52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

Distributed Algorithms for Optimization Problems with Equality Constraints

Ion Matei, John S. Baras

Abstract— **In this paper we introduce two discrete-time, distributed optimization algorithms executed by a set of agents whose interactions are subject to a communication graph. The algorithms 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 algorithms are not consensus-based and can be applied to non-convex optimization problems with equality constraints. We demonstrate that the first distributed algorithm results naturally from applying a first order method to solve the first order necessary conditions for a lifted optimization problem with equality constraints; the solution of our original problem is embedded in the solution of this lifted optimization problem. Using an augmented Lagrangian idea, we derive a second distributed algorithm that requires weaker conditions for local convergence compared to the first algorithm. For both algorithms we address the local convergence properties.**

I. Introduction

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 functions 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 (un)directed graph. This formulation is often found in wireless network resource allocation problems [16] or in finite horizon optimal control problems with separable cost functions [4].

A first distributed algorithm for solving an optimization problem of the type described above was introduced in [15]. The algorithm, referred to as "distributed subgradient method", is used to minimize a convex function expressed as a sum of convex functions. In this algorithm each agent combines a standard (sub)gradient descent step with a consensus step; the latter is added to deal with the limited information about the cost function and about the actions of the agents.

Many subsequent versions of this algorithm appeared in the literature. The introduction of communication noise and errors on subgradients was addressed in [13], [17], while the case where the communication network is modeled as a random graph was treated in [9], [11]. Analyses of asynchronous versions of the algorithm can be found in [13], [19]. A further extension was proposed in [10], where the authors considered state-dependent communication topologies.

A modified version of the distributed subgradient method was introduced in [7], 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. The algorithms discussed above became popular in the signal processing community as well, being used for solving distributed filtering and parameter identification problems [5], [18].

In this paper we study a distributed optimization problem similar to the formulation proposed in [15]. 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 have already been studied in the literature. The focus has been 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 projected (sub)gradient descent step. They assume that either all agents use the same constraint set [8], [13], [17] or each agent has its own set of constraints [14], [19]. In this paper we do not make any convexity assumptions on the cost and constraint functions, but we assume they are continuously differentiable. We propose two distributed, discrete-time algorithms that, under suitable assumptions on the cost and constraint functions, guarantee convergence to a local minimizer (at a linear rate), provided that the initial values of the agents are close enough to a (local) minimizer and a sufficiently small step-size is used. The most interesting aspect of these algorithms is that *they are not heuristic algorithms*, but they follow naturally from using a first order numerical method to solve the first order necessary conditions of a *lifted* optimization problem with equality constraints; the solution of our original problem is embedded in the solution of this lifted optimization problem.

The paper is organized as follows: in Section II we

Ion Matei is with the Palo Alto Research Center (PARC), Palo Alto, CA (email: ion.matei@parc.com); John S. Baras is with the Institute for Systems Research at the University of Maryland, College Park, MD (email: baras@umd.edu).

Research partially supported by the Adaptive Vehicle Make (AVM), Component, Context, and Manufacturing Model Library - 2(C2M2L-2) program award D12PC00435 to PARC through the University of Maryland, by US Air Force Office of Scientific Research MURI grant FA9550-09-1-0538, by National Science Foundation (NSF) grant CNS-1035655, and by National Institute of Standards and Technology (NIST) grant 70NANB11H148.

formulate the constrained optimization problem and introduce two distributed optimization algorithms for solving this problem. Section III presents the origins of the algorithms by demonstrating that our initial optimization problem is equivalent to a lifted optimization problem with equality constraints. Section IV introduces a set of results used for the convergence analysis of the two algorithms; analysis detailed in Sections V and VI. Due to space limitations we have omitted the proofs of some of the results. These proofs can be found in the technical report [12] (`http://hdl.handle.net/1903/13693`).

Notations 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, $A > 0$ ($A \geq 0$) means that A is positive (semi-positive) definite. The nullspace and range of A are denoted by $\text{Null}(A)$ and $\text{Range}(A)$, 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 x and Q be a vector and a set of vectors, respectively. By $x+Q$ we understand the set of vectors produced by adding x to each element of Q, that is, $x+Q \triangleq \{x+y \mid y \in Q\}$. Let $\|\cdot\|$ be a vector norm. By $\|x-Q\|$ we denote the distance between the vector x and the set Q, that is, $\|x-Q\| \triangleq \inf_{y\in Q}\|x-y\|$. 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

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

respectively. Let $\{A_i\}_{i=1}^N$ be a set of matrices. By $\text{diag}(A_i,\ i=1,\dots,N)$ 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 $\epsilon > 0$, such 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 two distributed optimization algorithms.

A. Communication model

A set of N agents interact with each other subject to 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 $\mathcal{E} = \{e_{ij}\}$ 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 the graph $\mathcal{G}$ defined as

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

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). Throughout the rest of the paper we are going to assume that the Laplacian L is symmetric, that is, $l_{ij} = l_{ji}$.

Remark 2.1: Let $\bar{N} = \sum_{i=1}^N |\mathcal{N}_i|$, where $|\cdot|$ denotes the cardinality of a set. For a symmetric Laplacian L, there exists a matrix $S \in \mathbb{R}^{\bar{N}\times N}$ so that $L = S'S$ and $\text{Null}(L) = \text{Null}(S)$.

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 $h \triangleq (h_1,h_2,\dots,h_N)'$, with $N \leq n$.

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

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

(b) Agent i has knowledge of only functions $f_i(x)$ and $h_i(x)$, 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 and the Laplacian L is symmetric.

The common goal of the agents is to solve the following optimization problem with equality constraints

$$(P_1) \qquad \begin{aligned} &\min_{x\in\mathbb{R}^n} && f(x), \\ &\text{subject to:} && h(x) = 0, \end{aligned}$$

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

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

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

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

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

Assumption 2.2: Let x^* be a local minimizer of (P_1). 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 typically used to 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 two distributed algorithms 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.

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):

$$\begin{aligned} x_{i,k+1} &= x_{i,k} - \alpha\nabla f_i(x_{i,k}) - \alpha\mu_{i,k}\nabla h_i(x_{i,k}) - \\ &- \alpha\sum_{j\in\mathcal{N}_i}\left(l_{ij}\lambda_{i,k} - l_{ji}\lambda_{j,k}\right),\ x_{i,0} = x_i^0, \qquad (2)\\ \mu_{i,k+1} &= \mu_{i,k} + \alpha h_i(x_{i,k}),\ \mu_{i,0} = \mu_i^0, \qquad (3)\\ \lambda_{i,k+1} &= \lambda_{i,k} + \alpha\sum_{j\in\mathcal{N}_i} l_{ij}(x_{i,k} - x_{j,k}),\ \lambda_{i,0} = \lambda_i^0, \qquad (4)\end{aligned}$$

where $\alpha > 0$ is the step-size of the algorithm, $\nabla f_i(x_{i,k})$ and $\nabla h_i(x_{i,k})$ denote the gradients of functions $f_i(x)$ and $h_i(x)$, respectively, computed at $x_{i,k}$, and x_i^0, μ_i^0 and λ_i^0 are given scalars. In addition, the positive scalars l_{ij} are the entries of the Laplacian L of the graph $\mathcal{G}$ defined in (1).

A modified version of the previous algorithm, called henceforth Algorithm (A_2) is given by:

$$\begin{aligned} x_{i,k+1} &= x_{i,k} - \alpha\nabla f_i(x_{i,k}) - \alpha\mu_{i,k}\nabla h_i(x_{i,k}) - \\ &- \alpha\sum_{j\in\mathcal{N}_i}\left(l_{ij}\lambda_{i,k} - l_{ji}\lambda_{j,k}\right) - \alpha c h_i(x_{i,k})\nabla h_i(x_{i,k}) \\ &- \alpha c\sum_{j\in\mathcal{N}_i} l_{ij}(x_{i,k} - x_{j,k}),\ x_{i,0} = x_i^0, \qquad (5)\\ \mu_{i,k+1} &= \mu_{i,k} + \alpha h_i(x_{i,k}),\ \mu_{i,0} = \mu_i^0, \qquad (6)\\ \lambda_{i,k+1} &= \lambda_{i,k} + \alpha\sum_{j\in\mathcal{N}_i} l_{ij}(x_{i,k} - x_{j,k}),\ \lambda_{i,0} = \lambda_i^0, \qquad (7)\end{aligned}$$

where in addition to the parameters of Algorithm (A_1), we have a new positive parameter c. As shown later in the paper, the advantage of this algorithm is that it requires weaker conditions to prove local convergence, compared to Algorithm (A_1).

In Algorithms (A_1) and (A_2) the index i (or j) designates an agent while k denotes the discrete time. It can be observed that the algorithms are indeed distributed since for updating its current estimate $x_{i,k}$ agent i uses only *local* information, that is, its own information ($x_{i,k}$, $\mu_{i,k}$, $\lambda_{i,k}$, $\nabla f_i(x_{i,k})$ 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 $l_{ij}\lambda_{i,k}$. In the case of Algorithm (A_1), equation (2) is comprised of a standard gradient descent step and two additional terms used to cope with the local equality constraint and the lack of complete information. The exact origin of equations (3) and (4) will be made clear in the next sections. Intuitively however, $\mu_{i,k}$ can be seen as the price paid by agent i for satisfying the local equality constraint, while $\lambda_{i,k}$ is the price paid by the same agent for having its estimate $x_{i,k}$ far away from the estimates of its neighbors. Compared to Algorithm (A_1), Algorithm (A_2) adds two additional terms in equation (5). These terms have their origin in the use of an augmented Lagrangian and ensure the local convergence to a local minimizer under weaker conditions than the conditions used in the convergence analysis 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 will be 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. In addition, Algorithm (A_2) comes as a result of applying a first-order method to the same first order necessary conditions, but expressed in terms of an augmented Lagrangian.

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

$$\begin{aligned} (P_2)\quad \min_{\mathbf{x}\in\mathbb{R}^{nN}}\ & \mathbf{F}(\mathbf{x}), \qquad (8)\\ \text{subject to:}\ & \mathbf{h}(\mathbf{x}) = 0, \qquad (9)\\ & \mathbf{g}(\mathbf{x}) = \mathbf{L}\mathbf{x} = 0. \qquad (10)\end{aligned}$$

The Lagrangian function of problem (P_2) is a function $\mathcal{L} : \mathbb{R}^{nN} \times \mathbb{R}^N \times \mathbb{R}^{nN} \to \mathbb{R}$, defined as

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}) \triangleq \mathbf{F}(\mathbf{x}) + \boldsymbol{\mu}'\mathbf{h}(\mathbf{x}) + \boldsymbol{\lambda}'\mathbf{L}\mathbf{x}. \qquad (11)$$

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).□

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 for a solution of (P_2) with 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 problems (P_1) and (P_2). They will be used for analyzing the local convergence properties of algorithms (A_1) and (A_2).

Let $\mathbf{x}^* = \mathbb{1} \otimes x^*$ denote a local minimizer of (P_2) and let $\nabla \mathbf{h}(\mathbf{x}^*)$ denote the matrix

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

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

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

$$= \left[\underbrace{0,\ldots,0}_{n \text{ zeros}},\ldots,\underbrace{0,\ldots,0}_{n \text{ zeros}},\underbrace{\nabla h_i(x^*)'}_{i^{\text{th}} \text{ component}},\underbrace{0,\ldots,0}_{n \text{ zeros}},\ldots,\underbrace{0,\ldots,0}_{n \text{ zeros}}\right], \quad (12)$$

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

We now characterize the tangent cone at a local minimizer of (P_2).

Proposition 4.1: Let Assumptions 2.1-(a) and 2.2 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, $\mathbf{\Omega} = \{\mathbf{x} \mid \mathbf{h}(\mathbf{x}) = 0, \mathbf{L}\mathbf{x} = 0\}$. Then the tangent cone to $\mathbf{\Omega}$ at $\mathbf{x}^*$ is given by

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

$$= \{\mathbb{1} \otimes z \mid z \in \text{Null}\left(\nabla h(x^*)'\right) = \text{TC}(x^*, \Omega)\},$$

where $\Omega = \{x \mid h(x) = 0\}$ is the constraint set of (P_1), and $\text{TC}(x^*, \Omega)$ is the tangent cone to Ω at x^*.□

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 [20], 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 $\boldsymbol{\lambda}^* \in \mathbb{R}^{nN}$ so that

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

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

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. 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 and 2.2 hold and let $\mathbf{x}^* = \mathbb{1} \otimes x^*$ be a local minimizer 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 $\boldsymbol{\lambda} \in \{\boldsymbol{\lambda}^* + \boldsymbol{\lambda}_\perp \mid \boldsymbol{\lambda}_\perp \in \text{Null}(\mathbf{L}')\}$.□

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

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

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

Proposition 4.3: Let Assumptions 2.1 and 2.2 hold, let $\mathbf{x}^* = \mathbb{1} \otimes x^*$ be a local minimizer of (P_2) and let ψ^* and $\boldsymbol{\mu}^*$ be the unique Lagrange multiplier vectors corresponding to the first order necessary conditions of (P_1) and (P_2), respectively. Then $\psi^* = \boldsymbol{\mu}^*$.□

The convergence properties of algorithms (A_1) and (A_2) depend on the spectral properties of a particular matrix; properties analyzed in the following result.

Lemma 4.1: Let Assumptions 2.1 and 2.2 hold, let α be a positive scalar, and let $\mathbf{x}^*$ be a local minimizer of (P_2). Then the eigenvalues of the matrix

$$\mathbf{B} = \begin{pmatrix} \mathbf{H} & \nabla \mathbf{h}(\mathbf{x}^*) & \mathbf{L}' \\ -\nabla \mathbf{h}(\mathbf{x}^*)' & \mathbf{0} & \mathbf{0} \\ -\mathbf{L} & \mathbf{0} & \frac{1}{\alpha}\mathbf{J} \end{pmatrix},$$

have positive real parts, where $\mathbf{H}$ is a positive definite matrix and $\mathbf{J} \triangleq \frac{\mathbb{1}\mathbb{1}'}{\mathbb{1}'\mathbb{1}} \otimes I$.□

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 apply directly existing results for regular (local) minimizers, such as Proposition 4.4.2, page 388, [1]. Still, for a local minimizer and Lagrange multipliers pair $(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$, with $\boldsymbol{\lambda}^* \in \text{Range}(\mathbf{L})$, we show that if the initial values $(\mathbf{x}_0, \boldsymbol{\mu}_0, (\mathbf{I} - \mathbf{J})\boldsymbol{\lambda}_0)$ are close enough to $(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$, 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 $\{\boldsymbol{\lambda}^* + \text{Null}(\mathbf{L}')\}$.

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

$$\nabla \mathbf{F}(\mathbf{x}) + \mathbf{L}'\boldsymbol{\lambda} + \nabla \mathbf{h}(\mathbf{x})\boldsymbol{\mu} = 0, \quad (13)$$
$$\mathbf{h}(\mathbf{x}) = 0, \quad (14)$$
$$\mathbf{L}\mathbf{x} = 0. \quad (15)$$

Solving (13)-(15) does not guarantee finding a local minimizer, but at least the local minimizers are among the solutions of the above nonlinear system of equations. An approach for solving (13)-(15) consists of using a first order method (see for instance Section 4.4.1, page 386, [1]), which is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha\left[\nabla \mathbf{F}(\mathbf{x}_k) + \nabla \mathbf{h}(\mathbf{x}_k)\boldsymbol{\mu}_k + \mathbf{L}'\boldsymbol{\lambda}_k\right], \quad (16)$$
$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \alpha \mathbf{h}(\mathbf{x}_k), \quad (17)$$
$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha \mathbf{L}\mathbf{x}_k, \quad (18)$$

where $\alpha > 0$ is chosen to ensure the stability of the algorithm. By reformulating the above iteration in terms of the n-dimensional components of the vectors $\mathbf{x}_k$ and $\boldsymbol{\lambda}_k$, and

in terms of the scalar components of the vector $\boldsymbol{\mu}_k$, we recover Algorithm (A_1), which shows the *distributed* and *non-heuristic* nature of the algorithm.

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 5.1: Let Assumptions 2.1 and 2.2 hold and let $(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ with $\boldsymbol{\lambda}^* \in \text{Range}(\mathbf{L})$, be a local minimizer-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 exists $\bar{\alpha}$, such that for all $\alpha \in (0, \bar{\alpha}]$, the set $(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^* + \text{Null}(\mathbf{L}'))$ is an attractor of iteration (16)-(18) and if the sequence $\{\mathbf{x}_k, \boldsymbol{\mu}_k, \boldsymbol{\lambda}_k\}$ converges to the set $(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^* + \text{Null}(\mathbf{L}'))$, the rate of convergence of $\|\mathbf{x}_k - \mathbf{x}^*\|$, $\|\boldsymbol{\mu}_k - \boldsymbol{\mu}^*\|$ and $\|\boldsymbol{\lambda}_k - [\boldsymbol{\lambda}^* + \text{Null}(\mathbf{L}')]\|$ is linear.

Proof: Using the Lagrangian function defined in (11), iteration (16)-(18) can be equivalently expressed as

$$\begin{pmatrix} \mathbf{x}_{k+1} \\ \boldsymbol{\mu}_{k+1} \\ \boldsymbol{\lambda}_{k+1} \end{pmatrix} = \bar{\mathbf{M}}_\alpha(\mathbf{x}_k, \boldsymbol{\mu}_k, \boldsymbol{\lambda}_k), \tag{19}$$

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 $\mathbf{J} = \frac{\mathbb{1}\mathbb{1}'}{\mathbb{1}'\mathbb{1}} \otimes I$. 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 (19) 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 if $\left(\mathbf{x}_k, \boldsymbol{\mu}_k, \tilde{\boldsymbol{\lambda}}_k\right)$ converges to $(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$, we in fact show that $(\mathbf{x}_k, \boldsymbol{\mu}_k, \boldsymbol{\lambda}_k)$ converges to $(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^* + \text{Null}(\mathbf{L}'))$. The derivative of the mapping $\mathbf{M}_\alpha(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda})$ at $(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ is given by

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

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 a sphere $\mathcal{S}_\epsilon = \left\{(\mathbf{x}', \boldsymbol{\mu}', \boldsymbol{\lambda}')' \mid \|(\mathbf{x}', \boldsymbol{\mu}', \boldsymbol{\lambda}')' - (\mathbf{x}^{*\prime}, \boldsymbol{\mu}^{*\prime}, \boldsymbol{\lambda}^{*\prime})'\| < \epsilon\right\}$ for some $\epsilon > 0$ so that the induced norm of $\nabla \mathbf{M}_\alpha(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda})$ is less than one within the sphere $\mathcal{S}_\epsilon$. Therefore, using the mean value theorem, it follows that $\mathbf{M}_\alpha(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda})$ is a contraction map for any vector in the sphere $\mathcal{S}_\epsilon$. By invoking the contraction map theorem (see for example Chapter 7 of [6]) it follows that $\left(\mathbf{x}_k, \boldsymbol{\mu}_k, \tilde{\boldsymbol{\lambda}}_k\right)$ converges to $(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ for any initial value in S_ϵ. ■

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

Corollary 5.1: Let Assumptions 2.1 and 2.2 hold and let (x^*, ψ^*) be a local minimizer-Lagrange multiplier pair of (P_1). Assume also that $\nabla^2 f_i(x^*) + \psi_i^* \nabla^2 h_i(x^*)$ are positive definite for all $i = 1, \ldots, N$. Then there exits $\bar{\alpha}$, such that for all $\alpha \in (0, \bar{\alpha}]$, (x^*, ψ^*) is a point of attraction for iteration (2) and (3), for all $i = 1, \ldots, N$, and if the sequence $\{x_{i,k}, \mu_{i,k}\}$ converges to (x^*, ψ^*), then the rate of convergence of $\|x_{i,k} - x^*\|$ and $\|\mu_{i,k} - \psi^*\|$ is linear.

VI. Convergence analysis of Algorithm (A_2)

In the previous section we gave sufficient conditions for convergence to a local minimizer using Algorithm (A_1). By using an augmented Lagrangian on problem (P_2), we reformulate the first order necessary conditions for (P_2). Applying as in the case of Algorithm (A_1) a first order method to solve the reformulated first order necessary conditions, we obtain algorithm (A_2), for which local convergence can be proven under more relaxed conditions than in the case of Algorithm (A_1).

Let S be a matrix as in Remark 2.1 and let $\mathbf{S} = S \otimes I$. It follows that $\mathbf{L} = \mathbf{S}'\mathbf{S}$. We define the following augmented Lagrangian for problem (P_2):

$$\mathcal{L}_c(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = \mathbf{F}(\mathbf{x}) + \boldsymbol{\mu}'\mathbf{h}(\mathbf{x}) + \boldsymbol{\lambda}'\mathbf{L}\mathbf{x} + \frac{c}{2}\|\mathbf{h}(\mathbf{x})\|^2 + \frac{c}{2}\mathbf{x}'\mathbf{S}'\mathbf{S}\mathbf{x}, \tag{20}$$

where c is a positive scalar. The gradient and the Hessian of $\mathcal{L}_c(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda})$ are given by

$$\nabla_{\mathbf{x}} \mathcal{L}_c(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = \nabla \mathbf{F}(\mathbf{x}) + \nabla \mathbf{h}(\mathbf{x})\boldsymbol{\mu} + \mathbf{L}'\boldsymbol{\lambda} + c\nabla \mathbf{h}(\mathbf{x})\mathbf{h}(\mathbf{x}) + c\mathbf{L}\mathbf{x}, \tag{21}$$

and

$$\nabla^2_{\mathbf{xx}} \mathcal{L}_c(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = \nabla^2 \mathbf{F}(\mathbf{x}) + \sum_{i=1}^N \boldsymbol{\mu}_i \nabla^2 \mathbf{h}_i(\mathbf{x}) + c\mathbf{L} +$$

$$+ c \sum_{i=1}^N \left(\mathbf{h}_i(\mathbf{x}) \nabla^2 \mathbf{h}_i(\mathbf{x}) + \nabla \mathbf{h}_i(\mathbf{x}) \nabla \mathbf{h}_i(\mathbf{x})'\right), \tag{22}$$

respectively. Note that in the additional quadratic cost we do not include $\mathbf{x}'\mathbf{L}'\mathbf{L}\mathbf{x}$ but $\mathbf{x}'\mathbf{L}\mathbf{x}$, since doing so would prevent the distributed implementation of the first order method. It turns out that the introduction of $\mathbf{x}'\mathbf{L}\mathbf{x}$ is enough to obtain the desired behavior of the Hessian $\nabla^2_{\mathbf{xx}} \mathcal{L}_c(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$.

The first order necessary conditions for (P_2), reformulated in terms of $\mathcal{L}_c(\mathbf{x},\boldsymbol{\mu},\boldsymbol{\lambda})$ become

$$\nabla\mathbf{F}(\mathbf{x})+\nabla\mathbf{h}(\mathbf{x})\boldsymbol{\mu}+\mathbf{L}'\boldsymbol{\lambda}+c\nabla\mathbf{h}(\mathbf{x})\mathbf{h}(\mathbf{x})+c\mathbf{L}\mathbf{x}=0, \quad (23)$$

$$\nabla\mathbf{h}(\mathbf{x})=0, \quad (24)$$

$$\mathbf{L}\mathbf{x}=0. \quad (25)$$

A first order numerical method that can be used to solve the necessary conditions (23)-(25) takes the form

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k-\alpha\left[\nabla\mathbf{F}(\mathbf{x}_k)+\nabla\mathbf{h}(\mathbf{x}_k)\boldsymbol{\mu}_k+\right.\\ &+ \left. c\nabla\mathbf{h}(\mathbf{x}_k)\mathbf{h}(\mathbf{x}_k)+\mathbf{L}'\boldsymbol{\lambda}_k+c\mathbf{L}\mathbf{x}_k\right], \quad (26)\\ \boldsymbol{\mu}_{k+1} &= \boldsymbol{\mu}_k+\alpha\mathbf{h}(\mathbf{x}_k), \quad (27)\\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k+\alpha\mathbf{L}\mathbf{x}_k, \quad (28)\end{aligned}$$

which is basically the compact representation of Algorithm (A_2).

The following result addresses the local convergence properties of the iteration (26)-(28).

Theorem 6.1: Let Assumptions 2.1 and 2.2 hold and let $(\mathbf{x}^*,\boldsymbol{\mu}^*,\boldsymbol{\lambda}^*)$ with $\boldsymbol{\lambda}^*\in\text{Range}(\mathbf{L})$, be a local minimizer-Lagrange multipliers pair of (P_2). Assume also that $\mathbf{x}'\nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^*,\boldsymbol{\mu}^*,\boldsymbol{\lambda}^*)\mathbf{x}>0$ for all $\mathbf{x}\in\text{TC}(\mathbf{x}^*,\boldsymbol{\Omega})$. Then there exists $\bar{c}>0$ so that for all $c>\bar{c}$ we can find $\bar{\alpha}(c)$ such that for all $\alpha\in(0,\bar{\alpha}(c)]$, the set $(\mathbf{x}^*,\boldsymbol{\mu}^*,\boldsymbol{\lambda}^*+\text{Null}(\mathbf{L}'))$ is an attractor of iteration (26)-(28). In addition, if the sequence $\{\mathbf{x}_k,\boldsymbol{\mu}_k,\boldsymbol{\lambda}_k\}$ converges to the set $(\mathbf{x}^*,\boldsymbol{\mu}^*,\boldsymbol{\lambda}^*+\text{Null}(\mathbf{L}'))$, the rate of convergence of $\|\mathbf{x}_k-\mathbf{x}^*\|$, $\|\boldsymbol{\mu}_k-\boldsymbol{\mu}^*\|$ and $\|\boldsymbol{\lambda}_k-[\boldsymbol{\lambda}^*+\text{Null}(\mathbf{L}')]\|$ is linear.

The following corollary gives conditions that ensure local convergence to a local minimizer of (P_1) for each agent following Algorithm (A_2).

Corollary 6.1: Let Assumptions 2.1 and 2.2 hold and let (x^*,ψ^*) be a local minimizer-Lagrange multiplier pair of (P_1). Assume also that $x'\left[\nabla^2 f_i(x^*)+\psi_i^*\nabla^2 h_i(x^*)\right]x>0$ for all $x\in\text{TC}(x^*,\Omega)$. Then there exists $\bar{c}>0$ so that for all $c\geq\bar{c}$ we can find $\bar{\alpha}(c)$ such that for all $\alpha\in(0,\bar{\alpha}(c)]$, (x^*,ψ^*) is a point of attraction for iteration (5)-(7), for all $i=1,\ldots,N$. In addition, if the sequence $\{x_{i,k},\mu_{i,k}\}$ converges to (x^*,ψ^*), then the rate of convergence of $\|x_{i,k}-x^*\|$ and $\|\mu_{i,k}-\psi^*\|$ is linear.

Remark 6.1: Algorithms (A_1) and (A_2) are part of the general class of methods, called Lagrangian methods (see for example Section 4.4.1, page 386, [1]). They are based on a first order method, and therefore they achieve a linear rate of convergence. The optimization literature includes more sophisticated methods for solving constrained optimization problems, such as the multipliers methods or the sequential quadratic programming methods (see for instance [2], [3]); methods that can achieve superior convergence rates. They are based on a sequence of unconstrained minimization problems that in our case would need to be solved at each iteration in a distributed manner. In other words, unlike Algorithms (A_1) and (A_2), they have two layers of iterations: one layer is used to compute the estimate of the minimizer while the second one is used to update the Lagrange multipliers. Still, using the approach introduced in this paper (that is, minimizing the lifted constrained optimization problem), it turns out that each of the unconstrained optimization problems can be solved in a distributed manner. This is mainly due to the separability of the augmented Lagrangian (20). We are currently working on extending the augmented Lagrangian method so that it can be implemented in a distributed manner. The extension of the method is based on dealing with the fact that the Lagrange multipliers corresponding to the equality constraint $\mathbf{L}\mathbf{x}=0$ are not unique and therefore, the (local) minimizers are not regular.

References

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

[2] D.P. Bertsekas. *Cosntrained Optimization and Langrange Multipliers Methods*. Athena Scientific, Belmont, MA, 1982.

[3] B.F. Bonnans, J.C. Gilbert, C. Lemarechal, and C.A. Sagastizbal. *Numerical Optimization: Theoretical and Practical Aspects*. Springer, 2006.

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

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] I. Matei and J.S. Baras. Distributed algorithms for optimization problems with equality constraints. Technical Report 2013-05, Institute for Systems Research, February 2013. `http://hdl.handle.net/1903/13693`.

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

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

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

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