ISR TECHNICAL REPORT 2013-04

A non-consensus based distributed optimization algorithm

Ion Matei, John S. Baras

A non-consensus based distributed optimization algorithm

Ion Matei, John S. Baras

Abstract

In this paper we introduce a discrete-time, distributed optimization algorithm executed by a set of agents whose interactions are subject to a communication graph. The algorithm can be applied to optimization costs that are expressed as sums of functions, where each function is associated to an agent. The algorithm can be applied to continuously differentiable cost functions, it is not consensus-based and is derived naturally by solving the first order necessary conditions of a lifted optimization problem with equality constraints. We show that, provided the agents' initial values are sufficiently closed to a local minimizer and the step-size is sufficiently small, each agent converges to the local minimizer at a linear rate. In addition, we revisit two popular consensus-based distributed optimization algorithms and give sufficient conditions so that there use is extended to non-convex functions as well. We take a closer look at their rate of convergence and also show that unlike our algorithm, for a constant step-size, the consensus-based algorithms do not converge to a local minimizer even though the agents start close enough to the local minimizer.

I. Introduction

Recent years' technological advances in wireless networks re-fueled the interest of the research community in applications where complex tasks are executed over large networks, by a large set of agents. Such applications can include autonomous/unmanned vehicles, parallel computing, sensor networks for monitoring and tracking, and so on. The execution of these applications over large networks makes a centralized coordination unfeasible. As a consequence, researchers have looked for distributed strategies where, although agent make decisions based on limited information, the overall result is comparable with the result obtained had a centralized strategy been used.

Multi-agent distributed optimization problems appear naturally in many distributed applications such as network resource allocation, collaborative control, estimation and identification, and so on. In these type of applications a group of agents has as common goal the optimization of a cost function under limited information and resources. The limited information may be induced by the fact that an agent can communicate with only a subset of the total set of agents, or/and by the fact that an agent is aware of only a part of the cost function or constraint sets.

A particular formulation of a distributed optimization problem refers to the case where the optimization cost is expressed as a sum of functions and each function in the sum corresponds to an agent. In this formulation the agents interact with each other subject to a communication network, usually modeled as a directed/undirected graph. This formulation is often found in resource allocation for wireless networks problems [12] or in finite horizon optimal control problems with separable cost functions [2].

A first distributed algorithm for solving an optimization problem of the type described above was introduced in [11]. The algorithm, referred to as "distributed subgradient method", is used to minimize a convex function expressed as a sum of convex functions:

$$\min_x \sum_{i=1}^N f_i(x).$$

In this algorithm, in addition to the standard (sub)gradient step, each agent executes a consensus step to deal with lack of complete information on the cost function, and is given by:

$$x_{i,k+1} = \sum_{j=1}^N a_{ij} x_{j,k} - \alpha_{i,k} d_{i,k}, \tag{1}$$

where the indices i (or j) and k refer to agents and discrete time, respectively, a_{ij} are the entries of a stochastic matrix whose structure depends on the communication graph, $d_{i,k}$ is the (sub)gradient of the function $f_i(x)$, computed at $x_{i,k}$, and $\alpha_{i,k}$ is the step-size of the (sub)gradient descent step. Under suitable assumptions on the step-size, the cost functions and the entries of the stochastic matrix, it is shown in [11] that iteration (1) indeed converges to a minimizer.

Many subsequent versions of this algorithm appeared in the literature. The introduction of communication noise and errors on subgradients was addressed in [10], [13], while the case where the communication network is modeled as a random graph was treated in [7], [9]. Analyses of asynchronous versions of the algorithm can be found in [10], [15]. A further extension was proposed in [8], where the authors considered state-dependent communication topologies.

A modified version of the distributed subgradient method was introduced in [5], [6], where the authors change the order in which the two operations of the algorithm are performed. More specifically, first the subgradient descent step is executed, followed by the consensus step, and takes the form

$$x_{i,k+1} = \sum_{j=1}^{N} \left(a_{ij} x_{j,k} - \alpha_{j,k} d_{j,k} \right).$$

The algorithms discussed above became popular in the signal processing community as well, being used for solving distributed filtering and parameter identification problems [3], [14].

In this paper we study a distributed optimization problem similar to the formulation proposed in [11], namely the goal is to minimize a function expressed as a sum of functions, where each function in the sum is associated to an agent. We do not make any convexity assumptions on the functions, but we assume they are continuously differentiable. We propose a distributed, discrete-time algorithm that guarantees convergence to a local minimizer (at a linear rate), provided that the initial values of the agents are close enough to the minimizer and a sufficiently small step-size is used. The most interesting aspect of this algorithm is that *it is not a heuristic algorithm*, but follows naturally from solving a *lifted* optimization problem with equality constraints, that can be prove to be equivalent (in a sense discussed later) with the original optimization problem. Specifically, the algorithm comes as a result of applying a first order method to solve a set of equations representing the first order necessary conditions of the lifted optimization problem. In addition, we revisit the consensus-based distributed optimization algorithm introduced in [11] and [6] and provide sufficient conditions so that their use is extended to non-convex functions as well. We show that if a constant step-size is used, unlike our algorithm, these consensus-based algorithms do not guarantee convergence to a local minimizer, even in the case where the initial values of the agents are close enough to the minimizer and a sufficiently small step-size is used, so that the stability of the algorithms is ensured. Moreover, we take a closer look to their rate of convergence.

The paper is organized as following: in Section II we describe the setup of the optimization problem and introduce a distributed optimization algorithm to solve it. Section III presents the idea behind our algorithm, by connecting the algorithm with solving a lifted optimization problem equivalent to the original problem. In Section IV we state and prove a number of auxiliary results used for the local convergence analysis of the algorithm; analysis shown in Section V. Section VI takes a closer look at the convergence properties of the algorithm and qualitatively connects the connectivity of the network with the rate of convergence of the algorithm. In Section VII we revisit two popular consensus-based distributed optimization algorithms, for which we give sufficient conditions for stability that can be used even when the cost functions are non-convex. We end the paper with some numerical simulations and conclusions.

Notation and definitions: For a matrix A, an entry (i,j) of this matrix is denoted by $[A]_{ij}$. If A is a symmetric matrix, $A \succ 0$ ($A \succeq 0$) means that A is positive (semi-positive) definite. The symbol $\otimes$ denotes the Kronecker product. If A is a matrix, Null(A) and Range(A) refer to the nullspace and range of A, respectively. The vector of all ones is denoted by $\mathbb{1}$. The set of eigenvalues of a matrix $A \in \mathbb{R}^{n\times n}$ is denoted by $\sigma(A) = \{\sigma_{1,A}, \sigma_{2,A}, \ldots, \sigma_{n,A}\}$, where $\sigma_{i,A} \leq \sigma_{j,A}$ if $i < j$. Given a vector x, a open ball around x^* of size ϵ is denoted by $B(x^*, \epsilon) \triangleq \{x \mid \|x - x^*\| < \epsilon\}$ while a closed ball is denoted by $B_c(x^*, \epsilon) \triangleq \{x \mid \|x - x^*\| \leq \epsilon\}$.

Let S be a set of vectors. By $x+S$ we understand the set of vector produced by adding x to each element of S, that is, $x+S \triangleq \{x+y \mid y \in S\}$. Let $\|\cdot\|$ be a vector norm, x a vector and S a set of vectors. By $\|x-S\|$ we denote the distance between the vector x and the set S, that is, $\|x-S\| \triangleq \inf_{y\in S} \|x-y\|$. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function. We denote by $\nabla f(x)$ and by $\nabla^2 f(x)$ the gradient and the Hessian of f at x, respectively. Let $\{A_i\}_{i=1}^N$ be a set of matrices. By $\mathrm{diag}(A_i, i = 1,\dots,N)$ we understand a block diagonal matrix, where the i^{th} block matrix is given by A_i. We say that the set $\mathcal{X}$ is an *attractor* for the dynamics $x_{k+1} = f(x_k)$, if there exists $\epsilon > 0$ so that for any $x_0 \in S_\epsilon$, with $S_\epsilon = \{x \mid \|x-\mathcal{X}\| < \epsilon\}$, $\lim_{k\to\infty} \|x_k - \mathcal{X}\| = 0$.

II. Problem description

In this section we describe the setup of our problem. We present first the communication model after which we introduce the optimization model and the distributed optimization algorithm.

A. Communication model

A set of N agents interact with each through a communication topology modeled as an undirected communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1,2,\dots,N\}$ is the set of nodes and $\mathcal{E} = \{e_{ij} \mid i, j = 1,\dots,N, i \neq j\}$ is the set of edges. An edge between two nodes i and j means that agents i and j can exchange information (or can cooperate). We denote by $\mathcal{N}_i \triangleq \{j \mid e_{ij} \in \mathcal{E}\}$ the set of neighbors of agent i, and by L the Laplacian of graph $\mathcal{G}$ defined as

$$[L]_{ij} = \begin{cases} -l_{ij} & j \in \mathcal{N}_i, \\ \sum_{j\in\mathcal{N}_i} l_{ij} & i = j, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

where l_{ij} are positive scalars chosen a priori that can be interpreted as weights put on the information transmitted on the links (i, j).

In the next sections we are going to make use of a set of properties of a (weighted) Laplacian of a graph; properties that are grouped in the following remark.

Remark 2.1: The Laplacian L of a connected graph satisfies the following properties:

(a) The matrix L has only one eigenvalue zero and the corresponding right and left eigenvectors are $\mathbb{1}$ and η, where η is a vector with non-zero entries of the same sign;

(b) The nullspaces of L and L' are given by $\mathrm{Null}(L) = \{\gamma\mathbb{1} \mid \gamma \in \mathbb{R}\}$, and $\mathrm{Null}(L') = \{\gamma\eta \mid \gamma \in \mathbb{R}\}$, respectively;

(c) Let $\mathbf{L} = L \otimes I$, where I is the n-dimensional identity matrix. Then the nullspaces of $\mathbf{L}$ and $\mathbf{L}'$ are given by $\mathrm{Null}(\mathbf{L}) = \{\mathbb{1} \otimes x \mid x \in \mathbb{R}^n\}$, and $\mathrm{Null}(\mathbf{L}') = \{\eta \otimes x \mid x \in \mathbb{R}^n\}$, respectively;

(d) Let $\mathbf{x}$ be a vector in $\mathbb{R}^{nN}$. Then the orthogonal projection of $\mathbf{x}$ on $\mathrm{Null}(\mathbf{L}')$ is given by $\mathbf{x}_\perp = \mathbf{J}\mathbf{x}$, where $\mathbf{J}$ is the orthogonal projection matrix (operator) defined as

$$\mathbf{J} \triangleq \frac{\eta\eta'}{\eta'\eta} \otimes I,$$

with η the left eigenvector of L corresponding to the zero eigenvalue.

B. Optimization model

We consider a function $f : \mathbb{R}^n \to \mathbb{R}$ expressed as a sum of N functions

$$f(x) = \sum_{i=1}^{N} f_i(x).$$

We make the following assumptions on the functions $f_i(x)$ and on the communication model.

Assumption 2.1: (a) Functions $f_i(x)$, $i = 1,\dots,N$ are twice continuously differentiable;

(b) Agent i has knowledge of only function $f_i(x)$ and scalars l_{ij}, for $j \in \mathcal{N}_i$;

(c) Agent i can exchange information only with agents belonging to the set of its neighbors $\mathcal{N}_i$;

(d) The communication graph $\mathcal{G}$ is connected.

The common goal of the agents is to minimize the following optimization problem

$$(P_1) \quad \min_{x\in\mathbb{R}^n} \quad f(x)$$

under Assumptions 2.1. Through the rest of the paper we assume that problem (P_1) has at least one local minimizer.

C. Distributed algorithms

Let x^* be a local minimizer of (P_1) and let $x_{i,k}$ denote agent i's *estimate* of x^*, at time-slot k. We propose the following distributed algorithm to solve the problem (P_1), referred henceforth as algorithm (A_1):

$$x_{i,k+1} = x_{i,k} - \alpha\nabla f_i(x_{i,k}) - \alpha \sum_{j\in\mathcal{N}_i}\left(l_{ij}\lambda_{i,k} - l_{ji}\lambda_{j,k}\right), \tag{3}$$

$$\lambda_{i,k+1} = \lambda_{i,k} + \alpha \sum_{j\in\mathcal{N}_i} l_{ij}(x_{i,k} - x_{j,k}), \tag{4}$$

where $\alpha > 0$ is the step-size of the algorithm and $\nabla f_i(x_{i,k})$ denotes the gradient of function $f_i(x)$ computed at $x_{i,k}$. In addition, the positive scalars l_{ij} are the entries of the Laplacian L of the graph $\mathcal{G}$ defined in (2).

Remark 2.2: Note that although the graph $\mathcal{G}$ is assumed undirected, the Laplacian L is not necessarily symmetric since we may have $l_{ij} \neq l_{ji}$. However, if $l_{ij} \neq 0$ then we must also have that $l_{ji} \neq 0$. In other words, if agent i sends information to agent j, agent j must send information to agent i, as well. It turns out that for a Laplacian satisfying these properties, the matrix $L'L$ is symmetric.

In Algorithm (A_1) the index i (or j) designates an agent while k denotes the discrete time. It can be observed that the algorithm is indeed distributed since for updating its current estimate $x_{i,k}$ agent i uses only *local* information, that is, its own information ($x_{i,k}$, $\lambda_{i,k}$ and $\nabla f_i(x_{i,k})$) and information from its neighbors ($x_{j,k}$, $\lambda_{j,k}$, for $j \in \mathcal{N}_i$). Therefore, at each time instant, agent i shares with its neighbors the quantities $x_{i,k}$ and $l_{ij}\lambda_{i,k}$. Equation (3) is comprised of a standard gradient descent step and one additional term used to cope with the lack of complete information. The exact origin of equation (4) will be made clear in the next sections. Intuitively however, $\lambda_{i,k}$ can be interpreted as the price paid by agent i for having its estimate $x_{i,k}$ far away from the estimates of its neighbors.

Remark 2.3: We made the assumption that the graph $\mathcal{G}$ is undirected. This assumption is in fact crucial for the implementation of the algorithm (A_1) in a distributed manner. Indeed, consider a directed graph with three nodes, where the neighborhoods of the nodes are $\mathcal{N}_1 = \{2\}$, $\mathcal{N}_2 = \{3\}$ and $\mathcal{N}_3 = \{1\}$. In this case the non-weighted Laplacian of the graph is given by

$$L = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix} \text{ and } L' = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix},$$

and the algorithm (A_1) becomes

$$\begin{aligned}
x_{1,k+1} &= x_{1,k} - \alpha\left[\lambda_{1,k} - \lambda_{3,k}\right] - \alpha\nabla_x f_1(x_{1,k}),\\
x_{2,k+1} &= x_{2,k} - \alpha\left[\lambda_{2,k} - \lambda_{1,k}\right] - \alpha\nabla_x f_2(x_{2,k}),\\
x_{3,k+1} &= x_{3,k} - \alpha\left[\lambda_{3,k} - \lambda_{2,k}\right] - \alpha\nabla_x f_3(x_{3,k}),\\
\lambda_{1,k+1} &= \lambda_{1,k} + \alpha\left[x_{1,k} - x_{2,k}\right],\\
\lambda_{2,k+1} &= \lambda_{2,k} + \alpha\left[x_{2,k} - x_{3,k}\right],\\
\lambda_{3,k+1} &= \lambda_{3,k} + \alpha\left[x_{3,k} - x_{1,k}\right].
\end{aligned}$$

Note that although agent i can update $\lambda_{i,k}$ using only information from its neighbors, the estimate of the minimizer $x_{i,k}$ cannot be updated since it requires information from agents outside his neighborhood.

In the next sections we start building the infrastructure that will allow us to prove local convergence of Algorithm (A_1).

III. An equivalent optimization problem with equality constraints

In this section we define a lifted optimization problem, from whose solution we can in fact extract the solution of problem (P_1). As made clear in what follows, Algorithm (A_1) comes as a result of applying a first-order method to solve the first order necessary conditions of the lifted optimization problem.

Let us define the function $\mathbf{F}:\mathbb{R}^{nN}\rightarrow\mathbb{R}$ given by

$$\mathbf{F}(\mathbf{x})=\sum_{i=1}^{N} f_i(x_i),$$

where $\mathbf{x}'=(x_1',x_2',\ldots,x_N')$, with $x_i\in\mathbb{R}^n$. In addition we introduce the vector-valued function $\mathbf{g}:\mathbb{R}^{nN}\rightarrow\mathbb{R}^{nN}$, where

$$\mathbf{g}(\mathbf{x})'=(g_1(\mathbf{x})',g_2(\mathbf{x})',\ldots,g_N(\mathbf{x})'),$$

with $g_i:\mathbb{R}^{nN}\rightarrow\mathbb{R}^n$ given by

$$g_i(\mathbf{x})=\sum_{j\in\mathcal{N}_i} l_{ij}(x_i-x_j),$$

where l_{ij} are the entries of the Laplacian L defined in (2). The vector-valued function $\mathbf{g}(\mathbf{x})$ can be compactly expressed as

$$g(\mathbf{x})=\mathbf{L}\mathbf{x},$$

where $\mathbf{L}=L\otimes I$, with I the n-dimensional identity matrix.

We define the optimization problem

$$(P_2)\quad \min_{\mathbf{x}\in\mathbb{R}^{nN}}\quad \mathbf{F}(\mathbf{x}), \tag{5}$$

$$\mathbf{g}(\mathbf{x})=\mathbf{L}\mathbf{x}=0. \tag{6}$$

The following proposition states that by solving (P_2) we solve in fact (P_1) as well, and vice-versa.

Proposition 3.1: Let Assumptions 2.1 hold. The vector x^* is a local minimizer of (P_1) if and only if $\mathbf{x}^*=\mathbb{1}\otimes x^*$ is a local minimizer of (P_2).

Proof: Since the Laplacian L corresponds to a connected graph, according to Remark 2.1-(c), the nullspace of $\mathbf{L}$ is given by $\text{Null}(\mathbf{L})=\{\mathbb{1}\otimes x \mid x\in\mathbb{R}^n\}$. From the equality constraint (6), we get that any local minimizer $\mathbf{x}^*$ of (P_2) must be of the form $\mathbf{x}^*=\mathbb{1}\otimes x^*$, for some $x^*\in\mathbb{R}^n$. Therefore, the solution of (P_2) must be searched in the set of vectors with structure given by $\mathbf{x}=\mathbb{1}\otimes x$. Applying this constraint, the cost function (5) becomes

$$\mathbf{F}(\mathbf{x})=\sum_{i=1}^{N} f_i(x)=f(x),$$

which shows that we have recovered the optimization problem (P_1). ■

Remark 3.1: We note from the above proposition the importance of having a connected communication topology. Indeed, if $\mathcal{G}$ is not connected, then the nullspace of $\mathbf{L}$ is much richer than $\{\mathbb{1}\otimes x \mid x\in\mathbb{R}^n\}$, and therefore the solution of (P_2) may not necessarily be of the form $\mathbf{x}^*=\mathbb{1}\otimes x^*$. However, the fact that we search a solution of (P_2) of this particular structure is *fundamental* for showing the equivalence of the two optimization problems.

IV. Auxiliary results

In this section we recall and prove a number of results concerning the optimization problem (P_2). They will be used for addressing the local converging properties of Algorithm (A_1).

Our first result characterizes the tangent cone at a local minimizer of (P_2) and it is going to be used to formulate the first order necessary conditions of (P_2).

Proposition 4.1: Let Assumption 2.1 hold, let $\mathbf{x}^* = \mathbb{1} \otimes x^*$ be a local minimizer of (P_2) and let $\mathbf{\Omega}$ denote the constraint set, that is, $\mathbf{\Omega} = \{\mathbf{x} \mid \mathbf{L}\mathbf{x} = 0\}$. Then the tangent cone to $\mathbf{\Omega}$ at $\mathbf{x}^*$ is given by

$$\mathrm{TC}(\mathbf{x}^*, \mathbf{\Omega}) = \mathrm{Null}(\mathbf{L}).$$

Proof: All we have to show is that any vector in $\mathrm{Null}(\mathbf{L})$ belongs to $\mathrm{TC}(\mathbf{x}^*, \mathbf{\Omega})$ as well, since it is well known that (the closure of the convex hull of) $\mathrm{TC}(\mathbf{x}^*, \mathbf{\Omega})$ is included in $\mathrm{Null}(\mathbf{L})$. Let $\mathbf{u} \in \mathrm{Null}(\mathbf{L})$ and therefore

$$\mathbf{L}\mathbf{u} = 0. \tag{7}$$

From equation (7), $\mathbf{u}$ must be of the form $\mathbf{u} = \mathbb{1} \otimes u$, for some $u \in \mathbb{R}^n$.

We need to show that a vector $\mathbf{u} = \mathbb{1} \otimes u$, with $u \in \mathbb{R}^n$ belongs to $\mathrm{TC}(\mathbf{x}^*, \mathbf{\Omega})$. More explicitly, using the definition of the tangent cone, we must find a function $\mathbf{o} : \mathbb{R} \to \mathbb{R}^{nN}$, with $\lim_{t\to 0, t>0} \frac{\mathbf{o}(t)}{t} = 0$, so that

$$\mathbf{x}^* + t\mathbf{u} + \mathbf{o}(t) \in \mathbf{\Omega} \ \forall t > 0.$$

Choose an arbitrary function $o : \mathbb{R} \to \mathbb{R}^n$ that satisfies $\lim_{t\to 0, t>0} \frac{o(t)}{t} = 0$. Setting $\mathbf{o}(t) = \mathbb{1} \otimes o(t)$, we note that $\mathbf{x}^* + t\mathbf{u} + \mathbf{o}(t) = \mathbb{1} \otimes (x^* + tu + o(t))$ and therefore

$$\mathbf{g}(\mathbf{x}^* + t\mathbf{u} + \mathbf{o}(t)) = \mathbf{L}(\mathbf{x}^* + t\mathbf{u} + \mathbf{o}(t)) = 0 \ \forall t > 0.$$

Consequently $\mathbf{u} \in \mathrm{TC}(\mathbf{x}^*, \mathbf{\Omega})$ and $\mathrm{TC}(\mathbf{x}^*, \mathbf{\Omega})$ is a closed and convex subspace. ■

Let $\mathbf{x}^* = \mathbb{1} \otimes x^*$ denote a local minimizer of (P_2). From the theory concerning optimization problems with equality constraints (see for example Chapter 3, page 15 of [17], or Chapter 3, page 253 of [1]), the first order necessary conditions for (P_2) ensure the existence of $\lambda_0^* \in \mathbb{R}$ and $\boldsymbol{\lambda}^* \in \mathbb{R}^{nN}$ so that

$$\lambda_0^* \nabla \mathbf{F}(\mathbf{x}^*) + \mathbf{L}' \boldsymbol{\lambda}^* = 0.$$

Note that since $\mathbf{L}$ is not full rank, the uniqueness of $\boldsymbol{\lambda}^*$ cannot be guaranteed. The following result characterizes the set of Lagrange multipliers verifying the first order necessary conditions of (P_2).

Proposition 4.2 (first order necessary conditions for (P_2)): Let Assumptions 2.1 hold and let $\mathbf{x}^* = \mathbb{1}_N \otimes x^*$ be a local minimizer for problem (P_2). There exists a unique vector $\boldsymbol{\lambda}^* \in \mathrm{Range}(\mathbf{L})$ so that

$$\nabla \mathbf{F}(\mathbf{x}^*) + \mathbf{L}' \boldsymbol{\lambda} = 0,$$

for all $\boldsymbol{\lambda} \in \{\boldsymbol{\lambda}^* + \boldsymbol{\lambda}_\perp \mid \boldsymbol{\lambda}_\perp \in \mathrm{Null}(\mathbf{L}')\}$.

Proof: By Lemma 1[1] , page 50 of [17] we have that $\nabla \mathbf{F}(\mathbf{x}^*)$ is orthogonal on the nullspace of $\mathbf{L}$ and therefore $\nabla \mathbf{F}(\mathbf{x}^*)$ must belong to $\mathrm{Range}(\mathbf{L}')$. Consequently, there exists a vector $\boldsymbol{\lambda} \in \mathbb{R}^{nN}$ so that

$$-\nabla \mathbf{F}(\mathbf{x}^*) = \mathbf{L}' \boldsymbol{\lambda}. \tag{8}$$

Note the $\mathbb{R}^{nN}$ can be written as a direct sum between the nullspace of $\mathbf{L}'$ and the range of $\mathbf{L}$, that is $\mathbb{R}^{nN} = \mathrm{Null}(\mathbf{L}') \oplus \mathrm{Range}(\mathbf{L})$. Consequently, there exist the orthogonal vectors $\boldsymbol{\lambda}^* \in \mathrm{Range}(\mathbf{L})$ and $\boldsymbol{\lambda}_\perp \in \mathrm{Null}(\mathbf{L}')$ so that $\boldsymbol{\lambda} = \boldsymbol{\lambda}^* + \boldsymbol{\lambda}_\perp$. Note that we can replace $\boldsymbol{\lambda}_\perp$ by any vector in $\mathrm{Null}(\mathbf{L}')$ and (8) will still hold. The only thing left is to prove the uniqueness of $\boldsymbol{\lambda}^*$. We use a contradiction argument. Let $\tilde{\boldsymbol{\lambda}}$ be another vector in $\mathrm{Range}(\mathbf{L})$ so that any vector of the form $\boldsymbol{\lambda} = \tilde{\boldsymbol{\lambda}} + \mathrm{Null}(\mathbf{L}')$ satisfies (8). Hence we have that $-\nabla \mathbf{F}(\mathbf{x}^*) = \mathbf{L}' \boldsymbol{\lambda}^*$, and $-\nabla \mathbf{F}(\mathbf{x}^*) = \mathbf{L}' \tilde{\boldsymbol{\lambda}}$ which gives $0 = \mathbf{L}' \left(\boldsymbol{\lambda}^* - \tilde{\boldsymbol{\lambda}} \right)$. On the one hand, this means that

[1] The results states that for any $h \in TC(x^*, \Omega)$, we have $h' \nabla_{\mathbf{x}} \mathbf{F}(\mathbf{x}^*) \geq 0$, but since $TC(x^*, \Omega)$ is a subspace, orthogonality follows.

$\lambda^* - \tilde{\lambda} \in \text{Null}(\mathbf{L}')$. On the other hand, since $\text{Range}(\mathbf{L})$ is closed under addition, $\lambda^* - \tilde{\lambda} \in \text{Range}(\mathbf{L})$, as well. Therefore, $\lambda^* - \tilde{\lambda}$ must be the zero vector, or equivalently $\lambda^* = \tilde{\lambda}$ and the result follows. ■

Remark 4.1: From the above Proposition 4.2, given that $\mathbf{x}^*$ is a local minimizer of (P_2), we have that $\nabla\mathbf{F}(\mathbf{x}^*)$ is orthogonal on the nullspace of $\mathbf{L}$. Equivalently, $\nabla\mathbf{F}(\mathbf{x}^*)'\mathbf{h} = 0$ for any $\mathbf{h} \in \text{Null}(\mathbf{L})$, or $h'\left[\sum_{i=1}^{N} \nabla f_i(x^*)\right] = 0$ for any $h \in \mathbb{R}^n$, where $\mathbf{x}^* = \mathbb{1} \otimes x^*$ and $\mathbf{h} = \mathbb{1} \otimes h$. Consequently, $\sum_{i=1}^{N} \nabla f_i(x^*) = \nabla f(x^*) = 0$, that is, we have recovered (as expected) the first order necessary optimality condition for (P_1).

In the next section, we are going to make use of the spectral properties of a particular type of matrix; properties analyzed in the next result.

Proposition 4.3: Let Assumptions 2.1-(d) hold and let α and $\mathbf{H}$ be a positive scalar and a positive-definite matrix, respectively. Then the eigenvalues of the matrix

$$\mathbf{B} = \begin{pmatrix} \mathbf{H} & \mathbf{L}' \\ -\mathbf{L} & \frac{1}{\alpha}\mathbf{J} \end{pmatrix} \tag{9}$$

have positive real parts, where $\mathbf{J} = \frac{\eta\eta'}{\eta'\eta} \otimes I$, with η the left eigenvector of L, corresponding to the zero eigenvalue. In addition, there are a number of n eigenvalues equal to $1/\alpha$ and they correspond to the eigenspace $\{\mathbf{x} \mid \mathbf{x} = (0', \ x' \otimes \eta')', \ x \in \mathbb{R}^n\}$.

Proof: Let β be an eigenvalue of $\mathbf{B}$ and let $(\mathbf{u}', \mathbf{v}')' \neq 0$ be the corresponding eigenvector, where $\mathbf{u}$ and $\mathbf{v}$ are complex vectors of appropriate dimensions. Denoting by $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ the conjugates of $\mathbf{u}$ and $\mathbf{v}$, respectively, we have

$$\begin{aligned} \text{Re}(\beta)\left(\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2\right) = \text{Re}\left\{\hat{\mathbf{u}}'\mathbf{H}\mathbf{u} + \hat{\mathbf{u}}'\mathbf{L}'\mathbf{v} - \hat{\mathbf{v}}'\mathbf{L}\mathbf{u} + \hat{\mathbf{v}}'\frac{1}{\alpha}\mathbf{J}\mathbf{v}\right\} = \\ \text{Re}\{\hat{\mathbf{u}}'\mathbf{H}\mathbf{u}\} + \text{Re}\left\{\hat{\mathbf{v}}'\frac{1}{\alpha}\mathbf{J}\mathbf{v}\right\}. \end{aligned} \tag{10}$$

Since $\mathbf{J} \triangleq \frac{\eta\eta'}{\eta'\eta} \otimes I$ is semi-positive definite and $\mathbf{H}$ is positive definite we have that

$$\text{Re}(\beta)\left(\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2\right) > 0, \ \forall \ \mathbf{u} \neq 0.$$

Therefore for all $\mathbf{u} \neq 0$ we have that $\text{Re}(\beta) > 0$. In the case $\mathbf{u} = 0$ we get

$$\mathbf{B}\begin{pmatrix} 0 \\ \mathbf{v} \end{pmatrix} = \beta \begin{pmatrix} 0 \\ \mathbf{v} \end{pmatrix},$$

from where we obtain $\mathbf{L}'\mathbf{v} = 0$ and $\frac{1}{\alpha}\mathbf{J}\mathbf{v} = \beta\mathbf{v}$. But this means that $\mathbf{v} = \eta \otimes v$ for some $v \in \mathbb{R}^n$ which leads to

$$\frac{1}{\alpha}\mathbf{J}\mathbf{v} = \frac{1}{\alpha}\mathbf{v} = \beta\mathbf{v},$$

where the second equality followed from the fact that the projection of $\mathbf{v}$ on $\text{Null}(\mathbf{L}')$ is $\mathbf{v}$ itself and consequently $\beta = 1/\alpha$. Therefore all eigenvalues of $\mathbf{B}$ have positive real parts, $1/\alpha$ is an eigenvalue of $\mathbf{B}$ and its corresponding eigenspace is $\{\mathbf{x} \mid \mathbf{x}' = (0', \ x' \otimes \eta'), \ x \in \mathbb{R}^n\}$. In addition, there are a number of n eigenvalues equal to $1/\alpha$ since every eigenvalue of L appears n times in the matrix $\mathbf{L}$ due to the Kronecker product properties. ■

We are ending this section by recalling a immediate result on the spectral properties of a continuous Hessian.

Proposition 4.4: Let $\mathbf{F}(\mathbf{x})$ be twice continuously differentiable and assume that its Hessian is positive definite at $\mathbf{x}^*$, that is, $\nabla^2\mathbf{F}(\mathbf{x}^*) \succ 0$. Then there exist positive scalars θ, m_l and m_u, with $m_l \leq m_u$, so that

$$\nabla^2\mathbf{F}(\mathbf{x}) \succ 0 \ \forall \mathbf{x} \in B_c(\mathbf{x}^*, \theta), \tag{11}$$

and

$$m_l\mathbf{I} \preceq \nabla^2\mathbf{F}(\mathbf{x}) \preceq m_u\mathbf{I} \ \forall \mathbf{x} \in B_c(\mathbf{x}^*, \theta). \tag{12}$$

Proof: Follows from the continuity of $\nabla^2\mathbf{F}(\mathbf{x})$. ■

V. Convergence analysis of Algorithm (A_1)

In this section we analyze the convergence properties of Algorithm (A_1). Since the matrix $\mathbf{L}$ is not full rank, we cannot directly apply existing results for regular (local) minimizers, such as Proposition 4.4.2, page 388, [1]. Still, for a local minimizer and Lagrange multiplier pair $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$, with $\boldsymbol{\lambda}^* \in \text{Range}(\mathbf{L})$, we show that if the initial value $\mathbf{x}_0$ is close enough to $\mathbf{x}^*$, for a small enough step-size and under some conditions on (the Hessians of) the functions $f_i(x)$, $i = 1, \ldots, N$, the sequence $\{\mathbf{x}_k\}$ does indeed converge to $\mathbf{x}^*$. However, although under the same conditions the sequence $\{\boldsymbol{\lambda}_k\}$ does converge, it cannot be guaranteed to converge to the unique $\boldsymbol{\lambda}^* \in \text{Range}(\mathbf{L})$ but rather to a point in the set $\{\boldsymbol{\lambda}^* + \text{Null}(\mathbf{L}')\}$.

In trying to find the solution for problem (P_2) the first thing we can think about is solving the set of necessary conditions:

$$\begin{aligned} \nabla \mathbf{F}(\mathbf{x}) + \mathbf{L}'\boldsymbol{\lambda} &= 0, && (13) \\ \mathbf{L}\mathbf{x} &= 0. && (14) \end{aligned}$$

Solving (13) and (14) does not guarantees finding local minimizers, but at least they are among the solutions of the above nonlinear system of equations. We can use a first order method to solve (13) and (14) (see for instance Section 4.4.1, page 386, [1]), given by:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - \alpha\left[\nabla \mathbf{F}(\mathbf{x}_k) + \mathbf{L}'\boldsymbol{\lambda}_k\right], && (15) \\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \alpha \mathbf{L}\mathbf{x}_k. && (16) \end{aligned}$$

Expressing the above algorithm for each of the n-dimensional components of the vectors $\mathbf{x}_k$ and $\boldsymbol{\lambda}_k$, we in fact recover algorithm (A_1), which shows the *non-heuristic* and *distributed* nature of the algorithm.

The following theorem addresses the local convergence properties of Algorithm (A_1).

Theorem 5.1: Let Assumptions 2.1 hold, let $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ be a local minimizer and Lagrange multiplier pair of (P_2), with $\boldsymbol{\lambda}^* \in \text{Range}(\mathbf{L})$. Assume also that $\nabla^2 \mathbf{F}(\mathbf{x}^*) \succ 0$. Then there exist positive integers $\theta > 0$ and $\bar{\alpha} > 0$ so that under iteration (15)-(16)

$$\lim_{k\to\infty} \|\mathbf{x}_k - \mathbf{x}^*\| = 0, \tag{17}$$

$$\lim_{k\to\infty} \|\boldsymbol{\lambda}_k - [\boldsymbol{\lambda}^* + \text{Null}(\mathbf{L}')]\| = 0 \tag{18}$$

for all $\mathbf{x}_0 \in B_c(\mathbf{x}^*, \theta)$ and $\alpha \in (0, \bar{\alpha}]$. In addition, the rate of convergence is linear.

Proof: From the Assumption 2.1-(a), according to the mean value theorem there exists a positive integer $\varphi \in (0,1)$ so that the gradient of $\mathbf{F}(\mathbf{x})$ at $\mathbf{x}_k$ can be expressed as

$$\nabla \mathbf{F}(\mathbf{x}_k) = \nabla \mathbf{F}(\mathbf{x}^*) + \nabla^2 \mathbf{F}(\mathbf{y}_k)(\mathbf{x}_k - \mathbf{x}^*),$$

where $\mathbf{y}_k \triangleq \varphi \mathbf{x}^* + (1-\varphi)\mathbf{x}_k$. Subtracting $\mathbf{x}^*$ from the right- and left-hand sides of equation (15), and using the above expansion of the gradient $\nabla \mathbf{F}(\mathbf{x}_k)$ we obtain

$$\mathbf{x}_{k+1} - \mathbf{x}^* = \mathbf{x}_k - \mathbf{x}^* - \alpha \nabla^2 \mathbf{F}(\mathbf{y}_k)(\mathbf{x}_k - \mathbf{x}^*) - \alpha \nabla \mathbf{F}(\mathbf{x}^*) - \alpha \mathbf{L}'\boldsymbol{\lambda}_k,$$

or

$$\begin{aligned} \mathbf{x}_{k+1} - \mathbf{x}^* &= \mathbf{x}_k - \mathbf{x}^* - \alpha \nabla^2 \mathbf{F}(\mathbf{y}_k)(\mathbf{x}_k - \mathbf{x}^*) - \alpha \nabla \mathbf{F}(\mathbf{x}^*) \\ &- \alpha \mathbf{L}'\boldsymbol{\lambda}_k - \alpha \mathbf{L}'\boldsymbol{\lambda}^* + \alpha \mathbf{L}'\boldsymbol{\lambda}^*. \end{aligned}$$

From the above we get

$$\mathbf{x}_{k+1} - \mathbf{x}^* = \mathbf{x}_k - \mathbf{x}^* - \alpha\left[\nabla^2 \mathbf{F}(\mathbf{y}_k)(\mathbf{x}_k - \mathbf{x}^*) + \mathbf{L}'(\boldsymbol{\lambda}_k - \boldsymbol{\lambda}^*)\right], \tag{19}$$

where the last equality followed from the first order necessary conditions. Proceeding in a similar manner, we subtract $\boldsymbol{\lambda}^*$ from the left- and right-hand sides of (16) and observing that $\mathbf{L}\mathbf{x}^* = 0$, we obtain

$$\boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}^* = \boldsymbol{\lambda}_k - \boldsymbol{\lambda}^* + \alpha \mathbf{L}(\mathbf{x}_k - \mathbf{x}^*). \tag{20}$$

Defining $\mathbf{z}_{1,k} \triangleq \mathbf{x}_k - \mathbf{x}^*$ and $\mathbf{z}_{2,k} \triangleq \boldsymbol{\lambda}_k - \boldsymbol{\lambda}^*$, equations (19) and (20) can compactly written as

$$\begin{pmatrix} \mathbf{z}_{1,k+1} \\ \mathbf{z}_{2,k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{z}_{1,k} \\ \mathbf{z}_{2,k} \end{pmatrix} - \alpha \begin{pmatrix} \nabla^2 \mathbf{F}(k) & \mathbf{L}' \\ -\mathbf{L} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{z}_{1,k} \\ \mathbf{z}_{2,k} \end{pmatrix}, \tag{21}$$

where $\nabla^2 \mathbf{F}(k)$ is the Hessian of $\mathbf{F}(\mathbf{x})$ computed at a point on a line between $\mathbf{x}^*$ and $\mathbf{x}_k$.

We note that the set of fixed points of iteration (21) is given by the set $\{(\mathbf{0}', \mathbf{v}')' \mid \mathbf{v} \in \text{Null}(\mathbf{L}')\}$, where $\text{Null}(\mathbf{L}') \triangleq \{\eta \otimes \lambda \mid \lambda \in \mathbb{R}^n\}$. Therefore if convergence is achieved, $\mathbf{x}_k$ will converge to $\mathbf{x}^*$ and $\boldsymbol{\lambda}_k$ will converge to some point in the set $\boldsymbol{\lambda}^* + \text{Null}(\mathbf{L}')$.

In the following, we are going the reformulate iteration (21) so that the vector $\mathbf{0}$ becomes the fixed point. Let us define $\tilde{\mathbf{z}}_{2,k} \triangleq (\mathbf{I} - \mathbf{J})\mathbf{z}_{2,k}$. Since $\mathbf{J}\mathbf{z}_{2,k}$ is the projection of $\mathbf{z}_{2,k}$ on $\text{Null}(\mathbf{L}')$, the vector $\tilde{\mathbf{z}}_{2,k}$ is the error between $\mathbf{z}_{2,k}$ and its projection on $\text{Null}(\mathbf{L}')$. As a consequence, in terms of $\tilde{\mathbf{z}}_{2,k}$, iteration (21) becomes

$$\begin{pmatrix} \mathbf{z}_{1,k+1} \\ \tilde{\mathbf{z}}_{2,k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{z}_{1,k} \\ \tilde{\mathbf{z}}_{2,k} \end{pmatrix} - \alpha \begin{pmatrix} \nabla^2 \mathbf{F}(k) & \mathbf{L}' \\ -\mathbf{L} & \frac{1}{\alpha}\mathbf{J} \end{pmatrix} \begin{pmatrix} \mathbf{z}_{1,k} \\ \tilde{\mathbf{z}}_{2,k} \end{pmatrix}, \tag{22}$$

where we used the fact that $(\mathbf{I} - \mathbf{J})\tilde{\mathbf{z}}_{2,k} = (\mathbf{I} - \mathbf{J})\mathbf{L}\mathbf{z}_{2,k}$ and $\mathbf{L}'\mathbf{J} = \mathbf{0}$. Therefore if the vector $\mathbf{0}$ is an attractor for (22), the set $\{(\mathbf{0}', \mathbf{v}')' \mid \mathbf{v} \in \text{Null}(\mathbf{L}')\}$ is an attractor for the iteration (21). By Proposition 4.3, the eigenvalues of the matrix

$$\mathbf{B}(\mathbf{x}^*) = \begin{pmatrix} \nabla^2 \mathbf{F}(\mathbf{x}^*) & \mathbf{L}' \\ -\mathbf{L} & \frac{1}{\alpha}\mathbf{J} \end{pmatrix}$$

have positive real parts and in addition there are a number of n eigenvalues equal to $1/\alpha$, which correspond to the eigenspace $\{(\mathbf{0}', \mathbf{v}')' \mid \mathbf{v} \in \text{Null}(\mathbf{L}')\}$. By Proposition 4.4 we can find a positive scalar θ so that $\nabla^2 \mathbf{F}(\mathbf{x}) \succ 0$ for all $\mathbf{x} \in B_c(\mathbf{x}^*, \theta)$. Using again Proposition 4.3, we can infer that the matrix $\mathbf{B}(\mathbf{x})$ has all its eigenvalues with positive real parts for all $\mathbf{x} \in B_c(\mathbf{x}^*, \theta)$, and consequently there exists a positive scalar $\bar{\alpha} > 0$ so that the eigenvalues of the matrix

$$\boldsymbol{\Theta}(\mathbf{x}) \triangleq \mathbf{I} - \alpha \mathbf{B}(\mathbf{x}), \ \alpha \in (0, \bar{\alpha}]$$

are within the unit circle. In addition, the matrix $\boldsymbol{\Theta}(\mathbf{x})$ has a number of n eigenvalues equal to zero corresponding to the eigenspace $\{(\mathbf{0}', \mathbf{v}')' \mid \mathbf{v} \in \text{Null}(\mathbf{L}')\}$.

Therefore, there exists a matrix induced norm and $0 < \gamma < 1$ so that $\|\boldsymbol{\Theta}(\mathbf{x})\| \leq \gamma$ for all $\mathbf{x} \in B_c(\mathbf{x}^*, \theta)$. Consequently, $\boldsymbol{\Theta}$ is a contraction map and from the contraction map theorem (see for example Chapter 7 of [4]) $\mathbf{z}_{1,k}$ and $\tilde{\mathbf{z}}_{2,k}$ converge to zero and the rate of convergence is linear, for all $\mathbf{z}_{1,0} \in B_c(0, \theta)$. ■

Remark 5.1: The above theorem shows that the algorithm (A_1) converges to a local minimizer provided that the initial value $\mathbf{x}_0$ is sufficiently close to the local minimizer and the step-size α is sufficiently small. However, the algorithm cannot guarantee convergence to the unique Lagrange multiplier vector $\boldsymbol{\lambda}^* \in \text{Range}(\mathbf{L})$. It is not difficult to show that if the sequence $\{\boldsymbol{\lambda}_k\}$ converges then $\lim_{k\to\infty} \boldsymbol{\lambda}_k = \boldsymbol{\lambda}^* + \mathbf{J}\boldsymbol{\lambda}_0$, where the last term is the projection of $\boldsymbol{\lambda}_0$ on $\text{Null}(\mathbf{L}')$.

The following corollary reformulates Theorem 5.1 so that it can be applied to problem (P_1) directly.

Corollary 5.1: Let Assumptions 2.1 hold, let x^* be a local minima of (P_1) and assume that $\nabla^2 f_i(x^*) \succ 0$ for all $i = 1, \ldots, N$. Then there exist positive integers $\theta_i > 0$ for $i = 1, \ldots, N$, and $\bar{\alpha} > 0$ so that under algorithm (A_1)

$$\lim_{k\to\infty} \|x_{i,k} - x^*\| = 0 \ \forall i \tag{23}$$

for all $x_{i,0} \in B_c(x^*, \theta_i)$ and $\alpha \in (0, \bar{\alpha}]$, and in addition the rate of convergence is linear.

Proof: Since $\nabla^2 f_i(x^*) \succ 0$ we have that $\nabla^2 \mathbf{F}(\mathbf{x}^*) = \text{diag}\left(\nabla^2 f_i(x^*), i = 1, \ldots, N\right) \succ 0$ and the result follows from Theorem 5.1, where θ_i can be chosen as $\theta_i = \frac{1}{\sqrt{N}}\theta$, with θ being given by Theorem 5.1. ■

The above corollary shows that provided the agents' initial values $x_{i,0}$ are close enough to a local minimizer x^* and the step-size α is sufficiently small, agents executing algorithm (A_1) solves problem (P_1) in a distributed manner.

For strongly convex functions, we can formulate immediately the following corollary.

Corollary 5.2: Let Assumptions 2.1 hold, let $f_i(x)$, $i = 1,\ldots,N$ be strongly convex functions, and let x^* be the unique global minimizer of (P_1). Then there exits $\bar{\alpha} > 0$ so that for all $\alpha \in (0,\bar{\alpha}]$, the sequence $\{x_{i,k}\}$ generated by Algorithm (A_1) converges linearly to x^*, that is

$$\lim_{k\to\infty} \|x_{i,k} - x^*\| = 0 \ \forall i. \tag{24}$$

Proof: Since the functions $f_i(x)$ are strongly convex, there exist positive scalars m_i so that $\nabla^2 f_i(x) \geq m_i I$, for all x and i. This implies that $\nabla^2 \mathbf{F}(\mathbf{x}) \geq \min_i\{m_i\}\mathbf{I}$ for all $\mathbf{x}$. Consequently the matrix

$$\begin{pmatrix} \nabla^2\mathbf{F}(k) & \mathbf{L}' \\ -\mathbf{L} & \frac{1}{\alpha}\mathbf{J} \end{pmatrix},$$

of equation (22) has its eigenvalues with positive real parts for all k. In addition the positive real parts are uniformly bounded away from zero, since $\nabla^2\mathbf{F}(\mathbf{x}) \geq \min_i\{m_i\}\mathbf{I}$ for all $\mathbf{x}$. The result follows by mimicking the steps of Theorem 5.1, where θ can be chosen arbitrarily large. ■

VI. Comments on the convergence rate of algorithm (A_1)

In the previous section we established that under suitable conditions, Algorithm (A_1) can solve problem (P_1) in a distributed manner, and the convergence rate of the algorithm is linear. In this section we take a closer look at the rate of convergence, and in particular we would like to connect the convergence rate with the parameters of the problem and in particular with the connectivity of the communication network.

Algorithm (A_1) can be regarded as a first order approximation of the following continuous-time, linear dynamics:

$$\dot{\mathbf{x}} = -\nabla\mathbf{F}(t) - \mathbf{L}'\boldsymbol{\lambda}, \tag{25}$$

$$\dot{\boldsymbol{\lambda}} = \mathbf{L}\mathbf{x}, \tag{26}$$

where $\nabla\mathbf{F}(t)$ is the gradient of $\mathbf{F}(\mathbf{x})$ computed at $\mathbf{x}(t)$. In fact the differential equations (25) and (26) are a continuous-time, distributed algorithm for solving the problem (P_1). Similarly, the discrete-time dynamics (21) is the first order approximation of the continuous-time dynamics

$$\begin{pmatrix} \dot{\mathbf{z}}_1 \\ \dot{\mathbf{z}}_2 \end{pmatrix} = -\begin{pmatrix} \nabla^2\mathbf{F}(t) & \mathbf{L}' \\ -\mathbf{L} & 0 \end{pmatrix}\begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{pmatrix}, \tag{27}$$

where we recall that $\mathbf{z}_1(t) = \mathbf{x}(t) - \mathbf{x}^*$ and $\mathbf{z}_2(t) = \boldsymbol{\lambda}(t) - \boldsymbol{\lambda}^*$. Therefore, the rate of convergence of Algorithm (A_1) is dictated by the spectral properties of the matrix

$$\mathbf{M}(t) = \begin{pmatrix} \mathbf{H}(t) & \mathbf{L}' \\ -\mathbf{L} & \mathbf{0} \end{pmatrix}, \tag{28}$$

where $\mathbf{H}(t) \triangleq \nabla^2\mathbf{F}(t)$ is a positive definite matrix with eigenvalues lower and upper bounded by two positive numbers, m_l and m_u, respectively, along the trajectory of $\mathbf{x}(t)$, as ensured by Proposition 4.4. In the previous section we proved the convergence of the Algorithm (A_1) by studying the spectral properties of the matrix

$$\bar{\mathbf{M}}(t) = \begin{pmatrix} \mathbf{H}(t) & \mathbf{L}' \\ -\mathbf{L} & \frac{1}{\alpha}\mathbf{J} \end{pmatrix}. \tag{29}$$

As shown in the next proposition, with the exception of the eigenvalues corresponding to the eigenspace $\{\mathbf{x} \mid \mathbf{x}' = (\mathbf{0}', \mathbf{v}'), \mathbf{v} \in \text{Null}(\mathbf{L}')\}$, the eigenvalues of $\mathbf{M}(t)$ and $\bar{\mathbf{M}}(t)$ are the same. That is, it is enough to focus only on the non-zero eigenvalues of $\mathbf{M}(t)$. For notational simplicity, in the following proposition, the time dependence is omitted.

Proposition 6.1: Let $\beta \neq \frac{1}{\alpha}$ be an eigenvalue of $\bar{\mathbf{M}}$. Then β is an eigenvalue of $\mathbf{M}$ as well and $\mathbf{M}$ has n zero eigenvalues corresponding to the eigenspace $\{(\mathbf{0}', \mathbf{v}')' \mid \mathbf{v} \in \text{Null}(\mathbf{L}') = \eta \otimes v, v \in \mathbb{R}^n\}$.

Proof: Let $\beta \neq \frac{1}{\alpha}$ be an eigenvalue of $\bar{\mathbf{M}}$ with corresponding eigenvector $(\mathbf{u}', \mathbf{v}')'$, that is

$$\begin{pmatrix} \mathbf{H} & \mathbf{L}' \\ -\mathbf{L} & \frac{1}{\alpha}\mathbf{J} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} = \beta \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}. \tag{30}$$

In the proof of Proposition 4.3 we showed that $\beta \neq \frac{1}{\alpha}$ if and only if $\mathbf{v} \notin \text{Null}(\mathbf{L}')$. But this means that there must exist $\mathbf{v}_1 \in \text{Range}(\mathbf{L})$ and $\mathbf{v}_2 \in \text{Null}(\mathbf{L}')$ so that $\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2$ and $\mathbf{v}_1 \neq 0$. From (30) we get

$$\mathbf{H}\mathbf{u} + \mathbf{L}'\mathbf{v} = \mathbf{H}\mathbf{u} + \mathbf{L}'\mathbf{v}_1 = \beta\mathbf{u}. \tag{31}$$

Recalling the fact that $\mathbf{J}$ is the orthogonal projection operator on $\text{Null}(\mathbf{L}')$, again from (30) we obtain

$$-\mathbf{L}\mathbf{u} + \frac{1}{\alpha}\mathbf{v_2} = \beta\mathbf{v}_1 + \beta\mathbf{v}_2.$$

Since $-\mathbf{L}\mathbf{u} \in \text{Range}(\mathbf{L})$ and $\mathbf{v}_2 \in \text{Null}(\mathbf{L}')$ and any vector in $\mathbb{R}^{nN}$ admits unique projections on $\text{Range}(\mathbf{L})$ and $\text{Null}(\mathbf{L}')$, the following must be true:

$$-\mathbf{L}\mathbf{u} = \beta\mathbf{v_1}, \tag{32}$$

$$\frac{1}{\alpha}\mathbf{v_2} = \beta\mathbf{v}_2. \tag{33}$$

But since $\beta \neq \frac{1}{\alpha}$ it follows that $\mathbf{v}_2 = 0$. Using (31) and (32), we obtain that β is an eigenvalue of $\mathbf{M}$ with the corresponding eigenvector $(\mathbf{u}', \mathbf{v}_1')'$.

Proceeding in a similar way as in the proof of Proposition 4.3, we obtain that

$$\text{Re}(\beta) > 0, \ \forall \ \mathbf{u} \neq 0.$$

For $\mathbf{u} = 0$ we get

$$\mathbf{L}'\mathbf{v} = 0, \text{ and } 0 = \beta\mathbf{v},$$

which basically says that $\beta = 0$ is an eigenvalue corresponding to the eigenspace

$$\{\mathbf{x} \mid \mathbf{x}' = (0', \mathbf{v}'), \mathbf{v} \in \text{Null}(\mathbf{L}')\} = \{\mathbf{x} \mid \mathbf{x}' = (0', \ x' \otimes \eta'), \ x \in \mathbb{R}^n\},$$

which concludes the proof. ■

To simplify the analysis of the eigenvalues of $\mathbf{M}(t)$, we apply to it the similarity transformation

$$\mathbf{Q}(t) = \begin{pmatrix} \mathbf{H}(t) & \mathbf{L}' \\ -\mathbf{I} & \bar{\mathbf{J}} \end{pmatrix}, \tag{34}$$

where $\bar{\mathbf{J}} = \left(\frac{1}{N}\mathbb{1}\mathbb{1}'\right) \otimes I$, with $\mathbb{1}$ the N-dimensional vector of all ones and I the n-dimensional identity matrix, and obtain

$$\tilde{\mathbf{M}}(t) = \begin{pmatrix} \mathbf{H}(t) & \mathbf{L}'\mathbf{L} \\ -\mathbf{I} & \mathbf{0} \end{pmatrix}. \tag{35}$$

It can be checked that $\mathbf{Q}(t)$ is invertible and therefore can be indeed used as a similarity transformation. Consequently, $\tilde{\mathbf{M}}(t)$ has the same eigenvalues as $\mathbf{M}(t)$ and therefore it suffice to focus on its non-zero eigenvalues. In addition, it is not difficult to check that the eigenspace corresponding to the zero eigenvalues is given by $\{(\mathbf{0}', \mathbf{v}')' \mid \mathbf{v} \in \text{Null}(\mathbf{L})\}$.

Let β be a non-zero eigenvalue of matrix $\tilde{\mathbf{M}}(t)$ and let $(\mathbf{u}', \mathbf{v}')'$ be the corresponding right eigenvector (in the following we are going to ignore the time dependence to simplify the notation). By (35), the eigenvector must satisfy the following equations:

$$\mathbf{H}\mathbf{u} + \mathbf{L}'\mathbf{L}\mathbf{v} = \beta\mathbf{u} \tag{36}$$

$$-\mathbf{u} = \beta\mathbf{v}. \tag{37}$$

Since $\mathbf{H}$ is positive definite and therefore invertible, we can safely multiply (37) from the left by $\mathbf{H}$ and obtain

$$\mathbf{Hu} = -\beta\mathbf{Hv},$$

from where (36) becomes

$$-\beta\mathbf{Hv} + \mathbf{L}'\mathbf{Lv} = -\beta^2\mathbf{v}.$$

By multiplying the above with the hermitian of $\mathbf{v}$, we finally get

$$\beta^2\hat{\mathbf{v}}'\mathbf{v} - \beta\hat{\mathbf{v}}'\mathbf{Hv} + \hat{\mathbf{v}}'\mathbf{L}'\mathbf{Lv} = 0,$$

or

$$\beta^2 - \omega_1\beta + \omega_0 = 0, \tag{38}$$

where

$$\omega_1 = \frac{\hat{\mathbf{v}}'\mathbf{Hv}}{\hat{\mathbf{v}}'\mathbf{v}} \text{ and } \omega_0 = \frac{\hat{\mathbf{v}}'\mathbf{L}'\mathbf{Lv}}{\hat{\mathbf{v}}'\mathbf{v}}.$$

Therefore, any non-zero eigenvalue β must satisfy

$$\beta = \frac{\omega_1 \pm \sqrt{\omega_1^2 - 4\omega_0}}{2},$$

which shows that the eigenvalue can be both real and complex.

From the properties of $\mathbf{H}$ and $\mathbf{L}'\mathbf{L}$ we get that

$$m_l \le \omega_1 \le m_u \text{ and } 0 \le \omega_0 \le \sigma_{N,L'L},$$

and therefore if $m_l^2 \ge 4\sigma_{N,L'L}$ all eigenvalues are real.

Let β be a non-zero, real eigenvalue. Noting that $\omega_1 \ge \sqrt{\omega_1^2 - 4\omega_0}$ we have that, as expected, all real eigenvalues are positive. In the following, our goal is to determine lower and upper bounds on the real eigenvalues of $\tilde{\mathbf{M}}(t)$. We approach this task by considering worst-case scenarios. Considering the previous upper-bounds, we note that considering ω_0 fixed, the following inequality holds

$$\beta_{\min}(\omega_0) \le \beta \le \beta_{\max}(\omega_0)$$

where

$$\begin{aligned} \beta_{\min}(\omega_0) &= \frac{1}{2}\left(m_u - \sqrt{m_u^2 - 4\omega_0}\right), & (39) \\ \beta_{\max}(\omega_0) &= \frac{1}{2}\left(m_u + \sqrt{m_u^2 - 4\omega_0}\right). & (40) \end{aligned}$$

Next, we note that $\beta_{\min}(\omega_0)$ is minimized when ω_0 takes the smallest possible value. Similarly, $\beta_{\max}(\omega_0)$ is maximized when ω_0 takes the smallest possible value. We pointed out earlier that ω_0 is lower bounded by zero. Using the definition of ω_0, it follows that $\omega_0 = 0$ only if $\mathbf{v} \in \text{Null}(\mathbf{L})$, but this is impossible since we assumed β to be a non-zero eigenvalues. Therefore, ω_0 must be a non-zero, positive scalar.

Since $\mathbf{L}'\mathbf{L}$ is a symmetric matrix, there exists a matrix $\mathbf{U}$ whose columns are orthogonal eigenvectors of $\mathbf{L}'\mathbf{L}$, so that

$$\mathbf{L}'\mathbf{L} = \mathbf{U}'\mathbf{\Lambda}\mathbf{U},$$

where $\mathbf{\Lambda} = \Lambda \otimes I$, and

$$\Lambda = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & \sigma_{2,L'L} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \sigma_{N,L'L} \end{pmatrix},$$

with $\sigma_{i,L'L} > 0$ for all $i \geq 2$. Using the notation $\bar{\mathbf{v}} = \mathbf{U}\mathbf{v}$, ω_0 can be equivalently expressed as

$$\omega_0 = \frac{\sum_{i=2}^{N} \sigma_{i,L'L} \|\bar{v}_i\|^2}{\|\bar{\mathbf{v}}\|^2},$$

where the n-dimensional vector $\bar{v}_i$ is the i^{th} component of $\bar{\mathbf{v}}$. A lower-bound on ω_0 is given by

$$\omega_0 \geq \frac{\sigma_{2,L'L} \sum_{i=2}^{N} \|\bar{v}_i\|^2}{\|\bar{\mathbf{v}}\|^2} = \sigma_{2,L'L}(1-\epsilon),$$

where $\epsilon = \|\bar{v}_1\|^2 / \|\bar{\mathbf{v}}\|^2$. Note that $\epsilon \neq 1$ since this would mean that $\beta = 0$, contradicting the initial assumption. In the expression of ω_0 we notice how the connectivity of the graph, expressed in terms of the second smallest eigenvalue of $L'L$, influence the real eigenvalues of the matrix $\mathbf{M}(t)$ and consequently the rate of convergence. We note that the better the connectivity of the graph $\mathcal{G}$, the larger the value of σ_2^L is and consequently the larger the value of $\sigma_{2,L'L}$, as shown in the next proposition. Therefore, the better the connectivity is, the larger the lower bound of β is.

Proposition 6.2: Let A be a matrix so that $AA' = A'A$. If (λ, a) is an eigenvalue, (right) eigenvector pair of A then $\left(\lambda^2, a\right)$ is an eigenvalue, eigenvector pair of $A'A$.

Proof: Let a be the right eigenvector of A corresponding to the eigenvalue λ, that is,

$$Aa = \lambda a.$$

We also have that

$$A'Aa = \lambda A'a = AA'a,$$

from where we have that $A'a$ is an eigenvector of A, corresponding to the eigenvalue λ. But this means that $A'a$ and a must be co-linear, and therefore there exists γ so that $\gamma a = A'a$, or $\gamma \|a\|^2 = a'A'a = \lambda \|a\|^2$. Therefore $\gamma = \lambda$. Consequently $AA'a = \lambda A'a = \lambda^2 a = A'Aa$. Therefore, λ^2 is an eigenvalue of $A'A$ with a the corresponding eigenvector. ■

We saw earlier that if $\omega_1^2 - 4\omega_0 < 0$, the eigenvalue β is complex, that is,

$$\beta_{1,2} = \frac{\omega_1 \pm j\sqrt{4\omega_0 - \omega_1^2}}{2},$$

and we can immediately determine that

$$\frac{m_l}{2} \leq \text{Re}(\beta) \leq \frac{m_u}{2}. \tag{41}$$

In addition the absolute value of β is given by $|\beta| = 4\omega_0$ and therefore

$$4(1-\epsilon)\sigma_{2,L'L} \leq |\beta| \leq 4\sigma_{N,L'L}. \tag{42}$$

We note from above that in the case of a complex eigenvalue, its real part is not dependent on the network. Hence in the continuous-time version of the algorithm, (part) of the state vector may convergence faster, but it will oscillate due to the complex part and the amplitude of oscillations depends on the network structure through ω_0. Note however, that in the discrete version of the algorithm, the absolute values of the eigenvalues determine the rate of converge; absolute values which do contain the complex part.

When the discrete-time version of the dynamics of the estimates is used, the eigenvalues of interest are of the form $1 - \alpha\beta$ and we are interested in values of α so that $|1 - \alpha\beta| < 1$.

In the case of real eigenvalues, from (39) we have that

$$0 < \alpha < \frac{2}{\beta} \leq \frac{4}{m_u + \sqrt{m_u^2 - 4(1-\epsilon)\sigma_{2,L'L}}}, \tag{43}$$

which shows that a better connectivity allows for a larger interval from which α can be chosen to ensure the stability of the algorithm.

In the case of complex eigenvalues we have

$$|1-\alpha\beta|^2 = [1-\alpha \mathrm{Re}(\beta)]^2 + \alpha^2 \mathrm{Im}(\beta)^2 = 1 - 2\alpha \mathrm{Re}(\beta) + \alpha^2|\beta|^2.$$

From (41) and (42) it follows that

$$1 - 2\alpha \mathrm{Re}(\beta) \leq 1 - \alpha m_l,$$

and

$$|\beta|^2 \leq [4\sigma_{N,L'L}]^2.$$

Consequently

$$|1-\alpha\beta|^2 \leq 1 - \alpha m_l + \alpha^2 [4\sigma_{N,L'L}]^2,$$

and a sufficient condition so that the complex eigenvalues are strictly within the unit circle is:

$$1 - \alpha m_l + \alpha^2 [4\sigma_{N,L'L}]^2 < 1, \tag{44}$$

from where we have that

$$\alpha < \frac{m_l}{[4\sigma_{N,L'L}]^2}. \tag{45}$$

Inequalities (43) and (45) give in fact an estimate for the parameter $\bar{\alpha}$ whose existence is shown in Theorem 5.1, namely

$$\bar{\alpha} = \min\left\{\frac{4}{m_u + \sqrt{m_u^2 - 4(1-\epsilon)\sigma_{2,L'L}}}, \frac{m_l}{[4\sigma_{N,L'L}]^2}\right\}.$$

We end this section with two observations. First, we can observe that an improved connectivity (reflected by a large value of $\sigma_{2,L}$, and consequently a large value of $\sigma_{2,L'L}$) may potentially increase the rate of convergence, since α may be chosen larger. Note however that there are limitations on the benefits of an improved connectivity. More specifically, a large value of $\sigma_{2,L}$ may make $4/\left(m_u + \sqrt{m_u^2 - 4(1-\epsilon)\sigma_{2,L'L}}\right)$ larger than $m_l/[4\sigma_{N,L'L}]^2$, at which point the latter becomes the value for the estimate of $\bar{\alpha}$. Therefore, improving connectivity after some threshold may not help the rate of convergence. The second observation is that the estimate for $\bar{\alpha}$ is rather theoretical, since to actually compute this estimate we would need to know the values of m_l, m_u and ϵ (so far we only showed there existence). Still, the analysis gives an interesting insight on the tradeoffs between the connectivity of the communication graph and the rate of convergence of the algorithm.

VII. Consensus-based distributed optimization algorithms revisited

In this section we revisit two of the most studied consensus-based distributed algorithms in the literature. We analyze them using the setup introduced in the previous sections and give additional conditions so that they can be applied for differentiable functions, not necessarily convex. We give estimates on their the rate of convergence and show that, except in some very favorable cases, unlike algorithm (A_1), the studied consensus-based algorithms do not converge to a local minimizer, but rather to a neighborhood around the minimizer; neighborhood whose diameter depends on the step-size of the algorithm. The analysis of the two algorithms is made for constant step-size.

The first consensus-based distributed algorithm, named henceforth Algorithm (A_2), was proposed in [11] and is given by

$$x_{i,k+1} = \sum_{j\in\mathcal{N}_i\cup i} a_{ij} x_{j,k} - \alpha \nabla f_i(x_{i,k}), \tag{46}$$

where $A = \left(a_{ij}\right)$ is a stochastic matrix, corresponding to the communication graph $\mathcal{G}$, and assumed *symmetric*. Using our formulation, algorithm (A_2) can be reformulated as

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k - \alpha\nabla\mathbf{F}(\mathbf{x}_k), \tag{47}$$

where $\mathbf{A} \triangleq A \otimes I$.

The second consensus-based distributed optimization algorithm, referred to henceforth as Algorithm (A_3) and introduced in [5], [6], is expressed as

$$x_{i,k+1} = \sum_{j\in\mathcal{N}_i\cup i} a_{ij}\left[x_{j,k} - \alpha\nabla f_j(x_{j,k})\right], \tag{48}$$

or equivalently

$$\mathbf{x}_{k+1} = \mathbf{A}\left[\mathbf{x}_k - \alpha\nabla\mathbf{F}(\mathbf{x}_k)\right], \tag{49}$$

where $\mathbf{A} \triangleq A \otimes I$. Note that compared to Algorithm (A_2), in Algorithm (A_3) matrix $\mathbf{A}$ multiplies the gradient $\nabla\mathbf{F}(\mathbf{x}_k)$ as well, suggesting a change in the order the two operations performed by the agents are executed: first the agents update their current estimates by advancing in the direction provided by the gradient, followed by sharing these updates with their neighbors, and performing a convex combination with them. Throughout the rest of this section, $\|\cdot\|$ refers to the Euclidean norm.

A. *Convergence analysis of Algorithm* (A_2)

In this subsection we give new (sufficient) conditions so that Algorithm (A_2) can be applied for twice differentiable, not necessarily convex cost functions. In addition, we show that except in some very favorable conditions, for constant step-size, Algorithm (A_2) does not converge to a local minimizer, even though the agents start with initial values close to the minimizer and use a small enough step-size so that the algorithm converges.

Let $\mathbf{x}^*$ be a local minimizer of (P_2). Then the gradient of $\mathbf{F}(\mathbf{x})$ can be expressed as

$$\nabla\mathbf{F}(\mathbf{x}) = \nabla\mathbf{F}(\mathbf{x}^*) + \nabla^2\mathbf{F}(\mathbf{y})(\mathbf{x}-\mathbf{x}^*),$$

where $\nabla^2\mathbf{F}(\mathbf{y})$ is the Hessian of $\mathbf{F}(\mathbf{x})$ computed at $\mathbf{y}$, a point between $\mathbf{x}$ and $\mathbf{x}^*$. In this case algorithm (A_2) becomes

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k - \alpha\left[\nabla\mathbf{F}(\mathbf{x}^*) + \nabla^2\mathbf{F}(k)(\mathbf{x}_k - \mathbf{x}^*)\right], \tag{50}$$

or subtracting $\mathbf{x}^*$ from the left and right hand sides of (50), we obtain

$$\mathbf{x}_{k+1} - \mathbf{x}^* = \left[\mathbf{A} - \alpha\nabla^2\mathbf{F}(k)\right]\left[\mathbf{x}_k - \mathbf{x}^*\right] - \alpha\nabla\mathbf{F}(\mathbf{x}^*), \tag{51}$$

where we used that fact that $\mathbf{A}\mathbf{x}^* = \mathbf{x}^*$. Defining $\mathbf{M}_2(k) \triangleq \mathbf{A} - \alpha\nabla^2\mathbf{F}(k)$, we further have

$$\mathbf{x}_{k+1} - \mathbf{x}^* = \mathbf{M}_2(k)\left[\mathbf{x}_k - \mathbf{x}^*\right] - \alpha\nabla\mathbf{F}(\mathbf{x}^*), \tag{52}$$

Let $\boldsymbol{\sigma}(\mathbf{A}) = \{-1 < \sigma_{1,\mathbf{A}} \leq \sigma_{2,\mathbf{A}} \leq, \ldots, \leq \sigma_{nN,\mathbf{A}} = 1\}$ be the set of eigenvalues of $\mathbf{A}$. The next proposition characterizes the spectral properties of the matrix $\mathbf{A} - \alpha\nabla^2\mathbf{F}(\mathbf{x})$ as a function of α.

Proposition 7.1: Let Assumptions 2.1 hold, let $\mathbf{x}^*$ be a local minimizer of (P_2) and assume that $\nabla^2\mathbf{F}(\mathbf{x}^*) \succ 0$. Then there exist positive numbers θ, m_l and m_u, with $m_l \leq m_u$ so that

$$\rho\left(\mathbf{A} - \alpha\nabla^2\mathbf{F}(\mathbf{x})\right) < 1,\ \forall \mathbf{x} \in B_c(\mathbf{x}^*, \theta),\ \forall \alpha \in \left(0, \frac{1+\sigma_{1,\mathbf{A}}}{m_u}\right), \tag{53}$$

and an upper bound on the spectral radius of $\mathbf{A} - \alpha\nabla^2\mathbf{F}(\mathbf{x})$ is given by

$$\rho\left(\mathbf{A} - \alpha\nabla\mathbf{F}(\mathbf{x})\right) \leq \begin{cases} 1 - \alpha m_l & \alpha \in \left(0, \frac{1+\sigma_{1,\mathbf{A}}}{m_l+m_u}\right), \\ -\sigma_{1,\mathbf{A}} + \alpha m_u & \alpha \in \left[\frac{1+\sigma_{1,\mathbf{A}}}{m_l+m_u}, \frac{1+\sigma_{1,\mathbf{A}}}{m_u}\right). \end{cases} \tag{54}$$

Proof: By Proposition 4.4, there exist the scalars θ, m_l and m_u so that

$$\nabla^2\mathbf{F}(\mathbf{x}) \succ 0,\ \forall \mathbf{x} \in B_c(\mathbf{x}^*,\theta)$$

and

$$m_l\mathbf{I} \preceq \nabla^2\mathbf{F}(\mathbf{x}) \preceq m_u\mathbf{I},\ \forall \mathbf{x} \in B_c(\mathbf{x}^*,\theta).$$

Since A is symmetric, we have that

$$\sigma_{1,\mathbf{A}}\mathbf{I} \preceq \mathbf{A} \preceq \mathbf{I},$$

and it follows that

$$(\sigma_{1,\mathbf{A}} - \alpha m_u)\mathbf{I} \prec \mathbf{A} - \alpha\nabla^2\mathbf{F}(\mathbf{x}) \preceq (1-\alpha m_l)\mathbf{I},\ \forall \mathbf{x} \in B_c(\mathbf{x}^*,\theta).$$

Therefore, the matrix $\mathbf{A} - \alpha\nabla^2\mathbf{F}(\mathbf{x})$ has all its eigenvalues strictly inside the unit circle if the following inequalities hold:

$$|\sigma_{1,\mathbf{A}} - \alpha m_u| < 1,\ |1-\alpha m_l| < 1.$$

From the first inequality we obtain that $\alpha \in \left(0, \frac{\sigma_{1,\mathbf{A}}+1}{m_u}\right)$ while from the second inequality we get that $\alpha \in \left(0, \frac{2}{m_l}\right)$. Noticing that $1+\sigma_{1,\mathbf{A}} < 2$ we get that

$$\alpha \in \left(0, \frac{\sigma_{1,\mathbf{A}}+1}{m_u}\right).$$

To derive (54) we consider three cases on the positions of $\sigma_{1,\mathbf{A}} - \alpha m_u$ and $1-\alpha m_l$ with respect to zero, within the unit circle.

Assume first that $0 < \sigma_{1,\mathbf{A}} - \alpha m_u < 1-\alpha m_l$ or equivalently $0 < \alpha < \frac{\sigma_{1,\mathbf{A}}}{m_u}$. In this case the spectral radius of $\mathbf{A} - \alpha\nabla^2\mathbf{F}(\mathbf{x})$ is upper bounded by $1-\alpha m_l$, and therefore we have

$$\rho\left(\mathbf{A} - \alpha\nabla^2\mathbf{F}(\mathbf{x})\right) \leq 1-\alpha m_l,\ \forall \alpha \in \left(0, \frac{\sigma_{1,\mathbf{A}}}{m_u}\right). \tag{55}$$

In the second case we assume that $\sigma_{1,\mathbf{A}} - \alpha m_u \leq 0 < 1-\alpha m_l$ which is true for $\alpha \in \left[\frac{\sigma_{1,\mathbf{A}}}{m_u}, \frac{1}{m_l}\right)$. Comparing the modulus of the two bounds we obtain

$$\rho\left(\mathbf{A} - \alpha\nabla^2\mathbf{F}(\mathbf{x})\right) \leq \begin{cases} 1-\alpha m_l & \alpha \in \left(\frac{\sigma_{1,\mathbf{A}}}{m_u}, \frac{1+\sigma_{1,\mathbf{A}}}{m_l+m_u}\right), \\ -\sigma_{1,\mathbf{A}} + \alpha m_u & \alpha \in \left[\frac{1+\sigma_{1,\mathbf{A}}}{m_l+m_u}, \frac{1}{m_l}\right). \end{cases} \tag{56}$$

In the third case we have $\sigma_{1,\mathbf{A}} - \alpha m_u < 1-\alpha m_l \leq 0$ form where we obtain that

$$\rho\left(\mathbf{A} - \alpha\nabla^2\mathbf{F}(\mathbf{x})\right) \leq -\sigma_{1,\mathbf{A}} + \alpha m_u,\ \forall \alpha \in \left[\frac{1}{m_l}, \frac{1+\sigma_{1,\mathbf{A}}}{m_u}\right). \tag{57}$$

The result follows from the combination of (55),(56) and (57). ■

The next result specifies sufficient conditions so that $\mathbf{x}_k - \mathbf{x}^*$ evolving according to (52) remains bounded.

Proposition 7.2: Let Assumptions 2.1 hold, let $\mathbf{x}^*$ be a local minimizer of (P_2).Assume also that $\nabla^2\mathbf{F}(\mathbf{x}^*) \succ 0$. Then there exist positive numbers θ, m_l and m_u (with $m_l \leq m_u$), $\bar{\alpha} = \frac{(1+\sigma_{1,\mathbf{A}})\theta}{m_u\theta + \|\nabla\mathbf{F}(\mathbf{x}^*)\|}$ and $0 < \delta < 1$ so that if $\mathbf{x}_0 \in B_c(\mathbf{x}^*,\theta)$, $\alpha \in (0,\bar{\alpha}]$ and

$$\|\nabla\mathbf{F}(\mathbf{x}^*)\| \leq \theta m_l, \tag{58}$$

then the sequence $\{\mathbf{x}_k\}$ generated by Algorithm (A_2) satisfies

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \quad \leq \quad \delta^k\|\mathbf{x}_0 - \mathbf{x}^*\| + \alpha\frac{1}{1-\delta}\left\|\nabla\mathbf{F}(\mathbf{x}^*)\right\|. \tag{59}$$

In addition, there exists a matrix $\mathbf{\Psi}_2^*$ so that

$$\lim_{k\to\infty} \mathbf{x}_k - \mathbf{x}^* = -\alpha\mathbf{\Psi}_2^*\nabla\mathbf{F}(\mathbf{x}^*), \tag{60}$$

where

$$\mathbf{\Psi}_2^* \triangleq \lim_{k\to\infty} \sum_{\tau=0}^{k-1} \mathbf{\Phi}_2(k,\tau), \tag{61}$$

with $\mathbf{\Phi}_2(k,\tau)$ the transition matrix of (51), that is, $\mathbf{\Phi}_2(k,\tau) = \mathbf{M}_2(k-1)\mathbf{M}_2(k-2)\ldots\mathbf{M}_2(\tau)$.

Proof: The existence of the positive numbers θ, m_l and m_u is ensured by Proposition 4.4. By (58), we have that $\frac{1+\sigma_{1,\mathbf{A}}}{m_u+m_l} \leq \bar{\alpha} \leq \frac{1+\sigma_{1,\mathbf{A}}}{m_u}$. Then for any $\alpha \in (0,\bar{\alpha}]$, according to Proposition 7.1, the eigenvalues of $\mathbf{A}-\alpha\nabla^2\mathbf{F}(\mathbf{x})$ for all $\mathbf{x} \in B_c(\mathbf{x}^*,\theta)$ are strictly within the unit circle. By choosing $\delta = \max\{-\sigma_{1,\mathbf{A}}+\alpha m_u, 1-\alpha m_l\}$, again by Proposition 7.1, we have that

$$\rho\Big(\mathbf{A}-\alpha\nabla^2\mathbf{F}(\mathbf{x})\Big) \leq \delta, \forall \mathbf{x} \in B_c(\mathbf{x}^*,\theta),\ \forall \alpha \in (0,\bar{\alpha}].$$

The main idea of the proof consists of showing that $\mathbf{x}_k - \mathbf{x}^* \in B_c(0,\theta)$ for all time k. If this is the case then $\rho(\mathbf{M}_2(k)) \leq \delta$ for all k since the Hessian $\nabla^2\mathbf{F}(k)$ from $\mathbf{M}_2(k)$ is computed at a point on a line between $\mathbf{x}^*$ and $\mathbf{x}_k$, and therefore it belongs to $B_c(\mathbf{x}^*,\theta)$.

We proceed by induction. Assume that $||\mathbf{x}_k-\mathbf{x}^*|| \leq \theta$. Then according to iteration (51)

$$||\mathbf{x}_{k+1}-\mathbf{x}^*|| \leq \delta||\mathbf{x}_k-\mathbf{x}^*|| + \alpha||\nabla\mathbf{F}(\mathbf{x}^*)|| \leq \delta\theta + \alpha||\nabla\mathbf{F}(\mathbf{x}^*)||. \tag{62}$$

Depending on the value of α, we distinguish two cases. If $\alpha \in \left(0,\ \frac{1+\sigma_{1,\mathbf{A}}}{m_l+m_u}\right)$ then $\delta = 1-\alpha m_l$ and (62) becomes

$$||\mathbf{x}_{k+1}-\mathbf{x}^*|| \leq (1-\alpha m_l)\theta + \alpha||\nabla\mathbf{F}(\mathbf{x}^*)|| = \theta - \alpha\left(m_l\theta - ||\nabla\mathbf{F}(\mathbf{x}^*)||\right) \leq \theta,$$

where the last inequality followed from (58). In fact one can check that $\theta - \alpha\left(m_l\theta - ||\nabla\mathbf{F}(\mathbf{x}^*)||\right) > 0$ for $\alpha \in \left(0,\ \frac{1+\sigma_{1,\mathbf{A}}}{m_l+m_u}\right)$, and therefore the above inequality makes sense.

If however $\alpha \in \left[\frac{1+\sigma_{1,\mathbf{A}}}{m_l+m_u},\ \bar{\alpha}\right)$ then $\delta = -\sigma_{1,\mathbf{A}}+\alpha m_u$ and (62) becomes

$$||\mathbf{x}_{k+1}-\mathbf{x}^*|| \leq (-\sigma_{1,\mathbf{A}}+\alpha m_u)\theta + \alpha||\nabla\mathbf{F}(\mathbf{x}^*)|| =$$
$$-\sigma_{1,\mathbf{A}}\theta + \alpha\left(m_u\theta + ||\nabla\mathbf{F}(\mathbf{x}^*)||\right) \leq -\sigma_{1,\mathbf{A}}\theta + \bar{\alpha}\left(m_u\theta + ||\nabla\mathbf{F}(\mathbf{x}^*)||\right)$$
$$= -\sigma_{1,\mathbf{A}}\theta + \frac{(1+\sigma_{1,\mathbf{A}})\theta}{m_u\theta + ||\nabla\mathbf{F}(\mathbf{x}^*)||}\left(m_u\theta + ||\nabla\mathbf{F}(\mathbf{x}^*)||\right) = \theta.$$

Therefore, for our choice of $\bar{\alpha}$ we have that $||\mathbf{x}_k-\mathbf{x}^*|| \leq \theta$ for all $k \geq 0$ and consequently

$$\rho(\mathbf{M}_2(k)) = \rho\Big(\mathbf{A}-\alpha\nabla^2\mathbf{F}(k)\Big) \leq \delta < 1,\ \forall k \geq 0.$$

The solution of (51) is given by

$$\mathbf{x}_k - \mathbf{x}^* = \mathbf{\Phi}_2(k,0)\left(\mathbf{x}_0-\mathbf{x}^*\right) - \alpha\sum_{\tau=0}^{k-1}\mathbf{\Phi}_2(k,\tau)\nabla\mathbf{F}(\mathbf{x}^*),$$

and since $||\mathbf{\Phi}_2(k,\tau)|| \leq ||\mathbf{M}_2(k-1)||||\mathbf{M}_2(k-2)||\ldots||\mathbf{M}_2(\tau)|| = \delta^{k-\tau}$ we obtain that

$$\left\|\sum_{\tau=0}^{k-1}\mathbf{\Phi}_2(k,\tau)\nabla\mathbf{F}(\mathbf{x}^*)\right\| \leq \sum_{\tau=0}^{k-1}||\mathbf{\Phi}_2(k,\tau)||||\nabla\mathbf{F}(\mathbf{x}^*)|| \leq$$
$$\sum_{\tau=0}^{k-1}\delta^{k-\tau}||\nabla\mathbf{F}(\mathbf{x}^*)|| \leq \frac{1}{1-\delta}||\nabla\mathbf{F}(\mathbf{x}^*)||,$$

from where (59) follows.

To obtain (60) it is enough to show that the series $\mathbf{\Psi}_2(k) = \sum_{\tau=0}^{k-1}\mathbf{\Phi}_2(k,\tau)$ converges since we already have that

$$\lim_{k\to\infty}\mathbf{\Phi}_2(k,0)\left[\mathbf{x}_0-\mathbf{x}^*\right] = \mathbf{0}.$$

We achieve this by showing that the series $\sum_{\tau=0}^{k-1}\left|\mathbf{e}_i'\mathbf{\Phi}_2(k,\tau)\mathbf{e}_j\right|$ converges, since this implies that the series $\sum_{\tau=0}^{k-1}\mathbf{e}_i'\mathbf{\Phi}_2(k,\tau)\mathbf{e}_j$ converges as well, where $\{\mathbf{e}_i\}_{i=1}^{nN}$ represents the standard Euclidean basis.

Using the Cauchy-Schwarz inequality and the fact that $\|\mathbf{e}_i\| = 1$ we get

$$\sum_{\tau=0}^{k-1}\left|\mathbf{e}_i'\mathbf{\Phi}_2(k,\tau)\mathbf{e}_j\right| \leq \sum_{\tau=0}^{k-1}\|\mathbf{\Phi}_2(k,\tau)\| \leq \sum_{\tau=0}^{k-1}\delta^{k-\tau} = \frac{1-\delta^k}{1-\delta} \leq \frac{1}{1-\delta},$$

and therefore the series is upper bounded. But since is also monotonically increasing, then it converges and therefore there exits $\mathbf{\Psi}_2^*$ so that

$$\lim_{k\to\infty}\sum_{\tau=0}^{k-1}\mathbf{\Phi}_2(k,\tau) = \mathbf{\Psi}_2^*. \tag{63}$$

Consequently, we get that

$$\lim_{k\to\infty}\mathbf{x}_k - \mathbf{x}^* = \lim_{k\to\infty} -\alpha\sum_{\tau=0}^{k-1}\mathbf{\Phi}_2(k,\tau)\nabla\mathbf{F}(\mathbf{x}^*) = -\alpha\mathbf{\Psi}_2^*\nabla\mathbf{F}(\mathbf{x}^*),$$

which concludes the proof.

■

We showed above that the stability of Algorithm (A_2) can be guaranteed if the gradient $\nabla\mathbf{F}(\mathbf{x}^*)$ is less than a threshold; threshold that depends on a set of parameters induced by the behavior of $\nabla^2\mathbf{F}(\mathbf{x})$ around the minimizer $\mathbf{x}^*$. We need this inequality to hold to make sure that at each iteration $\mathbf{x}_k$ is kept in a neighborhood around $\mathbf{x}^*$ in which $\nabla^2\mathbf{F}(\mathbf{x})$ is positive definite. Such a condition is not needed in the case of Algorithm (A_1). We also showed that algorithm (A_2) does not guarantee convergence to a minimizer of (P_1) but rather to a neighborhood around x^*, whose size depends on $\nabla f_i(x^*)$ and α. The neighborhood can be made arbitrarily small by making α very small, but this would reduce the rate of convergence. In fact, except in the case of some “fluke of nature”, mathematically translated as

$$\nabla\mathbf{F}(\mathbf{x}^*) \in \text{Null}\left(\mathbf{\Psi}_2^*\right),$$

convergence to a local minimizer does not happen. In particular, convergence is achieved if $\nabla\mathbf{F}(\mathbf{x}^*) = 0$, or equivalently $f_i(x^*) = 0$ for all i. However, this is not an interesting case since there is no need for cooperation between agents.

B. *Convergence analysis of Algorithm* (A_3)

This section focuses on the convergence properties of Algorithm (A_3). We give conditions to ensure the stability of Algorithm (A_3) in terms of the parameters of the problem. As in the case of Algorithm A_2, we show that in general, for constant step-size, Algorithm (A_3) does not converge to a local minimizer, even though the agents' initial values are close to the minimizer and the algorithm is stable. Still, there are more possible scenarios under which Algorithm (A_3) does converge to a local minimizer, compared to Algorithm (A_2).

Let $\mathbf{x}^*$ be a local minimizer of (P_2). Then proceeding as in the case of Algorithm (A_2), we can reformulate Algorithm (A_3) in terms of the Hessian of $\mathbf{F}(\mathbf{x})$ and obtain

$$\mathbf{x}_{k+1} - \mathbf{x}^* = \mathbf{A}\left[\mathbf{I} - \alpha\nabla^2\mathbf{F}(k)\right][\mathbf{x}_k - \mathbf{x}^*] - \alpha\mathbf{A}\nabla\mathbf{F}(\mathbf{x}^*), \tag{64}$$

or by defining $\mathbf{M}_3(k) \triangleq \mathbf{A}\left[\mathbf{I} - \alpha\nabla^2\mathbf{F}(k)\right]$, we have

$$\mathbf{x}_{k+1} - \mathbf{x}^* = \mathbf{M}_3(k)[\mathbf{x}_k - \mathbf{x}^*] - \alpha\mathbf{A}\nabla\mathbf{F}(\mathbf{x}^*). \tag{65}$$

The next result is the equivalent of Proposition 7.1 and shows that as long $\mathbf{x}$ is kept close enough to a local minimizer $\mathbf{x}^*$, the matrix $\mathbf{A}\left[\mathbf{I} - \alpha\nabla^2\mathbf{F}(\mathbf{x})\right]$ is positive definite provided that $\nabla^2\mathbf{F}(\mathbf{x}^*)$ is positive definite.

Proposition 7.3: Let Assumptions 2.1 hold, let $\mathbf{x}^*$ be a local minimizer of (P_2) and assume that $\nabla^2\mathbf{F}(\mathbf{x}^*) \succ 0$. Then there exist positive numbers θ, m_l and m_u (with $m_{\leq}m_u$) so that

$$\rho\left(\mathbf{A}\left[\mathbf{I}-\alpha\nabla^2\mathbf{F}(\mathbf{x})\right]\right) < 1,\ \forall \mathbf{x}\in B_c(\mathbf{x}^*,\theta),\ \forall\alpha\in\left(0,\frac{2}{m_u}\right), \tag{66}$$

and an upper bound on the spectral radius of $\mathbf{A}\left[\mathbf{I}-\alpha\nabla^2\mathbf{F}(\mathbf{x})\right]$ is given by

$$\rho\left(\mathbf{A}\left[\mathbf{I}-\alpha\nabla^2\mathbf{F}(\mathbf{x})\right]\right) \leq \begin{cases} 1-\alpha m_l & \alpha\in\left(0,\ \frac{2}{m_l+m_u}\right), \\ -1+\alpha m_u & \alpha\in\left[\frac{2}{m_l+m_u},\ \frac{2}{m_u}\right). \end{cases} \tag{67}$$

Proof: By Proposition 4.4, there exist the scalars θ, m_l and m_u so that

$$\nabla^2\mathbf{F}(\mathbf{x}) \succ 0\ \forall\mathbf{x}\in B_c(\mathbf{x}^*,\theta)$$

and

$$m_l\mathbf{I} \preceq \nabla^2\mathbf{F}(\mathbf{x}) \preceq m_u\mathbf{I}\ \forall\mathbf{x}\in B_c(\mathbf{x}^*,\theta).$$

Since A is symmetric, we have that $\rho(\mathbf{A}) = \|\mathbf{A}\| = 1$ and therefore

$$\rho\left(\mathbf{A}\left[\mathbf{I}-\alpha\nabla^2\mathbf{F}(\mathbf{x})\right]\right) \leq \rho\left(\mathbf{I}-\alpha\nabla^2\mathbf{F}(\mathbf{x})\right).$$

We also have that

$$(1-\alpha m_u)\mathbf{I} \preceq \mathbf{I}-\alpha\nabla^2\mathbf{F}(\mathbf{x}) \preceq (1-\alpha m_l)\mathbf{I}\ \forall\mathbf{x}\in B_c(\mathbf{x}^*,\theta).$$

Therefore, in order for the matrix $\mathbf{A}\left[\mathbf{I}-\alpha\nabla^2\mathbf{F}(\mathbf{x})\right]$ to have all its eigenvalues strictly inside the unit circle, it is sufficient for the following inequalities to be satisfied:

$$|1-\alpha m_u| < 1,\ |1-\alpha m_l| < 1,$$

or equivalently

$$\alpha\in\left(0,\frac{2}{m_u}\right).$$

To derive (67) we consider three cases on the positions of $1-\alpha m_u$ and $1-\alpha m_l$ with respect to zero.

Let us first assume that $0 < 1-\alpha m_u < 1-\alpha m_l$ or equivalently $0 < \alpha < \frac{1}{m_u}$. In this case the spectral radius of $\mathbf{A}\left[\mathbf{I}-\alpha\nabla^2\mathbf{F}(\mathbf{x})\right]$ is upper bounded by $1-\alpha m_l$, and therefore we have

$$\rho\left(\mathbf{A}\left[\mathbf{I}-\alpha\nabla^2\mathbf{F}(\mathbf{x})\right]\right) \leq 1-\alpha m_l\ \forall\alpha\in\left(0,\ \frac{1}{m_u}\right). \tag{68}$$

In the second case we assume that $1-\alpha m_u \leq 0 < 1-\alpha m_l$ which is true for $\alpha\in\left[\frac{1}{m_u},\frac{1}{m_l}\right)$. Comparing the modulus of the two bounds we obtain

$$\rho\left(\mathbf{A}\left[\mathbf{I}-\alpha\nabla^2\mathbf{F}(\mathbf{x})\right]\right) \leq \begin{cases} 1-\alpha m_l & \alpha\in\left(\frac{1}{m_u},\ \frac{2}{m_l+m_u}\right), \\ -1+\alpha m_u & \alpha\in\left[\frac{2}{m_l+m_u},\ \frac{1}{m_l}\right). \end{cases} \tag{69}$$

In the third case we have $1-\alpha m_u < 1-\alpha m_l \leq 0$ form where we obtain that

$$\rho\left(\mathbf{A}\left[\mathbf{I}-\alpha\nabla^2\mathbf{F}(\mathbf{x})\right]\right) \leq -1+\alpha m_u\ \forall\alpha\in\left[\frac{1}{m_l},\ \frac{2}{m_u}\right). \tag{70}$$

The result follows from the combination of (68),(69) and (70). ■

The following result give conditions under which Algorithm (A_3) converges. It also shows that in general $\mathbf{x}_k$ does not converge to $\mathbf{x}^*$ except in some particular cases.

Proposition 7.4: Let Assumptions 2.1 hold, let $\mathbf{x}^*$ be a local minimizer of (P_2) and assume that $\nabla^2 \mathbf{F}(\mathbf{x}^*) \succ 0$. Then there exist positive numbers θ, m_l and m_u (with $m_l \leq m_u$), $\bar{\alpha} = \frac{2\theta}{m_u\theta + \|\nabla \mathbf{F}(\mathbf{x}^*)\|}$ and $0 < \delta < 1$ so that if $\mathbf{x}_0 \in B_c(\mathbf{x}^*, \theta)$, $\alpha \in (0, \bar{\alpha}]$ and

$$\|\nabla \mathbf{F}(\mathbf{x}^*)\| \leq \theta m_l, \tag{71}$$

then the sequence $\{\mathbf{x}_k\}$ generated by Algorithm (A_3) satisfies

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \quad \leq \quad \delta^k \|\mathbf{x}_0 - \mathbf{x}^*\| + \alpha \frac{1}{1-\delta} \left\| \mathbf{A} \nabla \mathbf{F}(\mathbf{x}^*) \right\|. \tag{72}$$

In addition, there exists a matrix $\mathbf{\Psi}_3^*$ so that

$$\lim_{k \to \infty} \mathbf{x}_k - \mathbf{x}^* = -\alpha \mathbf{\Psi}_3^* \mathbf{A} \nabla \mathbf{F}(\mathbf{x}^*), \tag{73}$$

where

$$\mathbf{\Psi}_3^* \triangleq \lim_{k \to \infty} \sum_{\tau=0}^{k-1} \mathbf{\Phi}_3(k, \tau), \tag{74}$$

with $\mathbf{\Phi}_3(k,\tau)$ the transition matrix of (49), that is, $\mathbf{\Phi}_3(k,\tau) = \mathbf{M}_3(k-1)\mathbf{M}_3(k-2)\ldots\mathbf{M}_3(\tau)$.

Proof: Let θ, m_l and m_u be the positive numbers defined in Proposition 4.4. By (71), we have that $\frac{2}{m_u + m_l} \leq \bar{\alpha} \leq \frac{2}{m_u}$. Then for any $\alpha \in (0, \bar{\alpha}]$, according to Proposition 7.3, the eigenvalues of $\mathbf{A}\left[\mathbf{I} - \alpha \nabla^2 \mathbf{F}(\mathbf{x})\right]$ are strictly within the unit circle for all $\mathbf{x} \in B_c(\mathbf{x}^*, \theta)$. By choosing $\delta = \max\{-1 + \alpha m_u, 1 - \alpha m_l\}$, again by Proposition 7.3, we have that

$$\rho\left(\mathbf{A}\left[\mathbf{I} - \alpha \nabla^2 \mathbf{F}(\mathbf{x})\right]\right) \leq \delta \ \forall \mathbf{x} \in B_c(\mathbf{x}^*, \theta), \ \forall \alpha \in (0, \bar{\alpha}].$$

The main idea of the proof is showing that $\mathbf{x}_k - \mathbf{x}^* \in B_c(0, \theta)$ for all time k. If this is the case then $\rho(\mathbf{M}_3(k)) \leq \delta$ for all k since the Hessian $\nabla^2 \mathbf{F}(k)$ from $\mathbf{M}_3(k)$ is computed at a point on a line between $\mathbf{x}^*$ and $\mathbf{x}_k$, and therefore it belongs to $B_c(\mathbf{x}^*, \theta)$.

We proceed by induction. Assume that $\|\mathbf{x}_k - \mathbf{x}^*\| \leq \theta$. Then according to iteration (51)

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq \delta \|\mathbf{x}_k - \mathbf{x}^*\| + \alpha \|\nabla \mathbf{F}(\mathbf{x}^*)\| \leq \delta\theta + \alpha \|\nabla \mathbf{F}(\mathbf{x}^*)\|. \tag{75}$$

Depending on the value of α, we distinguish two cases. If $\alpha \in \left(0, \ \frac{2}{m_l + m_u}\right)$ then $\delta = 1 - \alpha m_l$ and (62) becomes

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq (1 - \alpha m_l)\theta + \alpha \|\nabla \mathbf{F}(\mathbf{x}^*)\| =$$
$$= \theta - \alpha\left(m_l \theta - \|\nabla \mathbf{F}(\mathbf{x}^*)\|\right) \leq \theta,$$

where the last inequality followed from (71). In fact one can check that $\theta - \alpha\left(m_l\theta - \|\nabla \mathbf{F}(\mathbf{x}^*)\|\right) > 0$ for $\alpha \in \left(0, \ \frac{2}{m_l + m_u}\right)$, and therefore the above inequality makes sense.

If however $\alpha \in \left[\frac{2}{m_l + m_u}, \ \bar{\alpha}\right)$ then $\delta = -1 + \alpha m_u$ and (75) becomes

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq (-1 + \alpha m_u)\theta + \alpha \|\nabla \mathbf{F}(\mathbf{x}^*)\| =$$
$$-\theta + \alpha\left(m_u\theta + \|\nabla \mathbf{F}(\mathbf{x}^*)\|\right) \leq -\theta + \bar{\alpha}\left(m_u\theta + \|\nabla \mathbf{F}(\mathbf{x}^*)\|\right) =$$
$$-\theta + \frac{2\theta}{m_u\theta + \|\nabla \mathbf{F}(\mathbf{x}^*)\|}\left(L\theta + \|\nabla \mathbf{F}(\mathbf{x}^*)\|\right) = \theta.$$

Therefore, for our choice of $\bar{\alpha}$ we have that $\|\mathbf{x}_k - \mathbf{x}^*\| \leq \theta$ for all $k \geq 0$ and consequently

$$\rho(\mathbf{M}_3(k)) = \rho\left(\mathbf{A}\left[\mathbf{I} - \alpha \nabla^2 \mathbf{F}(k)\right]\right) \leq \delta < 1, \ \forall k \geq 0.$$

The solution of (64) is given by

$$\mathbf{x}_k - \mathbf{x}^* = \mathbf{\Phi}_3(k, 0)\left(\mathbf{x}_0 - \mathbf{x}^*\right) - \alpha \sum_{\tau=0}^{k-1} \mathbf{\Phi}_3(k, \tau) \nabla \mathbf{F}(\mathbf{x}^*),$$

or

$$\|\mathbf{x}_k-\mathbf{x}^*\| \le \|\mathbf{\Phi}_3(k,0)\|\|\mathbf{x}_0-\mathbf{x}^*\|+\alpha\sum_{\tau=0}^{k-1}\|\mathbf{\Phi}_3(k,\tau)\|\|\mathbf{A}\nabla\mathbf{F}(\mathbf{x}^*)\|, \tag{76}$$

and since $\|\mathbf{\Phi}_3(k,\tau)\| \le \|\mathbf{M}_3(k-1)\|\|\mathbf{M}_3(k-2)\|\ldots\|\mathbf{M}_3(\tau)\| = \delta^{k-\tau}$, inequality(72) follows. Using the same approach as in the proof of Proposition 7.2, we can show that the series $\sum_{\tau=0}^{k-1}\mathbf{\Phi}_3(k,\tau)$ converges and therefore there exits $\mathbf{\Psi}_3^*$ so that

$$\lim_{k\to\infty}\sum_{\tau=0}^{k-1}\mathbf{\Phi}_3(k,\tau)=\mathbf{\Psi}_3^*.$$

We conclude the proof by noticing that

$$\lim_{k\to\infty}\mathbf{\Phi}_3(k,0)\,(\mathbf{x}(0)-\mathbf{x}^*)=\mathbf{0},$$

and therefore

$$\lim_{k\to\infty}\mathbf{x}_k-\mathbf{x}^* = -\alpha\mathbf{\Psi}_3^*\mathbf{A}\nabla\mathbf{F}(\mathbf{x}^*).$$

■

The above result shows that, similarly to Algorithm (A_2), Algorithm (A_3) does not guarantee convergence to a minimizer of (P_1), but rather to a neighborhood around x^* whose size depends on $\nabla f_i(x^*)$ and α. Convergence to x^* is ensured provided that

$$\nabla\mathbf{F}(\mathbf{x}^*)\in\text{Null}\left(\mathbf{\Psi}_3^*\mathbf{A}\right).$$

Interestingly, unlike Algorithm (A_2) if the communication graph is complete, Algorithm (A_3) does converge to a local minimizer $\mathbf{x}^*$. Indeed, if the communication graph is complete then $A=\frac{1}{N}\mathbb{1}\mathbb{1}'$ and each entry of the vector $\mathbf{A}\nabla\mathbf{F}(\mathbf{x}^*)$ is given by

$$[\mathbf{A}\nabla\mathbf{F}(\mathbf{x}^*)]_i=\sum_{i=1}^{N}\nabla f_i(x^*)=0,$$

where the last equality followed from the first order necessary conditions of (P_1).

C. Comparison of the performance of Algorithms (A_2) and (A_3)

We provided in the previous two subsections estimates on the rate of convergence of the Algorithms (A_2) and (A_3). A natural question is if these estimates are tights. Using a simple example, we show in the following that in fact the upper-bounds on the rate of convergence can be reached.

Let $f_i(x)=\frac{1}{2}x^2$ for all i and consequently $\nabla^2\mathbf{F}(\mathbf{x})=\mathbf{I}$. For this particular example, the matrices controlling the convergence of the algorithms (A_2) and (A_3) are $\mathbf{M}_2=\mathbf{A}-\alpha\mathbf{I}$ and $\mathbf{M}_3=(1-\alpha)\mathbf{A}$, respectively. According to the Propositions 7.1 and 7.3, the upper-bounds on the spectral radius of the matrices $\mathbf{M}_2$ and $\mathbf{M}_3$ are given by

$$\rho(\mathbf{M}_2)\le\begin{cases}1-\alpha & \alpha\in\left(0,\ \frac{1+\sigma_{1,\mathbf{A}}}{2}\right),\\ -\sigma_{1,\mathbf{A}}+\alpha & \alpha\in\left[\frac{1+\sigma_{1,\mathbf{A}}}{2},\ 1+\sigma_{1,\mathbf{A}}\right),\end{cases} \tag{77}$$

and

$$\rho(\mathbf{M}_3)\le\begin{cases}1-\alpha & \alpha\in(0,\ 1),\\ -1+\alpha & \alpha\in[1,\ 2),\end{cases} \tag{78}$$

respectively.

On the other hand the eigenvalues of the matrices $\mathbf{M}_2$ and $\mathbf{M}_3$ are given by

$$\sigma(\mathbf{M}_2)=\{\sigma_{1,\mathbf{A}}-\alpha,\sigma_{2,\mathbf{A}}-\alpha,\ldots,\sigma_{N-1,\mathbf{A}}-\alpha,1-\alpha\}, \tag{79}$$

and

$$\sigma(\mathbf{M}_3) = \{(1-\alpha)\sigma_{1,\mathbf{A}}, (1-\alpha)\sigma_{2,\mathbf{A}}, \dots, (1-\alpha)\sigma_{N-1,\mathbf{A}}, 1-\alpha\}, \tag{80}$$

and therefore, $\rho(\mathbf{M}_2) = \max\{|\sigma_{1,\mathbf{A}} - \alpha|, |1-\alpha|\}$ and $\rho(\mathbf{M}_3) = \max\{|(1-\alpha)\sigma_{1,\mathbf{A}}|, |1-\alpha|\} = |1-\alpha|$. Forcing the spectral radius of $\mathbf{M}_2$ and $\mathbf{M}_3$ to be smaller than one, we note that indeed the upper-bounds (77) and (78) are reached, and therefore the bounds are tight.

An interesting question is which one of the Algorithms (A_2) and (A_3) performs better. In the following we show that at least in a worse-case scenario, that is, the upper-bounds on the spectral radius of matrices $\mathbf{M}_2$ and $\mathbf{M}_3$ are reached, Algorithm (A_3) converges faster than Algorithm (A_2). Let $\rho_u(\mathbf{M}_2)$ and $\rho_u(\mathbf{M}_3)$ denote the upper-bounds on the spectral radius of the aforementioned matrices, as per Propositions 7.1 and 7.3. Using the results from the two propositions it can be checked that $\rho_u(\mathbf{M}_3) \leq \rho_u(\mathbf{M}_2)$ for all $\alpha \in \left(0, \frac{1+\sigma_{1,\mathbf{A}}}{m_u}\right)$. In particular we have that

$$\begin{array}{rcl} \rho_u(\mathbf{M}_2) & = & 1-\alpha m_l = \rho_u(\mathbf{M}_3) \\ & \forall & \alpha \in \left(0, \dfrac{1+\sigma_{1,\mathbf{A}}}{m_l+m_u}\right), \\ \rho_u(\mathbf{M}_2) & = & -\sigma_{1,\mathbf{A}} + \alpha m_l > 1-\alpha m_l = \rho_u(\mathbf{M}_3) \\ & \forall & \alpha \in \left[\dfrac{1+\sigma_{1,\mathbf{A}}}{m_l+m_u}, \dfrac{2}{m_l+m_u}\right), \\ \rho_u(\mathbf{M}_2) & = & -\sigma_{1,\mathbf{A}} + \alpha m_l > -1+\alpha m_u = \rho_u(\mathbf{M}_3) \\ & \forall & \alpha \in \left[\dfrac{2}{m_l+m_u}, \dfrac{1+\sigma_{1,\mathbf{A}}}{m_u}\right). \end{array} \tag{81}$$

In addition, for $\alpha \in \left[\frac{1+\sigma_{1,\mathbf{A}}}{m_u}, \frac{2}{m_u}\right)$ although we have that $\rho_u(\mathbf{M}_3) = -1+\alpha m_u < 1$, $\rho_u(\mathbf{M}_2)$ cannot be guaranteed to be smaller than one. Therefore we have that, at least in a worse-case scenario (that is, the upper-bounds on the spectral radius are reached) Algorithm (A_3) converges faster than Algorithm (A_2). In [16], it is indeed confirmed that Algorithm (A_3) outperforms Algorithm (A_2), at least from the point of view of the rate of convergence[2]. The authors were able to compute exactly the spectral radius of matrices $\mathbf{M}_2$ and $\mathbf{M}_3$, but only because they considered quadratic cost functions.

We saw earlier that the two consensus-based algorithms, in the case of a constant step-size, do not converge exactly to the (local) minimizer. Hence, another interesting question is how close do the two algorithms get to the local minimizer. In the following we show that, as in the case of the rate of convergence, in a worse-case scenario, Algorithm (A_3) is guaranteed to be closer to the (local) minimizer. Using the results of Proposition 7.2, in the case of Algorithm (A_2), we have

$$\lim_{k\to\infty} \sup \|\mathbf{x}_k - \mathbf{x}^*\| \leq \alpha \|\boldsymbol{\Psi}_2^*\| \|\nabla \mathbf{F}(\mathbf{x}^*)\| \leq \frac{\alpha}{1-\rho_u(\mathbf{M}_2)} \|\nabla \mathbf{F}(\mathbf{x}^*)\|. \tag{82}$$

Similarly, from Proposition 7.4, in the case of Algorithm (A_3) we have

$$\lim_{k\to\infty} \sup \|\mathbf{x}_k - \mathbf{x}^*\| \leq \alpha \|\boldsymbol{\Psi}_3^*\| \|\mathbf{A}\nabla \mathbf{F}(\mathbf{x}^*)\| \leq \frac{\alpha}{1-\rho_u(\mathbf{M}_3)} \|\nabla \mathbf{F}(\mathbf{x}^*)\|, \tag{83}$$

where $\|\cdot\|$ denotes the Euclidean norm, and the last inequalities followed from the definitions of $\boldsymbol{\Psi}_2^*$ and $\boldsymbol{\Psi}_3^*$, shown in (61) and (74), respectively. Since according to (81) we have that $\rho_u(\mathbf{M}_3) \leq \rho_u(\mathbf{M}_2)$, Algorithm (A_3) is guaranteed to converge to a point in a neighborhood around the minimizer; neighborhood whose diameter is smaller than the diameter of the neighborhood resulting from Algorithm (A_2). In addition, as expected, the size of the neighborhoods decreases with α, which in turn decreases the rate of convergence.

[2] The authors refer to Algorithm (A_2) as *consensus strategy* and to Algorithm (A_3) as *ATC diffusion*

VIII. Numerical example

In this section we test our distributed algorithm on the "Weber point of a set of points" problem, where a group of N agents want to find a point x in the plane whose sum of weighted distances from a given set of points y_1, …, y_N is minimized. Formally expressed, the common goal of the agents is to minimize the function

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{N} f_i(x),$$

where $f_i(x) = w_i\|x - y_i\|$, for $i = 1,\ldots,N$. In the following numerical simulations we choose $n = N = 12$, $w_i = i$ and $y_i = ie_i$, where $\{e_i\}$ is the standard Euclidean basis. We assume that the agents interact through a communication network with a circular structure, shown in Figure 1. In Figure 2 we plot the errors between the estimates of the agents and the minimizer $\|x_{i,k} - x^*\|$, as generated by Algorithm (A_1), where we use $\alpha = 0.025$ and the non-weighted Laplacian corresponding to the graph in Figure 1. We increase the eigenvalues of the Laplacian L by multiplying it with a scalar $c = 4$. To ensure the stability of the algorithm we are forced to pick smaller values for α. Figure 3 shows the numerical simulation of Algorithm (A_1) for $\alpha = 0.002$ and Laplacian $4L$. We note that the values of the estimate tend to be closer to each other. The intuition behind this phenomenon may be the heavier "weight" put on the dynamics induced by the equality constraint of Problem (P_2) which forces the values of the estimates to be closer. A similar phenomenon can be observed in the case of the consensus-based distributed algorithms, when the connectivity of the graph is improved.

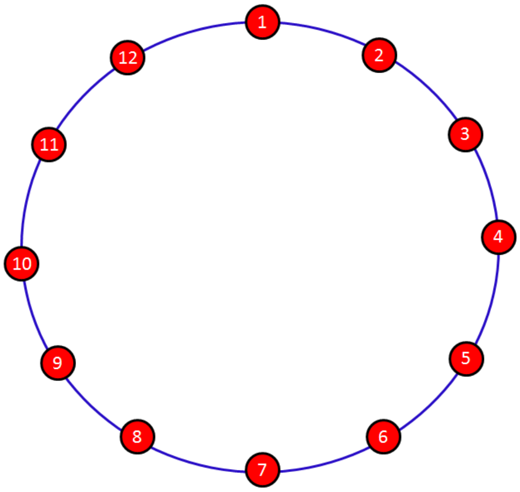

Fig. 1: Communication network

We compare in the following Algorithm (A_1) with Algorithms (A_2) and (A_3). We would like to point out that the comparison is not easy since the algorithms have different parameters. For instance, indeed we can create a stochastic matrix from Laplacian L using the formula $A = I - \gamma L$, but we can obtain an infinity of such stochastic matrices. Therefore we chose a stochastic matrix A that corresponds to the communication graph and that minimizes the second largest eigenvalue in modulus. Figures 4 and 5 present a comparison between Algorithms (A_1), (A_2) and (A_3), where in the case of (A_1) we used the non-weighted Laplacian. We plot the average of the errors between the current estimates and the minimizer, that is, $\frac{1}{N}\sum_{i=1}^{N}\|x_{i,k} - x^*\|$. As expected from the theoretical results, Algorithms (A_2) and (A_3) do not converge to the minimizer. For large values of α the consensus-based algorithm appear to converge faster. If however we want Algorithms (A_2) and (A_3) to be more precise, we need to decrease the step-size, but as a consequence we decrease the rate of convergence as well, as shown in Figure 5. In addition, as suggested by our analysis, for the considered example, Algorithm (A_3) does perform better (A_2), both in terms of rate of convergence and precision.

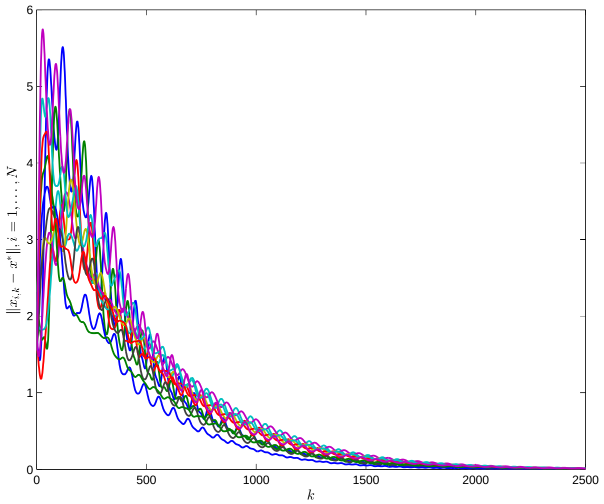

Fig. 2: Convergence of Algorithm (A_1) for $\alpha = 0.025$ and

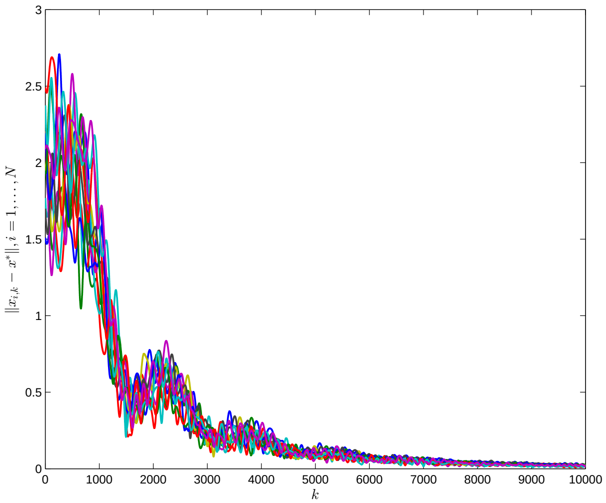

Fig. 3: Convergence of Algorithm (A_1) for $\alpha = 0.002$ and Laplacian $4L$

From the above simulations we note that for the same values of α, Algorithms (A_2) and (A_3) appear to be faster than Algorithm (A_1). However, if a small error between the estimates and the minimizer is desired, than the value of α that can achieve this error may result to be considerable small, and therefore the rate of convergence (A_2) and (A_3) is decreased considerable. For example, to achieve an average error of 0.2, the value of alpha should be roughly $\alpha = 0.0001$. In the case of Algorithm (A_1), as long as α is chosen to ensure the stability of the iteration, the average error will always converge to zero. Figure 6 shows the evolution of the average errors for the three algorithms. In the case of Algorithm (A_1) we chose $\alpha = 0.01$, which ensures the stability of the algorithm. In the case of Algorithm (A_2) and (A_3), we chose $\alpha = 0.0001$, so that the desired precision is reached. As expected, Algorithm (A_1) converge much faster than the consensus-based algorithms. In conclusion, if precision is required than Algorithm (A_1) would be the most indicated. If however we are willing to sacrifice precision to gain higher rate of the convergence, than the consensus-based distributed algorithm are more suitable.

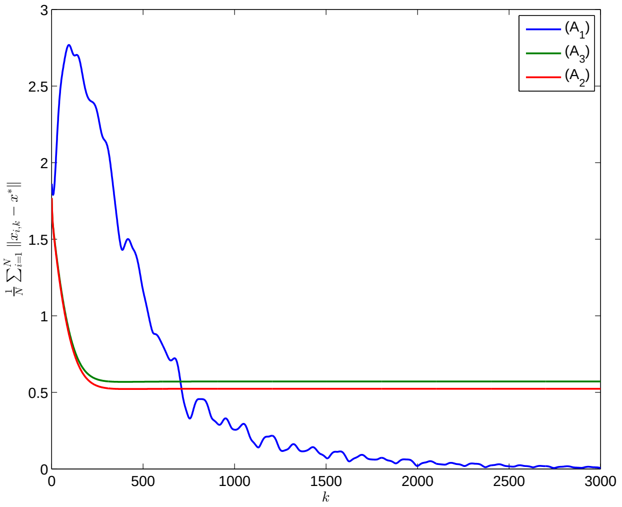

Fig. 4: Convergence of Algorithms (A_1), (A_2) and (A_3) for $\alpha = 0.01$

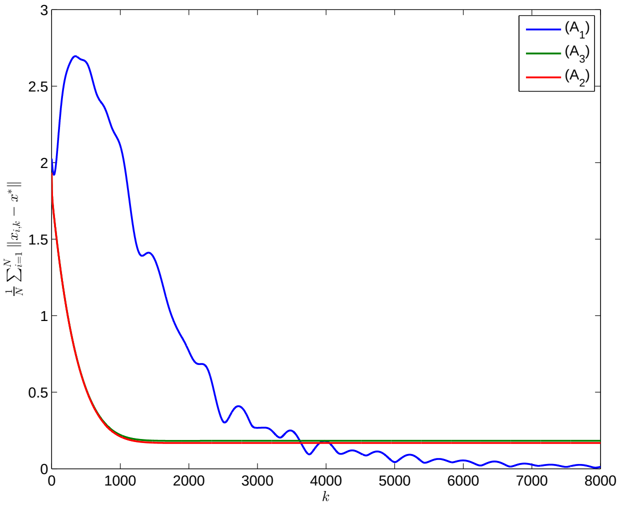

Fig. 5: Convergence of Algorithms (A_1), (A_2) and (A_3) for $\alpha = 0.003$

IX. Conclusions

We presented a distributed algorithm for solving a particular type of optimization problems. In this problem, the cost function is expressed as a sum of functions and each agent is aware of only one function of the sum. We demonstrated the non-heuristic nature of the algorithm by showing that it is the byproduct of applying a first-order method for solving the first order necessary conditions of a lifted optimization problem; optimization problem whose solution embeds the solution of our original problem. We presented a convergence analysis of the algorithm and showed qualitatively how the connectivity of the network influences the rate of convergence. In addition, we revisited two consensus-based distributed algorithms and gave sufficient conditions so that their use can be extended to non-convex cost functions. We showed that when a constant step-size is used, unlike our algorithm, the consensus-based algorithms do not guarantee convergence to a local minimizer even when the initial values of the agents are near a local minimizer, so that stability is ensured.

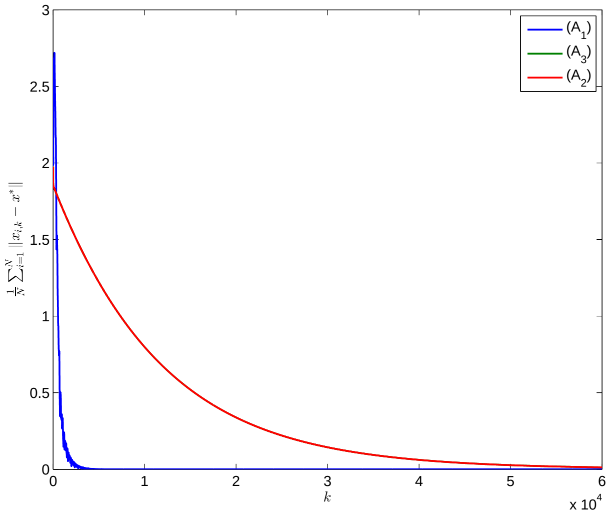

Fig. 6: Convergence of Algorithms (A_1) for $\alpha = 0.01$ and convergence of Algorithms (A_2) and (A_3) for $\alpha = 0.0001$

References

[1] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1999.

[2] E. Camponogara, D. Jia, B.H. Krogh, and S. Talukdar. Distributed model predictive control. *Control Systems, IEEE*, 22(1):44 –52, feb 2002.

[3] J. Chen and A.H. Sayed. Diffusion adaptation strategies for distributed optimization and learning over networks. *Signal Processing, IEEE Transactions on*, 60(8):4289 –4305, aug. 2012.

[4] V.I. Istratescu. *Fixed Point Theory, An Introduction*. D.Reidel, the Netherlands., 1981.

[5] B. Johansson. On distributed optimization in network systems. *Ph.D. dissertation in Telecommunication*, 2008.

[6] B. Johansson, T. Keviczky, M. Johansson, and K.H. Johansson. Subgradient methods and consensus algorithms for solving convex optimization problems. *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 4185–4190, Dec 2008.

[7] I. Lobel and A. Ozdaglar. Distributed subgradient methods for convex optimization over random networks. *Automatic Control, IEEE Transactions on*, 56(6):1291–1306, june 2011.

[8] I. Lobel, A. Ozdaglar, and D. Feijer. Distributed multi-agent optimization with state-dependent communication. *Mathematical Programming*, 129(2):255–284, 2011.

[9] I. Matei and J.S. Baras. Performance evaluation of the consensus-based distributed subgradient method under random communication topologies. *Selected Topics in Signal Processing, IEEE Journal of*, 5(4):754–771, aug. 2011.

[10] A. Nedic. Asynchronous broadcast-based convex optimization over a network. *Automatic Control, IEEE Transactions on*, 56(6):1337 –1351, june 2011.

[11] A. Nedic and A. Ozdalgar. Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control*, 54(1):48–61, Jan 2009.

[12] D.P. Palomar and Mung Chiang. A tutorial on decomposition methods for network utility maximization. *Selected Areas in Communications, IEEE Journal on*, 24(8):1439 –1451, aug. 2006.

[13] S. Sundhar Ram, A. Nedic, and V. V. Veeravalli. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of Optimization Theory and Applications*, pages 516–545, 2010.

[14] S.S. Ram, V.V. Veeravalli, and A. Nedic. Distributed and recursive parameter estimation in parametrized linear state-space models. *Automatic Control, IEEE Transactions on*, 55(2):488–492, feb. 2010.

[15] K. Srivastava and A. Nedic. Distributed asynchronous constrained stochastic optimization. *Selected Topics in Signal Processing, IEEE Journal of*, 5(4):772 –790, aug. 2011.

[16] S.Y. Tu and A.H. Sayed. Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. *IEEE Trans. Signal Processing*, 60(12):6217–6234, 2012.

[17] P.P. Varaiya. *Notes on optimization*. Van Nostrand Reinhold Co., 1972.