THE INSTITUTE FOR SYSTEMS RESEARCH

ISR TECHNICAL REPORT 2013-02

A non-heuristic distributed algorithm for non-convex constrained optimization

Ion Matei
John S. Baras

A non-heuristic distributed algorithm for non-convex constrained optimization

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 problems where the cost function is expressed as a sum of functions, and where each function is associated to an agent. In addition, the agents can have equality constraints as well. The algorithm can be applied to non-convex optimization problems with equality constraints, it is not consensus-based and it is not an heuristic. We demonstrate that the distributed algorithm results naturally from applying a first order method to solve the first order necessary conditions of an augmented optimization problem with equality constraints; optimization problem whose solution embeds the solution of our original problem. We show that, provided the agents' initial values are sufficiently close to a local minimum, and the step-size is sufficiently small, under standard conditions on the cost and constraint functions, each agent converges to the local minimum at a linear rate.

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 each agent makes 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 such as network resource allocation, collaborative control, estimation and identification, and so on. In these type of applications a group of agents has as goal the optimization of a cost function under limited information. The limited information can take the form of communication with only a set of neighboring agents or/and awareness of only a part of the cost function (or constraints, if they exist).

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 wireless network resource allocation problems [14] 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 [13]. 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 each agent uses a combination of the standard (sub)gradient descent step with a consensus step to deal with the limited information about the cost function and about the actions of the agents, and takes the form:

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

where the indices i 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 function $f_i(x)$, computed at $x_{i,k}$, and $\alpha_{i,k}$ is the step-size of the (sub)gradient descent step.

Many subsequent versions of this algorithm appeared in the literature. The introduction of communication noise and errors on subgradients was addressed in [11], [15], while the case where the communication network is modeled as a random graph was treated in [8], [10]. Analyses of asynchronous versions of the algorithm can be found in [11], [17]. A further extension was proposed in [9], 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], [16].

In this paper we study a distributed optimization problem similar to the formulation proposed in [13], 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. In addition, we assume that each agent has an equality constraint, as well. Distributed algorithms for solving constrained optimization problems were already studied in the literature. The focus was on convex problems: the cost and constraint sets are assumed convex. The algorithms are based on a combination of a consensus step (to cope with the lack of complete information) and a gradient projection step and they consider that either all agents use the same constraint set [7], [11], [15] or each agent has its own set of constraints [12], [17]. We do not make any convexity assumptions on the cost and constraint functions, but we assume they are continuously differentiable. We propose a distributed, discrete-time algorithm that, under standard assumptions on the cost and constraint functions, guarantees convergence to a local minimizer (at a linear rate), provided that the initial values of the agents are close enough to a (local) minimum and a sufficiently small step-size is used. The most interesting aspect of this algorithm is that *it is not an heuristic algorithm*, but follows naturally from using a first order numerical method to solve the first order necessary conditions of an *augmented* optimization problem with equality constraints; optimization problem whose solution embeds the solution of our original problem.

The paper is organized as following: in Section II we formulate the constrained optimization problem and introduce a distributed optimization algorithm to solve it. Section III presents the origins of the algorithm by demonstrating that our initial optimization problem is equivalent to an augmented optimization problem with equality constraints. In Section IV we give sufficient conditions so that local convergence to a local minimum is achieved. We end the paper with some numerical examples and conclusions.

Notation and definitions: For a matrix A, its (i, j) entry is denoted by $[A]_{ij}$ and its transpose is given by A'. If A is a symmetric matrix, $A \succ 0$ ($A \succeq 0$) means that A is positive (semi-positive) definite. The nullspace and range of A are denoted by Null(A) and Range(A), respectively. The symbol $\otimes$ is used to represent the Kronecker product between two matrices. The vector of all ones is denoted by $\mathbb{1}$. 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 $F:\mathbb{R}^n\times\mathbb{R}^m \to \mathbb{R}$ be a function of variables (x,y). The block descriptions of the gradient and of the Hessian of F at (x,y) are given by $\nabla F(x,y)' = \left(\nabla_x F(x,y)', \nabla_y F(x,y)'\right)$, and

$$\nabla^2 F(x,y) = \begin{pmatrix} \nabla^2_{xx} F(x,y) & \nabla^2_{x,y} F(x,y) \\ \nabla^2_{x,y} F(x,y) & \nabla^2_{yy} F(x,y) \end{pmatrix},$$

respectively. Let $\{A_i\}_{i=1}^N$ be a set of matrices. By diag$(A_i, i=1,\dots,N)$ we understand a block diagonal matrix, where the i^{th} block matrix is given by A_i.

II. Problem description

In this section we describe the setup of our problem. First we present the communication model after which we introduce the optimization model.

A. Communication model

A set of N agents interact with each other subject to an undirected communication graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$, where $\mathcal{V} = \{1,2,\dots,N\}$ is the set of nodes and $\mathcal{E} = \{e_{ij}\}$ 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 $\mathcal{N}_i \triangleq \{j \mid (i,j)\in\mathcal{E}\}$ the set of neighbors of agent i, and by L the Laplacian of graph $\mathcal{G}$ defined as

$$[L]_{ij} = \begin{cases} -l_{ij} & j \in \mathcal{N}_i, \\ \sum_{j\in\mathcal{N}_i} l_{ij} & i = j, \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

where l_{ij} are positive scalars.

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 $\text{Null}(L) = \{\gamma\mathbb{1} \mid \gamma \in \mathbb{R}\}$, and $\text{Null}(L') = \{\gamma\eta \mid \gamma \in \mathbb{R}\}$, 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 $\text{Null}(\mathbf{L}) = \{\mathbb{1} \otimes x \mid x \in \mathbb{R}^n\}$, and $\text{Null}(\mathbf{L}') = \{\eta \otimes x \mid x \in \mathbb{R}^n\}$, respectively;

(d) Let $\mathbf{x}$ be a vector in $\mathbb{R}^{nN}$. Then the orthogonal projection of $\mathbf{x}$ on $\text{Null}(\mathbf{L}')$ 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

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

and a vector valued function $h : \mathbb{R}^n \to \mathbb{R}^N$ where $h \triangleq (h_1, h_2, \ldots, h_N)'$, with $N \leq n$.

We make the following assumptions on the functions f and h and on the communication model.

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

(b) Agent i knows only functions $f_i(x)$ and $h_i(x)$;

(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 goal of the agents is to minimize the following optimization problem with equality constraints

$$(P_1) \quad \min_{x \in \mathbb{R}^n} \quad f(x),$$
$$\text{subject to:} \quad h(x) = 0,$$

under Assumptions 2.1.

Let x^* be a local minimum of (P_1) and let

$$\nabla h(x^*) \triangleq [\nabla h_1(x^*), \nabla h_2(x^*), \ldots, \nabla h_N(x^*)]$$

be a matrix whose columns are the gradients of functions $h_i(x)$ computed at x^*. The following assumption is used to guarantee the uniqueness of the Lagrange multiplier vector μ^* appearing in the first order necessary conditions of (P_1), namely

$$\nabla f(x^*) + \nabla h(x^*)\mu^* = 0.$$

Assumption 2.2: Let x^* be a local minimum. The matrix $\nabla h(x^*)$ is full rank, or equivalently, the vectors $\{\nabla h_i(x^*)\}_{i=1}^N$ are linearly independent.
Together with some additional assumptions on $f(x)$ and $h(x)$, Assumption 2.2 is also used the prove local convergence of a first-order numerical method for solving the first order necessary conditions of (P_1) (see for example Section 4.4.1, page 386 of [1]). As we will see in the next sections, the same assumption will be used to prove local convergence for a distributed algorithm used to solve (P_1).

Remark 2.2: We assumed that each agent has an equality constraint of the type $h_i(x) = 0$. All the results presented in what follows can be easily adapted for the case where only $m \leq N$ agents have equality constraints, as long as $m \leq n$.

C. Distributed algorithm

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

$$(A_1) \quad x_{i,k+1} = x_{i,k} - \alpha \nabla f_i(x_{i,k}) - \alpha \mu_{i,k} \nabla h_i(x_{i,k}) - \tag{2}$$

$$- \alpha \sum_{j \in \mathcal{N}_i} \left(l_{ij}\lambda_{i,k} - l_{ji}\lambda_{j,k} \right), \ x_{i,0} = x_i^0, \tag{3}$$

$$\mu_{i,k+1} = \mu_{i,k} + \alpha h_i(x_{i,k}), \ \mu_{i,0} = \mu_i^0, \tag{4}$$

$$\lambda_{i,k+1} = \lambda_{i,k} + \alpha \sum_{j \in \mathcal{N}_i} l_{ij}(x_{i,k} - x_{j,k}), \ \lambda_{i,0} = \lambda_i^0, \tag{5}$$

where $\nabla f_i(x_{i,k})$ and $\nabla h_i(x_{i,k})$ denote the gradients of functions $f_i(x)$ and $h_i(x)$, respectively, 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 (1).

Remark 2.3: 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.

In Algorithm (A_1) the first index (i or j) refers to a particular agent, while index k refers to the discrete time. It can be observed that the algorithm is indeed distributed since for updating its current estimate agent i uses only *local* information, that is, its own information ($x_{i,k}$, $\mu_{i,k}$, $\lambda_{i,k}$, $\nabla f_i(x_{i,k})$ and $\nabla h_i(x_{i,k})$) 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 $l_{ij}\lambda_{i,k}$. Note that equation (2) is a standard gradient descent step combined with two additional terms used to cope with the local equality constraint and the lack of complete information. The exact origin of equations (4) and (5) will be made clear in the next section. Intuitively however, $\mu_{i,k}$ can be seen as the price paid by agent i for having its estimate outside the local constraint set, while $\lambda_{i,k}$ is the price paid by the same agent for having its estimate far away from the estimates of its neighbors.

In the next section we start building the infrastructure that will allow us to prove local convergence of Algorithm (A_1). Specifically, this infrastructure will allow us to show that if the agents' initial values are close enough to a local minimum x^* and the step-size α is sufficiently small, under some standard conditions on functions $f_i(x)$ and $h_i(x)$, all estimates $x_i(k)$ converge to the local minimum x^*. More importantly, we will show that algorithm (A_1) *is not an heuristic algorithm*, but it can be traced back to solving an *augmented* optimization problem with additional equality constraints, whose solution *embeds* the solution of the optimization problem (P_1), as well.

III. An equivalent optimization problem with equality constraints

In this section we define an augmented 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 augmented optimization problem.

Let us define the function $\mathbf{F} : \mathbb{R}^{nN} \to \mathbb{R}$ 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 functions $\mathbf{h} : \mathbb{R}^{nN} \to \mathbb{R}^N$ and $\mathbf{g} : \mathbb{R}^{nN} \to \mathbb{R}^{nN}$, where

$$\mathbf{h}(\mathbf{x}) = (\mathbf{h}_1(\mathbf{x}), \mathbf{h}_2(\mathbf{x}), \ldots, \mathbf{h}_N(\mathbf{x}))',$$

with $\mathbf{h}_i : \mathbb{R}^{nN} \to \mathbb{R}$ given by $\mathbf{h}_i(\mathbf{x}) = h_i(x_i)$, and

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

with $g_i : \mathbb{R}^{nN} \to \mathbb{R}^n$ 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 (1). 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

$$\begin{aligned} (P_2) \quad \min_{\mathbf{x} \in \mathbb{R}^{nN}} \quad & \mathbf{F}(\mathbf{x}), \\ \text{subject to:} \quad & \mathbf{h}(\mathbf{x}) = 0, \\ & \mathbf{g}(\mathbf{x}) = 0, \end{aligned}$$

which can be expressed more explicitly as

$$(P_2) \quad \min_{\mathbf{x} \in \mathbb{R}^{nN}} \quad \mathbf{F}(\mathbf{x}), \tag{6}$$

$$\text{subject to:} \quad \mathbf{h}(\mathbf{x}) = 0, \tag{7}$$

$$\mathbf{L}\mathbf{x} = 0. \tag{8}$$

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 minimum of (P_1) if and only if $\mathbf{x}^* = \mathbb{1}_N \otimes x^*$ is a local minimum 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 (8), we get that any local minimum $\mathbf{x}^*$ of (P_2) must be of the form $\mathbf{x}^* = \mathbb{1}_N \otimes x^*$, 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 $\mathbf{x} = \mathbb{1}_N \otimes x$. Applying this constraint, the cost function (6) becomes

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

and the equality constraint (7) becomes

$$\mathbf{h}(\mathbf{x}) = h(x) = 0,$$

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 $\mathbf{x}^* = \mathbb{1}_N \otimes x^*$. 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.

Let $\mathbf{x}^* = \mathbb{1} \otimes x^*$ denote a local minimum of (P_2). From the theory concerning optimization problems with equality constraints (see for example Chapter 3, page 15 of [18], or Chapter 3, page 253 of [1]), the first order necessary conditions for (P_2) ensure the existence of $\lambda_0^* \in \mathbb{R}$, $\boldsymbol{\mu}^* \in \mathbb{R}^N$ and $\boldsymbol{\lambda}^* \mathbb{R}^{nN}$ so that

$$\lambda_0^* \nabla \mathbf{F}(\mathbf{x}^*) + \nabla \mathbf{h}(x^*)\boldsymbol{\mu}^* + \nabla \mathbf{g}(x^*)\boldsymbol{\lambda}^* =$$

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

where the matrix $\nabla \mathbf{h}(\mathbf{x}^*)$ is defined as

$$\nabla \mathbf{h}(\mathbf{x}^*) \triangleq [\nabla \mathbf{h}_1(\mathbf{x}^*), \nabla \mathbf{h}_2(\mathbf{x}^*), \ldots, \nabla \mathbf{h}_N(\mathbf{x}^*)].$$

The vectors $\nabla \mathbf{h}_i(\mathbf{x}^*)$ are the gradients of the functions $\mathbf{h}_i(\mathbf{x})$ at $\mathbf{x}^*$ with a structure given by

$$\nabla \mathbf{h}_i(\mathbf{x}^*)' =$$

$$= \left[\underbrace{0,\ldots,0}_{n \text{ zeros}}, \ldots, \underbrace{0,\ldots,0}_{n \text{ zeros}}, \underbrace{\nabla h_i(x^*)'}_{i^{\text{th}} \text{ component}}, \underbrace{0,\ldots,0}_{n \text{ zeros}}, \ldots, \underbrace{0,\ldots,0}_{n \text{ zeros}} \right], \tag{9}$$

as per definition of the function $\mathbf{h}_i(\mathbf{x})$.

Note that since $\mathbf{L}$ is not full rank, and therefore the matrix $[\nabla\mathbf{h}(\mathbf{x}^*), \mathbf{L}']$ is not full rank as well, the uniqueness of $\boldsymbol{\mu}^*$ and $\boldsymbol{\lambda}^*$ cannot be guaranteed. Before presenting the result concerning the first order necessary conditions for (P_2), we introduce two auxiliary results.

The first proposition recalls a well known result on the properties of the tangent cone to the constraint set at a local minimum of (P_1).

Proposition 3.2: Let Assumptions 2.1-(a) and 2.2 hold, let x^* be a local minimum of (P_1) and let Ω denote the constraint set, that is, $\Omega = \{x \mid h(x) = 0\}$. Then the tangent cone to Ω at x^* is given by

$$\mathrm{TC}(x^*, \Omega) = \mathrm{Null}\left(\nabla h(x^*)'\right),$$

where

$$\nabla h(x^*) \triangleq [\nabla h_1(x^*), \nabla h_2(x^*), \ldots, \nabla h_N(x^*)].$$

The second proposition characterizes the nullspace of the matrix $[\nabla\mathbf{h}(\mathbf{x}^*), \mathbf{L}']$, which will be used for expressing the tangent cone at a local minimum of (P_2).

Proposition 3.3: Let Assumptions 2.1 and 2.2 hold. The nullspace of the matrix $[\nabla\mathbf{h}(\mathbf{x}^*), \mathbf{L}']$ is given by

$$\mathrm{Null}\left([\nabla\mathbf{h}(\mathbf{x}^*), \mathbf{L}']\right) = \left\{(\mathbf{0}', \mathbf{v}')' \mid \mathbf{v} \in \mathrm{Null}\left(\mathbf{L}'\right)\right\}.$$

Proof: Let $\mathbf{u} \in \mathbb{R}^N$ and $\mathbf{v} \in \mathbb{R}^{nN}$ be two vectors. To characterize the nullspace of $[\nabla\mathbf{h}(\mathbf{x}^*), \mathbf{L}']$, we need to check for what values of $\mathbf{u}$ and $\mathbf{v}$ the equation

$$\nabla\mathbf{h}(\mathbf{x}^*)\mathbf{u} + \mathbf{L}'\mathbf{v} = \mathbf{0} \tag{10}$$

is satisfied. Using the definition of $\nabla\mathbf{h}_i(\mathbf{x}^*)$ shown in (9), equation (10) can be equivalently written as

$$\nabla h_i(x^*)\mathbf{u}_i + \sum_{j \in \mathcal{N}_i} \left(l_{ij}\mathbf{v}_i - l_{ji}\mathbf{v}_j\right) = 0, \;\; i = 1, \ldots, N,$$

where $\mathbf{u}_i$ are the entries of $\mathbf{u}$ and $\mathbf{v}_i$ are n-dimensional sub-vectors of $\mathbf{v}$.

Summing the above equations over i we obtain that

$$\sum_{i=1}^{N} \mathbf{u}_i \nabla h_i(x^*) = 0,$$

and since $\nabla h(x^*)$ is assumed full rank we must have that $\mathbf{u} = 0$ and the result follows. ■

We are now ready to state the first order necessary conditions, specialized for the problem (P_2).

Lemma 3.1 (first order necessary conditions for (P_2)*):* Let Assumptions 2.1 and 2.2 hold and let $\mathbf{x}^* = \mathbb{1}_N \otimes x^*$ be a local minimum for problem (P_2). There exist unique vectors $\boldsymbol{\mu}^*$ and $\boldsymbol{\lambda}^* \in \text{Range}(\mathbf{L})$ so that

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

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

Proof: The proof has two steps: first we show that the tangent cone to the constraint set at a local minimum is given by the nullspace of the matrix $[\nabla \mathbf{h}(\mathbf{x}^*), \mathbf{L}']'$; second we use the fact that $\nabla \mathbf{F}(\mathbf{x}^*)$ is orthogonal on the tangent cone, and use this to derive the necessary conditions.

Let us denote by $\boldsymbol{\Omega}$ the constraint set of (P_2), that is,

$$\boldsymbol{\Omega} \triangleq \{\mathbf{x} \mid \mathbf{h}(\mathbf{x}) = 0, \mathbf{g}(\mathbf{x}) = 0\}.$$

In the following we show that the tangent cone to $\boldsymbol{\Omega}$ at $\mathbf{x}^*$, denoted by $\text{TC}(\mathbf{x}^*, \boldsymbol{\Omega})$ is indeed the nullspace of $[\nabla \mathbf{h}(\mathbf{x}^*), \mathbf{L}']'$. In fact, all we have to show is that any vector in $\text{Null}\left([\nabla \mathbf{h}(\mathbf{x}^*), \mathbf{L}']'\right)$ belongs to $\text{TC}(\mathbf{x}^*, \boldsymbol{\Omega})$ as well, since it is well known that (the closure of the convex hull of) $\text{TC}(\mathbf{x}^*, \boldsymbol{\Omega})$ is included in $\text{Null}\left([\nabla \mathbf{h}(\mathbf{x}^*), \mathbf{L}']'\right)$. Let $\mathbf{u}$ be a vector in $\text{Null}\left([\nabla \mathbf{h}(\mathbf{x}^*), \mathbf{L}']'\right)$ and therefore it must satisfy

$$\nabla \mathbf{h}(\mathbf{x}^*)'\mathbf{u} = 0 \text{ and } \mathbf{L}'\mathbf{u} = 0. \tag{11}$$

From the second equation of (11), $\mathbf{u}$ must be of the form $\mathbf{u} = \mathbb{1} \otimes u$, for some $u \in \mathbb{R}^n$. From the first equation of (11), using the definition of $\nabla \mathbf{h}_i(\mathbf{x}^*)$ in (9) together with the particular structure of $\mathbf{u}$, we obtain that

$$\nabla h_i(x^*)'u = 0 \;\; \forall i = 1, \ldots, N,$$

or equivalently

$$u \in \text{Null}(\nabla h(x^*)').$$

For the first part of the proof we need to show that a vector $\mathbf{u} = \mathbb{1} \otimes u$, with $u \in \text{Null}(\nabla h(x^*)')$ belongs to $\text{TC}(\mathbf{x}^*, \boldsymbol{\Omega})$. 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

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

Choosing $\mathbf{o}(t) = \mathbb{1}_N \otimes o(t)$, where $o : \mathbb{R} \to \mathbb{R}^n$ is a function so that $\lim_{t\to 0, t>0} \frac{o(t)}{t} = 0$, we note that

$$\mathbf{g}\left(\mathbf{x}^* + t\mathbf{u} + \mathbf{o}(t)\right) = 0 \ \forall t > 0,$$

and therefore, all we are left to do is to check that

$$\mathbf{h}\left(\mathbf{x}^* + t\mathbf{u} + \mathbf{o}(t)\right) = 0 \ \forall t > 0, \tag{12}$$

as well. Making the observation that $\mathbf{x}^* + t\mathbf{u} + \mathbf{o}(t) = \mathbb{1} \otimes (x^* + tu + o(t))$, (12) is equivalent to showing that

$$h\left(x^* + tu + o(t)\right) = 0 \ \forall t > 0. \tag{13}$$

However, we showed previously that $u \in \text{Null}\left(\nabla h(x^*)'\right) = TC(x^*, \Omega)$, by Proposition 3.3. Therefore there exits a function $o(t)$ so that (13) is satisfied, which shows that indeed

$$\text{TC}\left(\mathbf{x}^*, \boldsymbol{\Omega}\right) = \text{Null}\left(\left[\nabla \mathbf{h}(\mathbf{x}^*), \mathbf{L}'\right]'\right),$$

and consequently $\text{TC}\left(\mathbf{x}^*, \boldsymbol{\Omega}\right)$ is a closed and convex subspace.

By Lemma 1[1] , page 50 of [18] we have that $\nabla \mathbf{F}(\mathbf{x}^*)$ is orthogonal on $\text{TC}\left(\mathbf{x}^*, \boldsymbol{\Omega}\right)$ and therefore $\nabla \mathbf{F}(\mathbf{x}^*)$ must belong to $\text{Range}\left(\left[\nabla \mathbf{h}(\mathbf{x}^*), \mathbf{L}'\right]\right)$. Consequently, there exist the vectors $\boldsymbol{\mu}^*$ and $\boldsymbol{\lambda}$ so that

$$-\nabla \mathbf{F}(\mathbf{x}^*) = \nabla \mathbf{h}(\mathbf{x}^*)\boldsymbol{\mu}^* + \mathbf{L}'\boldsymbol{\lambda}. \tag{14}$$

Noting that $\mathbb{R}^{nN}$ can be written as a direct sum between the nullspace of $\mathbf{L}'$ and the range of $\mathbf{L}$, there exist the orthogonal vectors $\boldsymbol{\lambda}^* \in \text{Range}\left(\mathbf{L}\right)$ and $\boldsymbol{\lambda}_\perp \in \text{Null}\left(\mathbf{L}'\right)$ so that $\boldsymbol{\lambda} = \boldsymbol{\lambda}^* + \boldsymbol{\lambda}_\perp$. Note that we can replace $\boldsymbol{\lambda}_\perp$ by any vector in $\text{Null}\left(\mathbf{L}'\right)$ and (14) still holds. The only thing left to do is to prove the uniqueness of $\boldsymbol{\mu}^*$ and $\boldsymbol{\lambda}^*$. We use a contradiction argument. Let $\tilde{\boldsymbol{\mu}} \neq \boldsymbol{\mu}^*$ and $\tilde{\boldsymbol{\lambda}} \neq \boldsymbol{\lambda}^*$ with $\tilde{\boldsymbol{\lambda}} \in \text{Range}\left(\mathbf{L}\right)$ be two vectors so that (14) is satisfied. Hence we have that

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

and

$$-\nabla \mathbf{F}(\mathbf{x}^*) = \nabla \mathbf{h}(\mathbf{x}^*)\tilde{\boldsymbol{\mu}} + \mathbf{L}'\tilde{\boldsymbol{\lambda}},$$

[1]The result states that given a local minimum x^* of a function $f(x)$, $h'\nabla f(x^*) \geq 0$ for all $h \in TC(x^*, \Omega)$. When $TC(x^*, \Omega)$ is a (closed, convex) subspace, orthogonality follows.

which gives

$$0 = \nabla \mathbf{h}(\mathbf{x}^*)(\boldsymbol{\mu}^* - \tilde{\boldsymbol{\mu}}) + \mathbf{L}'\left(\boldsymbol{\lambda}^* - \tilde{\boldsymbol{\lambda}}\right).$$

By Proposition 3.3 we have that

$$\text{Null}([\nabla \mathbf{h}(\mathbf{x}^*), \mathbf{L}']) = \left\{(\mathbf{0}', \mathbf{v}')' \mid \mathbf{v} \in \text{Null}(\mathbf{L}')\right\},$$

and therefore $\boldsymbol{\mu}^* = \tilde{\boldsymbol{\mu}}$ and $\boldsymbol{\lambda}^* = \tilde{\boldsymbol{\lambda}}$ since $\boldsymbol{\lambda}^* - \tilde{\boldsymbol{\lambda}} \in \text{Range}(\mathbf{L})$, and the result follows. ■

Under the assumption that the matrix $\nabla h(x^*)$ is full rank, the first order necessary conditions of (P_1) are given by

$$\nabla f(x^*) + \nabla h(x^*)\mu^* = 0,$$
$$h(x^*) = 0,$$

where the vector μ^* is unique (see for example Proposition 3.3.1, page 255, [1]). An interesting question is whether or not there is a connection between μ^* and $\boldsymbol{\mu}^*$ shown in the first order necessary conditions of (P_2). As proved in the following, the two vectors are in fact equal.

Proposition 3.4: Let Assumptions 2.1 and 2.2 hold, let $\mathbf{x}^* = \mathbb{1} \otimes x^*$ be a local minimum of (P_2) and let μ^* and $\boldsymbol{\mu}^*$ be the unique Lagrange multiplier vectors corresponding to the first order necessary conditions of (P_1) and (P_2), respectively. Then $\mu^* = \boldsymbol{\mu}^*$.

Proof: By Lemma 3.1, there exist two unique vector $\boldsymbol{\mu}^*$ and $\boldsymbol{\lambda}^* \in \text{Range}(\mathbf{L})$ so that

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

Using the structure of $\nabla \mathbf{F}(\mathbf{x}^*)$, $\mathbf{h}(\mathbf{x}^*)$ and $\mathbf{L}'$, the above equation can be equivalently expressed as

$$\nabla f_i(x^*) + \boldsymbol{\mu}_i^* \nabla h_i(x^*) + \sum_{j \in \mathcal{N}_i} \left(l_{ij}\boldsymbol{\lambda}_i^* - l_{ji}\boldsymbol{\lambda}_j^*\right),\ i = 1, \ldots, N, \tag{15}$$

where $\boldsymbol{\mu}_i^*$ are the scalar entries of $\boldsymbol{\mu}^*$ and $\boldsymbol{\lambda}_i^*$ are the n-dimensional sub-vectors of $\boldsymbol{\lambda}^*$. Summing up equations (15) over i, we obtain

$$\sum_{i=1}^{N} \nabla f_i(x^*) + \sum_{i=1}^{N} \nabla h_i(x^*)\boldsymbol{\mu}_i^* = 0.$$

Equivalently,

$$\nabla f(x^*) + \nabla h(x^*)\boldsymbol{\mu}^* = 0,$$

which is just the first order necessary condition for (P_1). But since $\boldsymbol{\mu}^*$ must be unique, it follows that $\boldsymbol{\mu}^* = \mu^*$. ■

To find a solution of problem (P_2) the first thing we can think about is solving the set of necessary conditions:

$$\nabla \mathbf{F}(\mathbf{x}) + \mathbf{L}^T \boldsymbol{\lambda} = 0, \tag{16}$$

$$\mathbf{h}(\mathbf{x}) = 0, \tag{17}$$

$$\mathbf{L}\mathbf{x} = 0. \tag{18}$$

Solving (16)-(18) does not guarantees finding local minimum (the sufficient conditions must also be checked), but at least they are among the solutions of the above nonlinear system of equations. One of the simplest approaches for solving (16)-(18) consists of using a first order method given by:

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

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \alpha \mathbf{h}(\mathbf{x}_k), \tag{20}$$

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha \mathbf{L}\mathbf{x}_k. \tag{21}$$

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

Remark 3.2: 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

$$\begin{aligned}
x_{1,k+1} &= x_{1,k} - \alpha\left[\lambda_{1,k} - \lambda_{3,k}\right] - \alpha\nabla f_1(x_{1,k}) - \alpha\nabla h_1(x_{1,k}),\\
x_{2,k+1} &= x_{2,k} - \alpha\left[\lambda_{2,k} - \lambda_{1,k}\right] - \alpha\nabla f_2(x_{2,k}) - \alpha\nabla h_2(x_{2,k}),\\
x_{3,k+1} &= x_{3,k} - \alpha\left[\lambda_{3,k} - \lambda_{2,k}\right] - \alpha\nabla f_3(x_{3,k}) - \alpha\nabla h_3(x_{3,k}),\\
\mu_{1,k+1} &= \mu_{1,k} + \alpha h_1(x_{1,k}),\\
\mu_{2,k+1} &= \mu_{2,k} + \alpha h_2(x_{2,k}),\\
\mu_{3,k+1} &= \mu_{3,k} + \alpha h_3(x_{3,k}),\\
\lambda_{1,k+1} &= \lambda_{1,k} + \alpha\left[x_{1,k} - x_{2,k}\right],\\
\lambda_{2,k+1} &= \lambda_{2,k} + \alpha\left[x_{2,k} - x_{3,k}\right],\\
\lambda_{3,k+1} &= \lambda_{3,k} + \alpha\left[x_{3,k} - x_{1,k}\right].
\end{aligned}$$

Note that although each agent can update its Lagrange multipliers using only information from its neighbors, it cannot update its estimate since it requires information from agents outside its neighborhood.

IV. 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 apply existing results for regular (local) minimizers, such as Proposition 4.4.2, page 388, [1], directly. Still, for a local minimum and Lagrange multipliers pair $(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$, 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 $f_i(x)$ and $h_i(x)$, $i = 1, \ldots, N$, the vectors $\mathbf{x}_k$ and $\boldsymbol{\mu}_k$ do indeed converge to $\mathbf{x}^*$ and $\boldsymbol{\mu}^*$, respectively. However, although under the same conditions $\boldsymbol{\lambda}_k$ does converge, it cannot be guaranteed that it converges to the unique $\boldsymbol{\lambda}^* \in \text{Range}(\mathbf{L})$ but rather to a point in the set $\{\boldsymbol{\lambda}^* + \text{Null}(\mathbf{L}')\}$.

The convergence of the algorithm (A_1) depends on the spectral properties of a particular matrix; properties analyzed in the following proposition. Before stating our first result of this section let us define the Lagrangian function of problem (P_2)

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}) \triangleq \mathbf{F}(\mathbf{x}) + \boldsymbol{\mu}'\mathbf{h}(\mathbf{x}) + \boldsymbol{\lambda}'\mathbf{L}\mathbf{x}, \tag{22}$$

used to simplify notation in the following lemma.

Lemma 4.1: Let Assumptions 2.1 and 2.2 hold, let α be a positive scalar, and let $(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ with $\boldsymbol{\lambda}^* \in \text{Range}(\mathbf{L})$, be a local minimum-Lagrange multipliers pair of (P_2). In addition, let the Hessian with respect to $\mathbf{x}$ of $\mathcal{L}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda})$ be positive definite, that is, $\nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*) \succ 0$. Then each eigenvalue of the matrix

$$\mathbf{B} = \begin{pmatrix} \nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*) & \nabla \mathbf{h}(\mathbf{x}^*) & \mathbf{L}' \\ -\nabla \mathbf{h}(\mathbf{x}^*)' & \mathbf{0} & \mathbf{0} \\ -\mathbf{L} & \mathbf{0} & \frac{1}{\alpha}\mathbf{J} \end{pmatrix},$$

has positive real part, where $\mathbf{J} \triangleq \frac{\eta\eta'}{\eta'\eta} \otimes I$, with η the left eigenvector of L corresponding to the zero eigenvalue.

Proof: Let β be an eigenvalue of $\mathbf{B}$ and let $(\mathbf{u}', \mathbf{v}', \mathbf{z}')' \neq 0$ be the corresponding eigenvector, where $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{z}$ are complex vectors of appropriate dimensions. We have that

$$\text{Re}\left\{ (\hat{\mathbf{u}}', \hat{\mathbf{v}}', \hat{\mathbf{z}}')\mathbf{B} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{z} \end{pmatrix} \right\} = \text{Re}(\beta)\left(\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 + \|\mathbf{z}\|^2 \right), \tag{23}$$

where $\hat{\mathbf{u}}$, $\hat{\mathbf{v}}$ and $\hat{\mathbf{z}}$ are the conjugates of $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{z}$, respectively. Using the structure of the matrix $\mathbf{B}$, we get

$$\text{Re}\left\{ (\hat{\mathbf{u}}', \hat{\mathbf{v}}', \hat{\mathbf{z}}')\mathbf{B} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{z} \end{pmatrix} \right\} = \text{Re}\left\{ \beta(\hat{\mathbf{u}}', \hat{\mathbf{v}}', \hat{\mathbf{z}}') \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{z} \end{pmatrix} \right\} =$$

$$\text{Re}\left\{ (\hat{\mathbf{u}}', \hat{\mathbf{v}}', \hat{\mathbf{z}}')\mathbf{B} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{z} \end{pmatrix} \right\} = \text{Re}\Big\{ \hat{\mathbf{u}}'\nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)\mathbf{u}$$

$$+\hat{\mathbf{u}}'\mathbf{L}'\mathbf{z} - \hat{\mathbf{z}}'\mathbf{L}\mathbf{u} + \hat{\mathbf{u}}'\nabla \mathbf{h}(\mathbf{x}^*)\mathbf{v} - \hat{\mathbf{v}}'\nabla \mathbf{h}(\mathbf{x}^*)'\mathbf{u} + \hat{\mathbf{z}}'\frac{1}{\alpha}\mathbf{J}\mathbf{z} \Big\} =$$

$$\text{Re}\left\{ \hat{\mathbf{u}}^T \nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)\mathbf{u} + \hat{\mathbf{z}}'\frac{1}{\alpha}\mathbf{J}\mathbf{z} \right\}. \tag{24}$$

By using (23) and (24) we further obtain

$$\text{Re}(\beta)\left(\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 + \|\mathbf{z}\|^2 \right) =$$

$$\text{Re}\left\{ \hat{\mathbf{u}}'\nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)\mathbf{u} \right\} + \text{Re}\left\{ \hat{\mathbf{z}}'\frac{1}{\alpha}\mathbf{J}\mathbf{z} \right\}.$$

Since $\mathbf{J} \triangleq \frac{\eta\eta^T}{\eta^T\eta} \otimes I_n$ is a semi-positive definite matrix and $\nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ is positive definite we have that

$$\mathrm{Re}(\beta)\left(\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 + \|\mathbf{z}\|^2\right) > 0,$$

as long as $\mathbf{u} \neq 0$ or $\mathbf{z} \notin \mathrm{Range}(\mathbf{L})$ and therefore $\mathrm{Re}(\beta) > 0$. In the case $\mathbf{u} = 0$ and $\mathbf{z} \in \mathrm{Range}(\mathbf{L})$ we get

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

from where we obtain

$$\nabla \mathbf{h}(\mathbf{x}^*)\mathbf{v} + \mathbf{L}'\mathbf{z} = 0.$$

But from Proposition 3.3, we have that $\mathbf{v} = 0$ and $\mathbf{z} \in \mathrm{Null}(\mathbf{L}')$ and since $\mathbf{z} \in \mathrm{Range}(\mathbf{L})$ as well, it must be that $\mathbf{z} = 0$. Hence we have a contradiction since we assumed that $(\mathbf{u}', \mathbf{v}', \mathbf{z}') \neq \mathbf{0}'$ and therefore the real part of β must be positive. In addition, it can be easily checked that the matrix $\mathbf{B}$ has n eigenvalues equal to $\frac{1}{\alpha}$ and their corresponding eigenspace is $\{(\mathbf{0}', \mathbf{0}', \mathbf{z}')' \mid \mathbf{z} \in \mathrm{Null}(\mathbf{L}')\}$. ■

The following theorem addresses the local convergence properties of Algorithm (A_1), which, under some assumptions on the functions $f_i(x)$ and $h_i(x)$, states that provided the initial values used in the Algorithm (A_1) are close enough to a solution of the first order necessary conditions of (P_2), and a small enough step-size α is used, the sequence $\{\mathbf{x}_k, \boldsymbol{\mu}_k, \boldsymbol{\lambda}_k\}$ converges to this solution.

Theorem 4.1: Let Assumptions 2.1 and 2.2 hold and let $(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ with $\boldsymbol{\lambda}^* \in \mathrm{Range}(\mathbf{L})$, be a local minimum-Lagrange multipliers pair of (P_2). Assume also that $\nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ is positive definite. Then there exits $\bar{\alpha}$, such that for all $\alpha \in (0, \bar{\alpha}]$, $(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^* + \mathrm{Null}(\mathbf{L}'))$ are points of attraction of iteration (19)-(21) and if the sequence $\{\mathbf{x}_k, \boldsymbol{\mu}_k, \boldsymbol{\lambda}_k\}$ converges to the set $(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^* + \mathrm{Null}(\mathbf{L}'))$, the rate of convergence of $\|\mathbf{x}_k - \mathbf{x}^*\|$, $\|\boldsymbol{\mu}_k - \boldsymbol{\mu}^*\|$ and $\left\|\boldsymbol{\lambda}_k - [\boldsymbol{\lambda}^* + \mathrm{Null}(\mathbf{L}')]\right\|$ is linear.

Proof: Using the Lagrangian function defined in (22), iteration (19)-(21) can be equivalently expressed as

$$\begin{pmatrix} \mathbf{x}_{k+1} \\ \boldsymbol{\mu}_{k+1} \\ \boldsymbol{\lambda}_{k+1} \end{pmatrix} = \bar{\mathbf{M}}_\alpha(\mathbf{x}_k, \boldsymbol{\mu}_k, \boldsymbol{\lambda}_k), \tag{25}$$

with

$$\bar{\mathbf{M}}_\alpha(\mathbf{x},\boldsymbol{\mu},\boldsymbol{\lambda}) = \begin{pmatrix} \mathbf{x}-\alpha\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x},\boldsymbol{\mu},\boldsymbol{\lambda}) \\ \boldsymbol{\mu}+\alpha\nabla_{\boldsymbol{\mu}}\mathcal{L}(\mathbf{x},\boldsymbol{\mu},\boldsymbol{\lambda}) \\ \boldsymbol{\lambda}+\alpha\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\mathbf{x},\boldsymbol{\mu},\boldsymbol{\lambda}) \end{pmatrix}.$$

It can be easily checked that $(\mathbf{x}^*,\boldsymbol{\mu}^*,\boldsymbol{\lambda}^*+\text{Null}(\mathbf{L}'))$ is a set of fixed points of $\bar{\mathbf{M}}_\alpha$. Let us now consider the transformation $\tilde{\boldsymbol{\lambda}} = (\mathbf{I}-\mathbf{J})\boldsymbol{\lambda}$, where $\mathbf{J} = \frac{\eta\eta'}{\eta'\eta}\otimes I$, with η the left eigenvector of the Laplacian L, corresponding to the zero eigenvalue. This transformation extracts the projection of $\boldsymbol{\lambda}$ on the nullspace of $\mathbf{L}'$ from $\boldsymbol{\lambda}$ and therefore $\tilde{\boldsymbol{\lambda}}$ is the error between $\boldsymbol{\lambda}$ and its orthogonal projection on $\text{Null}(\mathbf{L}')$. Under this transformation, iteration (25) becomes

$$\begin{pmatrix} \mathbf{x}_{k+1} \\ \boldsymbol{\mu}_{k+1} \\ \tilde{\boldsymbol{\lambda}}_{k+1} \end{pmatrix} = \mathbf{M}_\alpha(\mathbf{x}_k,\boldsymbol{\mu}_k,\tilde{\boldsymbol{\lambda}}_k)$$

with

$$\mathbf{M}_\alpha(\mathbf{x},\boldsymbol{\mu},\tilde{\boldsymbol{\lambda}}) = \begin{pmatrix} \mathbf{x}-\alpha\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x},\boldsymbol{\mu},\tilde{\boldsymbol{\lambda}}) \\ \boldsymbol{\mu}+\alpha\nabla_{\boldsymbol{\mu}}\mathcal{L}(\mathbf{x},\boldsymbol{\mu},\tilde{\boldsymbol{\lambda}}) \\ (\mathbf{I}-\mathbf{J})\tilde{\boldsymbol{\lambda}}+\alpha\nabla_{\tilde{\boldsymbol{\lambda}}}\mathcal{L}(\mathbf{x},\boldsymbol{\mu},\tilde{\boldsymbol{\lambda}}) \end{pmatrix},$$

where we used the fact that $(\mathbf{I}-\mathbf{J})\tilde{\boldsymbol{\lambda}} = (\mathbf{I}-\mathbf{J})\boldsymbol{\lambda}$ and $\mathbf{L}'\mathbf{J} = \mathbf{J}\mathbf{L} = \mathbf{0}$. Clearly $(\mathbf{x}^*,\boldsymbol{\mu}^*,\boldsymbol{\lambda}^*)$ is a fixed point for $\mathbf{M}_\alpha$ and we have that

$$\nabla\mathbf{M}_\alpha(\mathbf{x}^*,\boldsymbol{\mu}^*,\boldsymbol{\lambda}^*) = \mathbf{I}-\alpha\mathbf{B},$$

where

$$\mathbf{B} = \begin{pmatrix} \nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^*,\boldsymbol{\mu}^*,\boldsymbol{\lambda}^*) & \nabla\mathbf{h}(\mathbf{x}^*) & \mathbf{L}' \\ -\nabla\mathbf{h}(\mathbf{x}^*)' & \mathbf{0} & \mathbf{0} \\ -\mathbf{L} & \mathbf{0} & \frac{1}{\alpha}\mathbf{J} \end{pmatrix}.$$

By Lemma 4.1 we have that the real parts of the eigenvalues of $\mathbf{B}$ are positive and therefore we can find an $\bar{\alpha}$ so that for all $\alpha \in (0.\bar{\alpha}]$ the eigenvalues of $\nabla\mathbf{M}_\alpha(\mathbf{x}^*,\boldsymbol{\mu}^*,\boldsymbol{\lambda}^*)$ are strictly within the unit circle. Using a similar argument as in Proposition 4.4.1, page 387, [1], there exist a norm $\|\cdot\|$ and an open sphere $\mathcal{S}$ with respect to the norm centered at $(\mathbf{x}^*,\boldsymbol{\mu}^*,\boldsymbol{\lambda}^*)$ such that the induced norm of $\nabla\mathbf{M}_\alpha(\mathbf{x},\boldsymbol{\mu},\boldsymbol{\lambda})$ is less than one within the sphere $\mathcal{S}$. Therefore, using the mean value theorem, it can be argued that $\mathbf{M}_\alpha(\mathbf{x},\boldsymbol{\mu},\boldsymbol{\lambda})$ is a contraction map in the sphere $\mathcal{S}$ and the result follows by invoking the contraction map theorem (see for example Chapter 7 of [4]). ■

Let us know reformulate the above theorem so that the local convergence result can be applied to problem (P_1).

Corollary 4.1: Let Assumptions 2.1 and 2.2 hold and let (x^*,μ^*) be a local minimum-Lagrange multiplier pair of (P_1). Assume also that $\nabla^2 f_i(x^*)+\mu_i^*\nabla^2 h_i(x^*)$ is positive definite for all $i=1,\ldots,N$. Then there exits $\bar{\alpha}$, such that for all $\alpha\in(0,\bar{\alpha}]$, (x^*,μ^*) is a point of attraction for iteration (2) and (4), for all $i=1,\ldots,N$, and if the sequence $\{x_{i,k},\mu_{i,k}\}$ converges to (x^*,μ^*), then the rate of convergence of $\|x_{i,k}-x^*\|$ and $\|\mu_{i,k}-\mu^*\|$ is linear.

Proof: By Proposition 3.1 we have that $\mathbf{x}^*=\mathbb{1}\otimes x^*$ is a local minimum of (P_2) with corresponding Lagrange multipliers $(\boldsymbol{\mu}^*,\boldsymbol{\lambda}^*+\text{Null}(\mathbf{L}'))$, with $\boldsymbol{\lambda}^*\in\text{Range}(\mathbf{L})$. In addition, by Proposition 3.4 we have that $\boldsymbol{\mu}^*=\mu^*$. Using the definition of the Lagrangian function introduced in (22), we discover that

$$\nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^*,\boldsymbol{\mu}^*,\boldsymbol{\lambda}^*)=\text{diag}\left(\nabla^2 f_i(x^*)+\mu_i^*\nabla^2 h_i(x^*),i=1,\ldots,N\right).$$

But since we assumed that $\nabla^2 f_i(x^*)+\mu_i^*\nabla^2 h_i(x^*)\succ 0$ for all i, it follows that $\nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^*,\boldsymbol{\mu}^*,\boldsymbol{\lambda}^*)\succ 0$ as well. Using Theorem 4.1, the result follows. ■

Remark 4.1: In the previous corollary matrices $\nabla^2 f_i(x^*)+\mu_i^*\nabla^2 h_i(x^*)$ were assumed to be positive definite for all $i=1,\ldots,N$. If we apply directly results from the optimization literature (see for instance Proposition 4.4.2, page 388, [1]) concerning convergence of first-order methods used to compute local minima and their corresponding Lagrange multipliers, we only require $\sum_{i=1}^N\nabla^2 f_i(x^*)+\mu_i^*\nabla^2 h_i(x^*)$ to be positive definite, and not each element of sum. Obviously the assumption in Corollary 4.1 does imply the latter, but is not necessary. It appears that this additional constraint on $f_i(x)$ and $h_i(x)$ is the price paid for being able to prove local convergence of the distributed algorithm.

V. Conclusions

We presented a multi-agent distributed algorithm for solving a particular type of non-convex optimization problem with equality constraints. 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 and has its own local equality constraint. We demonstrated the non-heuristic nature of the algorithm by showing that it resulted from applying a first order numerical method to solve the first order necessary conditions of an augmented optimization problem; optimization problem whose solution embeds the solution of our original problem. In addition, we gave sufficient conditions for local convergence of the algorithms; conditions similar to the conditions used to prove local convergence of centralized algorithms.

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] B. Johansson, M. Rabi, and K.H. Johansson. A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Optimization*, 20(3):1157–1170, 2009.

[8] 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.

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

[10] 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.

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

[12] A. Nedic, A. Ozdaglar, and P.A. Parrilo. Constrained consensus and optimization in multi-agent networks. *Automatic Control, IEEE Transactions on*, 55(4):922 –938, april 2010.

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

[14] 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.

[15] 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.

[16] 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.

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

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