the companion technical report [33].

II. Problem formulation

We assume that in a network of N agents, each agent is endowed with a local strictly convex cost function $f_i : \mathbb{R} \mapsto \mathbb{R}$. We define the global cost function as

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

and we assume that the goal of each agent is to distributedly minimize $\overline{f}$, i.e. compute

$$x^* := \arg\min_x \overline{f}(x) \tag{2}$$

through low-complexity distributed optimization schemes. The communication network is modeled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ whose vertexes $\mathcal{V} = \{1, 2, \ldots, N\}$ represent the agents and the edges $(i, j) \in \mathcal{E}$ represent the available communication links. We assume that the graph is undirected and connected. We say that a stochastic matrix $P \in \mathbb{R}^{N \times N}$, i.e. a matrix whose elements are non-negative and $P\mathbb{1} = \mathbb{1}$ where $\mathbb{1} = [1\, 1 \cdots 1]^T \in \mathbb{R}^N$, is consistent with a graph $\mathcal{G}$ if $P_{ij} > 0$ only if $(i, j) \in \mathcal{E}$. If P is also symmetric and includes all edges, i.e. $P_{ij} > 0$ if $(i, j) \in \mathcal{E}$, then $\lim_{k\to\infty} P^k = \frac{1}{N}\mathbb{1}\mathbb{1}^T$. Such matrix P is also often referred as *consensus matrix*.

For the purposes of the paper, we define the shorthands

$$g_i(x_i(k)) := f_i''(x_i(k)) \cdot x_i(k) - f_i'(x_i(k)) \tag{3}$$

$$h_i(x_i(k)) := f_i''(x_i(k)) \tag{4}$$

$$\mathbf{x}(k) := \begin{bmatrix} x_1(k) \\ \vdots \\ x_N(k) \end{bmatrix} \tag{5}$$

$$\mathbf{g}(\mathbf{x}(k)) := \begin{bmatrix} g_1(x_1(k)) \\ \vdots \\ g_N(x_N(k)) \end{bmatrix} \tag{6}$$

$$\mathbf{h}(\mathbf{x}(k)) := \begin{bmatrix} h_1(x_1(k)) \\ \vdots \\ h_N(x_N(k)) \end{bmatrix} \tag{7}$$

where we used the shorthand notation $f' := \frac{df}{dx}$ and $f'' := \frac{d^2 f}{dx^2}$, bold fonts to indicate vectors, and plain italic fonts to indicate scalars. In general we will use the fraction bar to indicate the Hadamard division, i.e. the component-wise division of vectors:

$$\frac{\mathbf{g}(\mathbf{x}(k))}{\mathbf{h}(\mathbf{x}(k))} := \left[\frac{g_1(x_1(k))}{h_1(x_1(k))}, \ldots, \frac{g_N(x_N(k))}{h_N(x_N(k))}\right]^T. \tag{8}$$

We will also use bold fonts to indicate vectorial quantities or functions which range is vectorial, plain italic fonts to indicate scalar quantities or functions which range is a scalar.

To simplify the proofs, in the following we will exploit the succeeding assumptions:

Assumption 1. The functions $f_i \in \mathcal{C}^2, \forall i$, i.e. they are continuous up to the second derivatives, their second derivatives f_i'' are strictly positive, bounded, and they are defined for all $x \in \mathbb{R}$. Moreover the global minimizer x^* does not take on the extended values $\pm\infty$.

We notice that from the strictly convexity assumptions it follows that x^* is unique. Moreover the assumption $x^* \neq \pm\infty$ is to obtain convergence proofs that do not require modifications of the standard multi-time-scales approaches for singular perturbation model analysis [34], [35, Chap. 11].

III. Distributed Newton-Raphson Consensus

It is well known that if the cost functions are quadratic, then it is possible to distributively compute the optimal solution x^* using the output of two average consensus algorithms [36], [37]. In fact, consider $f_i(x) = \frac{1}{2}a_i(x-b_i)^2$ where $a_i > 0$. Then straightforward computations show that the minimizer is given by

$$x^* = \frac{\sum_{i=1}^N a_i b_i}{\sum_{i=1}^N a_i} = \frac{\frac{1}{N}\sum_{i=1}^N a_i b_i}{\frac{1}{N}\sum_{i=1}^N a_i}$$

i.e. it is the ratio of two averages. Therefore if each agent defines the following local variables $y_i(0) := a_i b_i$ and $z_i(0) := a_i$, and then updates them based on two average consensus algorithms:

$$\begin{array}{rcl} \mathbf{y}(k+1) & = & P\mathbf{y}(k) \\ \mathbf{z}(k+1) & = & P\mathbf{z}(k) \end{array} \qquad \mathbf{x}(k) = \frac{\mathbf{y}(k)}{\mathbf{z}(k)}$$

then $\lim_{k\to\infty} \mathbf{x}(k) = x^*\mathbb{1}$ where P is a consensus matrix. This means that each $x_i(k) = y_i(k)/z_i(k)$ corresponds to the local estimate at time k that each agent has about the global minimizer x^*. If the cost functions are not quadratic, then the previous strategy cannot be applied as it is but needs to be modified. First of all, it is important to notice that

$$a_i b_i = f_i''(x)x - f_i'(x) = g_i(x), \quad a_i = f_i''(x) = h_i(x)$$

for all x. However we cannot simply run the previous strategy with initial conditions $y_i(0) = g_i(x_i(0))$ and $z_i(0) = h_i(x_i(0))$, since $x_i(k)$ changes over time and therefore one should change accordingly the initial conditions of the consensus algorithms. Therefore, it is necessary to modify the consensus algorithm to track the changing signals $g_i(x_i(k))$ and $h_i(x_i(k))$. Secondly, setting $x_i(k) = y_i(k)/z_i(k)$ might be too aggressive since the estimate is not correct at the beginning, therefore only a small step should be taken towards the estimated global minimum $y_i(k)/z_i(k)$. These observations have been used to propose the following Algorithm 1, where initialization given in line 4 is critical for convergence to the global minimizer, lines 6-7 are local operations that make sure that the Newton-Raphson computation is done based on the current estimate $x_i(k)$ location, lines 9-10 perform the consensus operations, and line 11 is again a local operation which performs a convex combination between the past estimate $x_i(k-1)$ and the new estimate $y_i(k)/z_i(k)$.

To explain why Algorithm 1 distributedly computes the global optimum x^*, we rewrite it as

$$\begin{cases} \mathbf{v}(k) = \mathbf{g}(\mathbf{x}(k-1)) \\ \mathbf{w}(k) = \mathbf{h}(\mathbf{x}(k-1)) \\ \mathbf{y}(k) = P^M\big(\mathbf{y}(k-1) + \mathbf{g}(\mathbf{x}(k-1)) - \mathbf{v}(k-1)\big) \\ \mathbf{z}(k) = P^M\big(\mathbf{z}(k-1) + \mathbf{h}(\mathbf{x}(k-1)) - \mathbf{w}(k-1)\big) \\ \mathbf{x}(k) = (1-\varepsilon)\mathbf{x}(k-1) + \varepsilon\dfrac{\mathbf{y}(k-1)}{\mathbf{z}(k-1)}\,. \end{cases} \tag{9}$$

Algorithm 1

(storage allocation and constraints on parameters)

1: $\mathbf{x}(k), \mathbf{y}(k,m), \mathbf{z}(k,m) \in \mathbb{R}^N, m = 0,\dots,M; k = 0,1,\dots$
2: $P \in \mathbb{R}^{N\times N}$, positive and doubly stochastic
3: $\varepsilon \in (0,1)$

(initialization)

4: set: $\mathbf{x}(0) = \mathbf{0}$, $\mathbf{g}(\mathbf{x}(-1)) = \mathbf{h}(\mathbf{x}(-1)) = \mathbf{0}$, $\mathbf{y}(0,M) = \mathbf{z}(0,M) = \mathbf{0}$

(main algorithm)

5: **for** $k = 1,2,\dots$ **do**
6: $\quad \mathbf{y}(k,0) = \mathbf{y}(k-1,M) + \mathbf{g}(\mathbf{x}(k-1)) - \mathbf{g}(\mathbf{x}(k-2))$
7: $\quad \mathbf{z}(k,0) = \mathbf{z}(k-1,M) + \mathbf{h}(\mathbf{x}(k-1)) - \mathbf{h}(\mathbf{x}(k-2))$
8: $\quad$ **for** $m = 1,\dots,M$ **do**
9: $\quad\quad \mathbf{y}(k,m) = P\mathbf{y}(k,m-1)$
10: $\quad\quad \mathbf{z}(k,m) = P\mathbf{z}(k,m-1)$
11: $\quad \mathbf{x}(k) = (1-\varepsilon)\mathbf{x}(k-1) + \varepsilon\dfrac{\mathbf{y}(k,M)}{\mathbf{z}(k,M)}$

Consider then the continuous-time system

$$\begin{cases} \varepsilon\dot{\mathbf{v}}(t) = -\mathbf{v}(t) + \mathbf{g}(\mathbf{x}(t)) \\ \varepsilon\dot{\mathbf{w}}(t) = -\mathbf{w}(t) + \mathbf{h}(\mathbf{x}(t)) \\ \varepsilon\dot{\mathbf{y}}(t) = -K\mathbf{y}(t) + (I-K)\left[\mathbf{g}(\mathbf{x}(t)) - \mathbf{v}(t)\right] \\ \varepsilon\dot{\mathbf{z}}(t) = -K\mathbf{z}(t) + (I-K)\left[\mathbf{h}(\mathbf{x}(t)) - \mathbf{w}(t)\right] \\ \dot{\mathbf{x}}(t) = -\mathbf{x}(t) + \dfrac{\mathbf{y}(t)}{\mathbf{z}(t)} \end{cases} \tag{10}$$

with $K := I - P^M$. By construction the matrix K is positive semidefinite and its kernel is generated by the vector $\mathbb{1}$, and its eigenvalues are $0 = \lambda_1 < \mathrm{Re}[\lambda_2] \le \cdots \le \mathrm{Re}[\lambda_N] < 2$. It is immediate to check that system (9) is a discretized version of (10), i.e. (9) can be derived from (10) through an Euler discretization with time interval $T = \varepsilon$, thus qualitatively behaving in the same manner for sufficiently small ε. In this form, it is immediate to recognize the existence of a two-time scales dynamical system regulated by the parameter ε. Therefore, we can split the dynamics in the two time scales and study them separately for sufficiently small ε. The fast dynamics, i.e. the first four equations of system (10), imply that $\mathbf{y}(t) \approx \big(\frac{1}{N}\mathbb{1}^T\mathbf{g}(\mathbf{x}(t))\big)\mathbb{1}$ and $\mathbf{z}(t) \approx \big(\frac{1}{N}\mathbb{1}^T\mathbf{h}(\mathbf{x}(t))\big)\mathbb{1}$. If these equations are inserted into the slow dynamics, i.e. the last equation of system (10), then it follows that $\mathbf{x}(t) \approx \overline{x}(t)\mathbb{1}$ where the quantity $\overline{x}(t)$ evolves with good approximation following the ordinary differential equation

$$\dot{\overline{x}}(t) = -\frac{\overline{f}'(\overline{x}(t))}{\overline{f}''(\overline{x}(t))} \tag{11}$$

corresponding to a continuous Newton-Raphson algorithm that, under our simplificative Assumption 1, converges to the global optimum x^*[1]. These observations are formally stated in the following:

Proposition 2. Consider Algorithm 1, which is equivalent to system (9) with initial conditions $\mathbf{v}(0) = \mathbf{w}(0) = \mathbf{y}(0) =$

[1]Asymptotic properties of the continuous time Newton-Raphson method can be found e.g. in [38], [39].

$\mathbf{z}(0) = \mathbf{0}$. If Assumption 1 holds true, then there exists 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. $\lim_{k\to+\infty} \mathbf{x}(k) = x^*\mathbb{1}$.

The claim of the previous proposition is valid only for the specific initial conditions defined in line 4 of Algorithm 1. Although these are initial conditions of an algorithm and therefore can be arbitrarily designed, nonetheless it is important to evaluate the robustness of the algorithm for different initial conditions, since this gives suggestions about its robustness to numerical errors and communication noise. It turns out that the initial conditions on the initial estimates $x_i(0)$ can be arbitrary, however initial conditions $v_i(0), w_i(0), y_i(0), z_i(0)$ can change the final convergence point and might even lead to instability for sufficiently large values. This is formally stated in the next proposition:

Proposition 3. Consider system (10) with arbitrary initial conditions $\mathbf{v}(0), \mathbf{w}(0), \mathbf{y}(0), \mathbf{z}(0), \mathbf{x}(0)$, and define the following scalars:

$$\begin{aligned} \alpha(0) &:= \frac{1}{N}\mathbb{1}^T(\mathbf{y}(0) - \mathbf{v}(0)) \\ \beta(0) &:= \frac{1}{N}\mathbb{1}^T(\mathbf{z}(0) - \mathbf{w}(0)) \,. \end{aligned} \tag{12}$$

If Assumption 1 holds true, then there exists an $\bar{\varepsilon}, \bar{\alpha}, \bar{\beta} \in \mathbb{R}_+$ such that if $\varepsilon < \bar{\varepsilon}$, $|\alpha(0)| < \bar{\alpha}$, $|\beta(0)| < \bar{\beta}$ then $\lim_{t\to+\infty} \mathbf{x}(t) = \xi(\alpha(0), \beta(0))\mathbb{1}$ where $\xi(\alpha(0), \beta(0))$ is a scalar continuous function of its arguments and has the property that $\xi(0, 0) = x^*$.

IV. Numerical Examples

We consider a ring network where agents can communicate only to their left and right neighbors through the symmetric circulant consensus matrix

$$P = \begin{bmatrix} 0.5 & 0.25 & & & 0.25 \\ 0.25 & 0.5 & 0.25 & & \\ & \ddots & \ddots & \ddots & \\ & & 0.25 & 0.5 & 0.25 \\ 0.25 & & & 0.25 & 0.5 \end{bmatrix} . \tag{13}$$

We also assume that agents perform only one consensus step per cycle of the algorithm, i.e. that $M = 1$. The local objective functions are randomly generated as

$$f_i(x) = c_i e^{a_i x} + d_i e^{-b_i x}, \quad i = 1, \ldots, N \tag{14}$$

where $a_i, b_i \sim \mathcal{U}[0, 0.2]$ and $c_i, d_i \sim \mathcal{U}[0, 1]$.

Fig. 1 compares the evolution of the local states x_i of the continuous system (10) 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 $x_i(0) = 0$ (top panel). As ε decreases, the difference between the two time scales becomes more evident and all trajectories of $x_i(t)$ become closer to the trajectory given by the slow Newton-Raphson dynamics $\overline{x}(t)$ defined in (11), which is guaranteed to converge to the global minimizer x^* (middle and bottom panels).

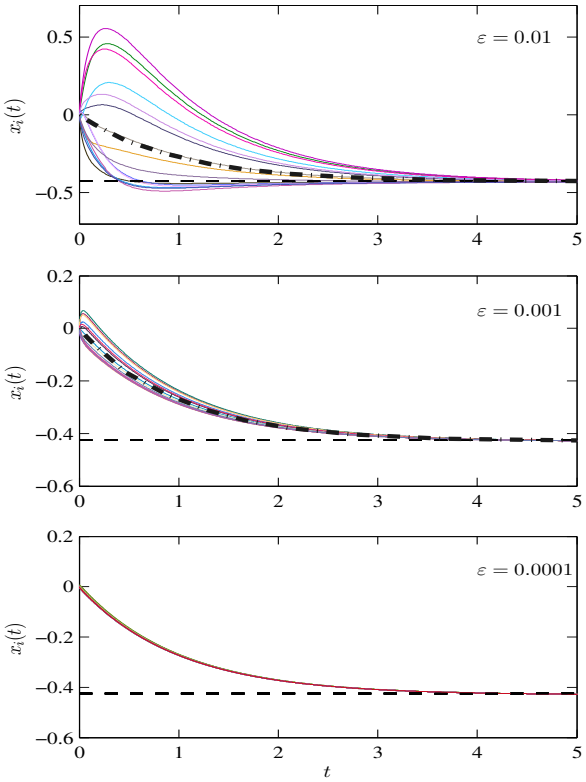

Fig. 1. Temporal evolution of system (10) for different values of ε ($N = 15$). The black dashed line indicates x^*. The dashed-dotted line indicates the slow dynamics $\overline{x}(t)$. As ε decreases. the difference between the time scale of the slow and fast dynamics increases, and the local states $x_i(t)$ converge to the manifold of $\overline{x}(t)$.

Fig. 2 shows that, as stated in Proposition 3, if $\alpha(0) = \beta(0) = 0$, then local states $x_i(t)$ converge to the optimal solution x^* for arbitrary initial conditions $x_i(0)$.

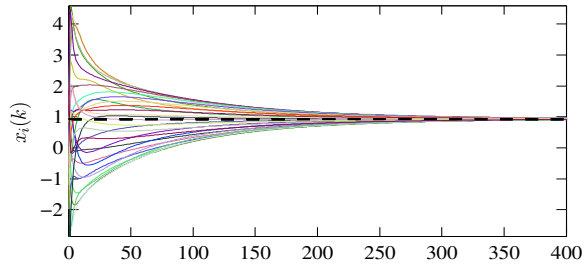

Fig. 2. Time evolution of the states $x_i(k)$, $i = 1, \ldots, N$, for $N = 30$, $\varepsilon = 0.01$, $\mathbf{v}(0) = \mathbf{w}(0) = \mathbf{y}(0) = \mathbf{z}(0) = \mathbf{0}$ and $x_i(0) \sim \mathcal{U}[-2, 2]$.

Finally, Fig. 3 illustrates the robustness of the computation of the global optimum with respect to perturbed initial conditions $\mathbf{v}(0), \mathbf{w}(0), \mathbf{y}(0), \mathbf{z}(0)$. More precisely we apply Algorithm 1 to a set of independently generated f, injecting each time artificial and independent perturbations on the initial conditions s.t. $\alpha(0), \beta(0) \sim \mathcal{U}[-\sigma, \sigma]$. Fig. 3 shows the boxplots of the errors $x_i(+\infty) - x^*$ for different σ's (300 Monte Carlo runs, $\varepsilon = 0.01$, $N = 30$).

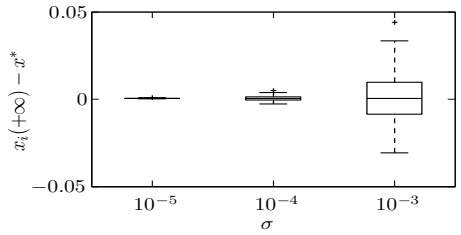

Fig. 3. Empirical distribution of the errors $x_i(+\infty) - x^*$ under artificially perturbed initial conditions $\alpha(0), \beta(0) \sim \mathcal{U}[-\sigma, \sigma]$ for different σ values.

V. Comparison with alternative distributed algorithms

We now compare our Newton-Raphson consensus (NRC) algorithm with the DSM and the ADMM methods, considering again an undirected ring graph with $N = 30$ nodes and a synchronous implementation.

DSM, as proposed in [29], alternates consensus steps on the current estimated global minimum $x_i(k)$ with subgradient updates of each $x_i(k)$ towards the minimum of the local cost f_i. To guarantee the convergence, it is required to appropriately decrease the amplitude of the local subgradient steps. Algorithm 2 presents a synchronous DSM implementation, where ρ is a tuning parameter and P is defined in (13).

Algorithm 2 DSM [29]

(storage allocation and constraints on parameters)

$\mathbf{x}^{(c)}(k), \mathbf{x}^{(\ell)}(k) \in \mathbb{R}^N$ for $k = 0, 1, \ldots$

$\rho \in \mathbb{R}_+$

(initialization)

set: $\mathbf{x}^{(\ell)}(0) = \mathbf{0}$

(main algorithm)

for $k = 0, 1, \ldots$ **do**

  $\mathbf{x}^{(c)}(k) = P\mathbf{x}^{(\ell)}(k)$

  for $i = 1, \ldots, N$ **do**

    $x_i^{(\ell)}(k+1) = x_i^{(c)}(k) - \frac{\rho}{k} f_i'\left(x_i^{(c)}(k)\right)$

The ADMM instead requires the augmentation of the system through additional constraints which do not change the optimal solution but allow the usage of Lagrange multipliers. There exist different implementations of such algorithm in a distributed context [22], [4]. Here we consider the following problem, equivalent to (1)-(2) and consistent with an undirected ring communication graph:

$$\begin{aligned} &\min_{x_1,\ldots,x_N,z_1,\ldots,z_N} \textstyle\sum_{i=1}^N f_i(x_i) \\ &\text{s.t.} \quad z_i = x_{i-1} = x_i = x_{i+1}, \quad i = 1, \ldots, N \end{aligned}$$

where $x_0 := x_N$ and $x_{N+1} := x_1$. Algorithm 3 presents a distributed implementation of the previous optimization problem which has been obtained following [8, pp. 253-261], where δ is a tuning parameter.

Fig. 4 shows a comparison of the three strategies where the tuning parameters ε, ρ and δ have been manually optimized for fastest convergence. For this specific simulation DSM is the slowest to converge, while ADMM is the fastest one. Despite being slower in this synchronous implementation, our NRC strategy can be easily adapted in an asynchronous scenarios where the topology of graph is time-varying. Differently, even if ADMM can be implemented asynchronously with some effort, it can hardly cope with time-varying topologies since the dual variables y_i strongly depend on the specific constrain imposed between the variables z_i and x_i.

Algorithm 3 ADMM [8, pp. 253-261]

(storage allocation and constraints on parameters)

$\mathbf{x}(k),\ \mathbf{z}(k),\ \mathbf{y}^{(\ell)}(k),\ \mathbf{y}^{(c)}(k),\ \mathbf{y}^{(r)}(k) \in \mathbb{R}^N$ for $k = 0, 1, \ldots$

$\delta \in (0, 1)$

$L_i(x_i, k) := f_i(x_i) + y_i^{(\ell)}(k)(x_i - z_{i-1}(k)) + y_i^{(c)}(k)(x_i - z_i(k)) + y_i^{(r)}(k)(x_i - z_{i+1}(k)) + \frac{\delta}{2}|x_i - z_{i-1}(k)|^2 + \frac{\delta}{2}|x_i - z_i(k)|^2 + \frac{\delta}{2}|x_i - z_{i+1}(k)|^2$

(initialization)

set: $\mathbf{x}(0) = \mathbf{0}$

$\mathbf{y}^{(\ell)}(0) = \mathbf{y}^{(c)}(0) = \mathbf{y}^{(r)}(0) = \mathbf{0}$

$\mathbf{z}(0) = \mathbf{0}$

(main algorithm)

for $k = 0, 1, \ldots$ **do**

  for $i = 1, \ldots, N$ **do**

    $x_i(k+1) = \arg\min_{x_i} L_i(x_i, k)$

    $z_i(k+1) = \frac{1}{3\delta}(y_{i+1}^{(\ell)}(k) + y_i^{(c)}(k) + y_{i-1}^{(r)}(k)) + \frac{1}{3}x_{i-1}(k+1) + \frac{1}{3}x_i(k+1) + \frac{1}{3}x_{i+1}(k+1)$

    $y_i^{(\ell)}(k+1) = y_i^{(\ell)}(k) + \delta(x_i(k+1) - z_{i-1}(k+1))$

    $y_i^{(c)}(k+1) = y_i^{(c)}(k) + \delta(x_i(k+1) - z_i(k+1))$

    $y_i^{(r)}(k+1) = y_i^{(r)}(k) + \delta(x_i(k+1) - z_{i+1}(k+1))$

VI. Conclusions and Future works

In this work we proposed a novel distributed convex optimization strategy that combines a Newton-Raphson optimization strategy with few parallel consensus algorithms. We also provided analysis of robustness in terms of initial conditions under some simplifying assumptions, like the use of scalar smooth convex functions and synchronous implementations. Finally we provided some numerical simulations confirming the properties of the proposed algorithm and we compared it with popular distributed optimization strategies. Although in these comparisons it did not score as the best algorithm, we believe that its strength will be more evident in an asynchronous implementation in time-varying network topologies since it inherits all the good features of consensus algorithms.

Many future research directions are open, such as the extensions to multivariable settings, to non-smooth convex functions and to asynchronous implementations. Also of

paramount importance is the design of on-line strategies for tuning parameter ε which affects the speed of convergence as well as stability.

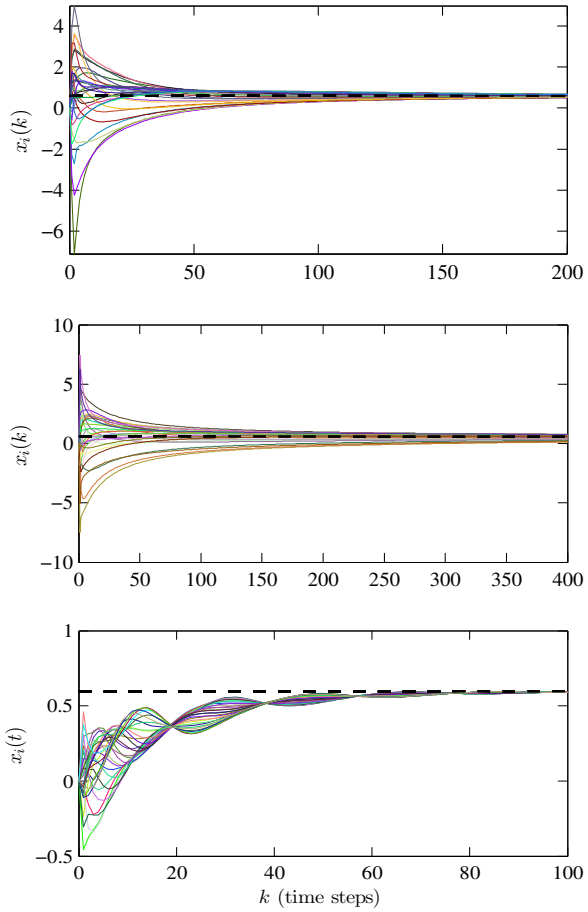

Fig. 4. Time evolution of the states $x_i(k)$, $i = 1, \ldots, N$ for the case $N = 30$. First panel, NRC (Algorithm 1, $\varepsilon = 0.8$). Second panel, DSM (Algorithm 2, $\rho = 100$). Third panel, ADMM (Algorithm 3, $\delta = 0.1$). The black dashed lines indicate the position of the global optimum x^*. Note the axes have different scales in each plot.

REFERENCES

[1] D. Puccinelli and M. Haenggi, "Wireless sensor networks: applications and challenges of ubiquitous sensing," *IEEE Circuits and Systems Magazine*, vol. 5, no. 3, pp. 19 – 31, 2005.

[2] A. Ipakchi and F. Albuyeh, "Grid of the future - are we ready to transition to a smart grid?" *IEEE Power Energy Magazine*, vol. 7, no. 2, pp. 52 – 62, 2009.

[3] P. J. Huber, *Robust statistics*. John Wiley & Sons Inc, 2009.

[4] 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.

[5] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, MIT, 1984.

[6] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Athena Scientific, 2003.

[7] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

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

[9] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Belmont, Massachusetts: Athena Scientific, 1998.

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

[11] B. Johansson, "On distributed optimization in networked systems," Ph.D. dissertation, KTH Electrical Engineering, 2008.

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

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

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

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

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

[17] A. Nedić, "Asynchronous broadcast-based convex optimization over a network," *IEEE TAC*, vol. PP, no. 99, pp. 1 – 1, 2010.

[18] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *IPSN*, 2004.

[19] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson, "Subgradient methods and consensus algorithms for solving convex optimization problems," in *CDC*, December 2008, pp. 4185 – 4190.

[20] M. G. Rabbat, R. D. Nowak, and J. A. Bucklew, "Generalized consensus computation in networked systems with erasure links," in *Sig. Proc. Advances in Wireless Comm.*, June 2005, pp. 1088 – 1092.

[21] L. Xiao, M. Johansson, and S. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Trans. on Comm.*, vol. 52, no. 7, pp. 1136 – 1144, 2004.

[22] 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. on Sig. Proc.*, vol. 56, pp. 350 – 364, 2008.

[23] C. Fischione, "F-Lipschitz optimization with Wireless Sensor Networks applications," *IEEE TAC*, vol. to appear, pp. –, 2011.

[24] C. Fischione and U. Jönsson, "Fast-Lipschitz optimization with Wireless Sensor Networks applications," in *IPSN*, 2011.

[25] J. Van Ast, R. Babška, and B. D. Schutter, "Particle swarms in optimization and control," in *IFAC World Congress*, 2008.

[26] E. Alba and J. M. Troya, "A survey of parallel distributed genetic algorithms," *Complexity*, vol. 4, no. 4, pp. 31 – 52, 1999.

[27] R. O. Saber, J. Fax, and R. Murray, "Consensus and cooperation in multi-agent networked systems," *Proceedings of IEEE*, vol. 95, no. 1, pp. 215–233, January 2007.

[28] F. Garin and L. Schenato, *Networked Control Systems*. Springer, 2011, ch. A Survey on distributed estimation and control applications using linear consensus algorithms, pp. 75–107.

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

[30] ——, "On the rate of convergence of distributed subgradient methods for multi-agent optimization," in *CDC*, 2007.

[31] A. Jadbabaie, A. Ozdaglar, and M. Zargham, "A distributed Newton method for network optimization," in *CDC*, 2009.

[32] M. Zargham, A. Ribeiro, A. Ozdaglar, and A. Jadbabaie, "Accelerated dual descent for network optimization," in *ACC*, 2011.

[33] F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-raphson consensus for distributed convex optimization," http://paduaresearch.cab.unipd.it/, Tech. Rep., 2011.

[34] P. Kokotović, H. K. Khalil, and J. O'Reilly, *Singular Perturbation Methods in Control: Analysis and Design*, ser. Classics in applied mathematics. SIAM, 1999, no. 25.

[35] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2001.

[36] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *IPSN*, 2005, pp. 63– 70.

[37] S. Bolognani, S. D. Favero, L. Schenato, and D. Varagnolo, "Consensus-based distributed sensor calibration and least-square parameter identification in wsns," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 2, pp. 176–193, 2010.

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

[39] R. Hauser and J. Nedić, "The continuous Newton-Raphson method can look ahead," *SIAM J. on Optim.*, vol. 15, pp. 915 – 925, 2005.