53rd IEEE Conference on Decision and Control
December 15-17, 2014. Los Angeles, California, USA

An Extension of the Method of Multipliers for Distributed Nonlinear Programming

Ion Matei, John S. Baras, Marzieh Nabi and Tolga Kurtoglu

Abstract— **In this paper we consider a distributed optimization problem, where a set of agents interacting through a communication graph have as common goal the minimization of a function expressed as a sum of (possibly non-convex) differentiable functions. Each function in the sum corresponds to an agent and each agent has associated an equality constraint. In this paper we investigate how the standard method of multipliers can be used to solve an optimization problem with equality constraints in a distributed manner. The method of multipliers is applied to a lifted optimization problem whose solution embeds the solution of the original problem. We modify the standard convergence results to deal with the fact the (local) minimizers of the lifted optimization problem are not regular, as a results of the distributed formulation.**

I. Introduction

Multi-agent, distributed optimization algorithms received a lot of attention in the recent years due to their applications in network resource allocation, collaborative control, estimation and identification problems. In these type of problems 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 distributed optimization algorithm was introduced in [16], where the convex 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 undirected graph. The algorithm combines a standard (sub)gradient descent step with a consensus step; the latter being added to deal with the limited information about the cost function and about the actions of the agents. Extensions of this initial version followed in the literature. [14], [17] include communication noise and errors on subgradients, [10], [12] assume a random communication graph, [14], [19] study asynchronous versions of the algorithm, [11] considers state-dependent communication topologies, while [4] assumes directed communication graphs. Another modification of the algorithm described in [16] was introduced in [8], where the authors change the order in which the consensus-step and the subgradient descent step are executed. Consensus-based distributed optimization algorithms were further used to solve constrained convex optimization problems where all agents have the same constraint set [9], [14], [17] or where each agent has its own set of constraints [15], [19]. Other approaches for obtaining distributed algorithms use dual decomposition [20], augmented Lagragian [6], [7], or in particular, distributed versions of the Alternating Direction Method of Multipliers (ADMM) algorithm [3], [18], [22].

In this paper we study as well a distributed optimization problem whose goal is to minimize an objective function expressed as a sum of functions. Each function in the sum is associated to an agent that has assigned an equality constraint, as well. We propose a distributed algorithm based on the method of multipliers applied to a lifted constrained optimization problem; a problem whose solution embeds the solution of the original optimization problem. This method is based on solving a sequence of unconstrained minimization subproblems, that have to be solved in a distributed manner. We show that the solutions of each of these problems can be obtained in a distributed manner, using a first order method. The main message of this paper is that standard optimization techniques can be used to solve optimization problems in a distributed manner, provided appropriate changes in the convergence proofs are made to deal with the fact that the standard assumptions no longer hold as a result of lack of complete information. We make no convexity assumptions on the cost and constraint functions, but we assume they are continuously differentiable. Consequently, our convergence results are local. Distributed algorithms for solving constrained, non-convex optimization problems were also proposed in [13] and [23]. In [13] the solution of the first-order necessary conditions is obtained using a first-order numerical algorithm, while in [23] an approximate dual subgradient algorithm is used to solve a non-convex optimization problem with inequality constraints.

The paper is organized as follows: in Section II we formulate the constrained optimization problem and introduce a distributed optimization algorithm 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. Section IV introduces a set of results used for the convergence analysis of the algorithm; analysis detailed in Sections V and VI.

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

Ion Matei, Marzieh Nabi and Tolga Kurtoglu are with the System Sciences Laboratory at Palo Alto Research Center (PARC), Palo Alto, CA (emails: ion.matei@parc.com, marzieh.nabi@parc.com, tolga.kurtoglu@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 and by grants NSF CNS-1035655 and NIST 70NANB11H148 to the University of Maryland.

(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 $\{A_i\}_{i=1}^N$ be a set of matrices. By diag($A_i,\ i=1,\ldots,N$) we understand a block diagonal matrix, where the i^{th} block matrix is given by A_i. By $\mathcal{S}(x;\varepsilon)$ we refer to a neighborhood around x, of radius ε. 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.

II. Problem description

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

A. Communication model

A set of N agents interact with each other through a communication topology modeled as an undirected communication graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$, where $\mathcal{V}=\{1,2,\ldots,N\}$ 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 assume that at each time instant k the agents can synchronously exchange information with their neighbors. We denote by $\mathcal{N}_i \triangleq \{j \mid e_{ij}\in\mathcal{E}\}$ the set of neighbors of agent i. Consider the set of pairs $\{(i,j), j\in\mathcal{N}_i,\ i=1,\ldots,N\}$ and let $\bar{N}=\sum_{i=1}^N|\mathcal{N}_i|$, where $|\cdot|$ denotes the cardinality of a set. We denote by $S\in\mathbb{R}^{\bar{N}\times N}$ the (weighted) edge-node incidence matrix of graph $\mathcal{G}$, for which each row number corresponds to a unique pair (i,j) from the previously defined set. The matrix S is defined as

$$[S]_{(ij),l} = \begin{cases} s_{ij} & i=l, \\ -s_{ij} & j=l, \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

where s_{ij} are given positive scalars.

Remark 2.1: It is not difficult to observe that the matrix $L=S'S$ is a (weighted) Laplacian matrix corresponding to the graph $\mathcal{G}$ and Null(L) = Null(S).□

In the next sections we are going to make use of a set of properties of the matrix S; properties that are grouped in the following remark.

Proposition 2.1: The matrix S defined with respect to a connected graph $\mathcal{G}$ satisfies the following properties:

(a) The nullspace of S is given by Null(S) = $\{\gamma\mathbb{1} \mid \gamma\in\mathbb{R}\}$;
(b) Let $\mathbf{S}=S\otimes I$, where I is the n-dimensional identity matrix. Then the nullspace of $\mathbf{S}$ is given by Null($\mathbf{S}$) = $\{\mathbb{1}\otimes x \mid x\in\mathbb{R}^n\}$.□

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 $h_i:\mathbb{R}^n\to\mathbb{R}$ and $N\le n$.

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 $h_i(x)$, $i=1,\ldots,N$ are twice continuously differentiable;
(b) Agent i has knowledge of only functions $f_i(x)$ and $h_i(x)$, and scalars s_{ij}, for $j\in\mathcal{N}_i$;
(c) Agent i can exchange information only with agents in the set of neighbors defined by $\mathcal{N}_i$;
(d) The communication graph $\mathcal{G}$ is connected.

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

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

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 $\nabla h(x^*)\triangleq[\nabla h_1(x^*),\nabla h_2(x^*),\ldots,\nabla h_N(x^*)]$ 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 typically used to prove local convergence for the "original" method of multipliers applied to Problem (P_1) (see for example Section 2.2, page 104 of [2]). As we will see in the next sections, the same assumption will be used to prove local convergence for a distributed version of the method of multipliers used to solved a 'lifted' optimization problem with equality constrains.

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 $m\le N$ agents have equality constraints.

C. Distributed algorithm

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. In addition, let us denote by $\mathbf{x}_k\in\mathbb{R}^{nN}$, $\boldsymbol{\mu}_k\in\mathbb{R}^N$ and $\boldsymbol{\lambda}_k\in\mathbb{R}^{n\bar{N}}$ the vectors $\mathbf{x}_k=(x_{i,k})$, $\boldsymbol{\mu}_k=(\mu_{i,k})$ and $\boldsymbol{\lambda}_k=(\lambda_{i,k})$, with $\lambda_{i,k}=(\lambda_{ij,k})$ for all $j\in\mathcal{N}_i$. We propose the following distributed algorithm to solve (P_1), denoted henceforth Algorithm (A_1), which is the algorithm behind the method of multipliers for solving an optimization problem equivalent to (P_1) :

$$\begin{aligned} \mathbf{x}_k \;=\; & \arg\min_{\mathbf{x}} \sum_i f_i(x_i)+\mu_{i,k}h_i(x_i)+ \\ & +\sum_{j\in\mathcal{N}_i}\lambda_{ij,k}s_{ij}(x_i-x_j)+\frac{c_k}{2}h_i(x_i)^2+ \\ & +\frac{c_k}{2}\sum_{j\in\mathcal{N}_i}(s_{ij}^2+s_{ji}^2)x_i(x_i-x_j),\ \mathbf{x}_0=\mathbf{x}^0, \end{aligned} \tag{2}$$

$$\mu_{i,k+1} \;=\; \mu_{i,k}+c_k h_i(x_{i,k}),\ \mu_{i,0}=\mu_i^0,\ i=1,\ldots,N, \tag{3}$$

$$\lambda_{ij,k+1} \;=\; \lambda_{ij,k}+c_k s_{ij}(x_{i,k}-x_{j,k}),\ \lambda_{ij,0}=\lambda_{ij}^0,\ j\in\mathcal{N}_i, \tag{4}$$

where, $\mathbf{x}^0$, μ_i^0 and λ_{ij}^0 are given initial conditions, $\{c_k\}$ is an non-decreasing sequence of positive numbers known by

all agents, and s_{ij} are positive scalars chosen by the agents. At each time instant k, we need to solve the unconstrained optimization problem (2). Note that in Algorithm (A_1) iterations (3)-(4) can be implemented in a distributed manner, since they use only local information, that is, μ_i, h_i, x_i, s_{ij} and λ_{ij}, and information from neighbors, that is, x_j. For Algorithm (A_1) to be distributed we need to provide a distributed algorithm that solves (2). Due to the structure of the cost function in (2), such an algorithm results from using a gradient-descent method, namely

$$\begin{aligned} x_{i,\tau+1} &= x_{i,\tau} - \alpha_\tau \Big[\nabla f_i(x_{i,\tau}) + \mu_{i,k} \nabla h_i(x_{i,\tau}) + \\ &+ \sum_{j \in \mathcal{N}_i} (s_{ij}\lambda_{ij,k} - s_{ji}\lambda_{ji,k}) + c_k \nabla h_i(x_{i,\tau}) h_i(x_{i,\tau}) \\ &+ c_k \sum_{j \in \mathcal{N}_i} (s_{ij}^2 + s_{ji}^2)(x_{i,\tau} - x_{j,\tau}) \Big], \ i = 1, \ldots, N \end{aligned} \tag{5}$$

with $x_{i,0} = (\mathbf{x}_k)_i$, and where $\{\alpha_\tau\}$ is a globally known sequence of step-sizes for the iteration (5). Note that we denote by τ the iteration counter for the algorithm used to solve (2). In addition, excluding the globally known sequence of stepsizes $\{\alpha_\tau\}$, each agent uses only local information, that is, x_i, $\nabla f_i(x_i)$, $\nabla h_i(x_i)$ s_{ij}, λ_{ij}, μ_i, and information from neighbors, that is, x_j, s_{ji} and λ_{ji}, for $j \in \mathcal{N}_i$. Consequently, Algorithm $(A1)$ is indeed *distributed*. We would like to emphasize that unless the communication topology is assumed *undirected*, the last sum in (5) cannot be implemented in a distributed manner since an agent would require information from agents not in its neighborhood.

III. An equivalent optimization problem with equality constraints

In this section we define a lifted optimization problem, from whose solution we can in fact extract the solution of problem (P_1). As made clear in what follows, Algorithm (A_1) comes as a result of applying the method of multipliers on the lifted 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 $g_i : \mathbb{R}^{nN} \to \mathbb{R}^{|\mathcal{N}_i| n}$ given by $g_i(\mathbf{x}) = (g_{ij}(\mathbf{x}))$, where $g_{ij}(\mathbf{x}) = s_{ij}(x_i - x_j)$, with s_{ij} positive scalars. The vector valued function $\mathbf{g}(\mathbf{x})$ can be compactly expressed as $g(\mathbf{x}) = \mathbf{S}\mathbf{x}$, where $\mathbf{S} = S \otimes I$, with I the n-dimensional identity matrix and S defined in (1). We introduce the optimization problem

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

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

$$\mathbf{g}(\mathbf{x}) = \mathbf{S}\mathbf{x} = 0. \tag{8}$$

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

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}) \triangleq \mathbf{F}(\mathbf{x}) + \boldsymbol{\mu}' \mathbf{h}(\mathbf{x}) + \boldsymbol{\lambda}' \mathbf{S}\mathbf{x}. \tag{9}$$

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 $\mathbf{S}$ is much richer than $\{\mathbb{1} \otimes x \mid x \in \mathbb{R}^n\}$, and therefore the solution of (P_2) may not necessarily be of the form $\mathbf{x}^* = \mathbb{1} \otimes x^*$. However, the fact that we search a solution of (P_2) of this particular structure is *fundamental* for showing the equivalence between the two optimization problems.

IV. Characterization of the solution of Problem (P_2)

In this section we introduce a number of results concerning the optimization problems (P_1) and (P_2). They will be used for analyzing the local converging properties of algorithm (A_1).

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

Proposition 4.1: Let Assumptions 2.1-(a) and 2.2 hold, let x^* be a local minimizer 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 $\mathrm{TC}(x^*, \Omega) = \mathrm{Null}(\nabla h(x^*)')$, where $\nabla h(x^*) \triangleq [\nabla h_1(x^*), \nabla h_2(x^*), \ldots, \nabla h_N(x^*)]$.□

Let $\mathbf{x}^* = \mathbb{1} \otimes x^*$ denote a local minimizer of (P_2) and let $\nabla \mathbf{h}(\mathbf{x}^*)$ denote the matrix $\nabla \mathbf{h}(\mathbf{x}^*) \triangleq [\nabla \mathbf{h}_1(\mathbf{x}^*), \nabla \mathbf{h}_2(\mathbf{x}^*), \ldots, \nabla \mathbf{h}_N(\mathbf{x}^*)]$. 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

$$\nabla \mathbf{h}_i(\mathbf{x}^*)' = \Big[\underbrace{0, \ldots, 0}_{n \text{ zeros}}, \ldots, \underbrace{0, \ldots, 0}_{n \text{ zeros}}, \ \underbrace{\nabla h_i(x^*)'}_{i^{\text{th}} \text{ component}}, \underbrace{0, \ldots, 0}_{n \text{ zeros}}, \ldots, \underbrace{0, \ldots, 0}_{n \text{ zeros}} \Big], \tag{10}$$

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

The second result of this section is concerned with the nullspace of the matrix $[\nabla \mathbf{h}(\mathbf{x}^*), \mathbf{S}']$, which will be used to characterize the tangent cone at a local minimizer of (P_2).

Proposition 4.2: Let Assumptions 2.1 and 2.2 hold. The nullspace of the matrix $[\nabla \mathbf{h}(\mathbf{x}^*), \mathbf{S}']$ is given by $\mathrm{Null}([\nabla \mathbf{h}(\mathbf{x}^*), \mathbf{S}']) = \{(\mathbf{0}', \mathbf{v}')' \mid \mathbf{v} \in \mathrm{Null}(\mathbf{S}')\}$.□

We now have all the machinery necessary to characterize the tangent cone at a local minimizer of (P_2).

Proposition 4.3 ([13]): 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 $\boldsymbol{\Omega}$ denote the constraint set, that is, $\boldsymbol{\Omega} = \{\mathbf{x} \mid \mathbf{h}(\mathbf{x}) = 0, \mathbf{S}\mathbf{x} = 0\}$. Then the tangent cone to $\boldsymbol{\Omega}$ at $\mathbf{x}^*$ is given by $\mathrm{TC}(\mathbf{x}^*, \boldsymbol{\Omega}) = \mathrm{Null}([\nabla \mathbf{h}(\mathbf{x}^*), \mathbf{S}']') = \{\mathbb{1} \otimes h \mid h \in \mathrm{Null}(\nabla h(x^*)') = \mathrm{TC}(x^*, \Omega)\}$.□

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 [21], 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 $\boldsymbol{\lambda}^* \in \mathbb{R}^{n\bar{N}}$ so that $\lambda_0^* \nabla \mathbf{F}(\mathbf{x}^*) + \nabla \mathbf{h}(x^*) \boldsymbol{\mu}^* + \mathbf{S}' \boldsymbol{\lambda}^* = 0$.

Note that since $\mathbf{S}$ is not full rank, and therefore the matrix $[\nabla \mathbf{h}(\mathbf{x}^*), \mathbf{S}']$ is not full rank either, 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 4.4 (first order necessary conditions [13]): 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 $\boldsymbol{\lambda}^* \in \text{Range}(\mathbf{S})$ so that $\nabla \mathbf{F}(\mathbf{x}^*) + \nabla \mathbf{h}(\mathbf{x}^*)\boldsymbol{\mu}^* + \mathbf{S}'\boldsymbol{\lambda} = 0$ for all $\boldsymbol{\lambda} \in \{\boldsymbol{\lambda}^* + \boldsymbol{\lambda}_\perp \mid \boldsymbol{\lambda}_\perp \in \text{Null}(\mathbf{S}')\}$. □

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 shown in the following, the two vectors are in fact equal.

Proposition 4.5 ([13]): 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 $\psi^* = \boldsymbol{\mu}^*$.□

V. Specialized results for the method of multipliers applied to (P_2)

In this section we prove a set of results that will be used to give conditions under which Algorithm (A_1) converges to a local minimizer of (P_1). The results are modifications of standard results concerning the method of multipliers (see for example Section 2.2, [2]). In the standard case, a regularity assumption on the minimizers is used to prove convergence of the method of multipliers. In our setup, this is not the case anymore and therefore, the standard results need to be modified accordingly.

The augmented Lagrangian function of problem (P_2) is given by

$$\mathcal{L}_c(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}) \triangleq \mathbf{F}(\mathbf{x}) + \boldsymbol{\mu}'\mathbf{h}(\mathbf{x}) + \boldsymbol{\lambda}'\mathbf{S}\mathbf{x} + \frac{c}{2}\|\mathbf{h}(\mathbf{x})\|^2 + \frac{c}{2}\mathbf{x}'\mathbf{S}'\mathbf{S}\mathbf{x}. \quad (11)$$

The gradient and the Hessian of $\mathcal{L}_c(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda})$ with respect to $\mathbf{x}$ are given by

$$\nabla_{\mathbf{x}}\mathcal{L}_c(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = \nabla \mathbf{F}(\mathbf{x}) + \nabla \mathbf{h}(\mathbf{x})\boldsymbol{\mu} + c\nabla \mathbf{h}(\mathbf{x})\mathbf{h}(\mathbf{x}) + \mathbf{S}'\boldsymbol{\lambda} + c\mathbf{S}'\mathbf{S}\mathbf{x}, \quad (12)$$

and

$$\nabla^2_{\mathbf{xx}}\mathcal{L}_c(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = \nabla^2 \mathbf{F}(\mathbf{x}) + \sum_{i=1}^{N} \mu_i \nabla^2 \mathbf{h}_i(\mathbf{x}) + \\ + c\sum_{i=1}^{N} \left(\mathbf{h}_i(\mathbf{x})\nabla^2 \mathbf{h}_i(\mathbf{x}) + \nabla \mathbf{h}_i(\mathbf{x})\nabla \mathbf{h}_i(\mathbf{x})'\right) + c\mathbf{S}'\mathbf{S}, \quad (13)$$

respectively. Note that the Hessian $\nabla^2_{\mathbf{xx}}\mathcal{L}_c$ does not depend on $\boldsymbol{\lambda}^*$. Convergence of many numerical optimization algorithms are based on the properties of the aforementioned Hessian, such as positive-definitiveness or invertibility. The next proposition, which appeared originally in [5], states that the Hessian $\nabla^2_{\mathbf{xx}}\mathcal{L}_c(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ can be made positive definite, provided the scalar c is chosen large enough.

Proposition 5.1 ([5]): Let $(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ be a local minimizer-Lagrange multipliers pair of (P_2) and assume that $\mathbf{z}'\nabla^2_{\mathbf{xx}}\mathcal{L}_0(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)\mathbf{z} > 0$ for all $\mathbf{z} \in \text{TC}(\mathbf{x}^*, \boldsymbol{\Omega})$. Then there exists a positive scalar $\bar{c}$, such that the Hessian $\nabla^2_{\mathbf{xx}}\mathcal{L}_c(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*) > 0$ for all $c \geq \bar{c}$.□

For notational simplicity we are going to group the Lagrange multipliers in one vector, that is, $\boldsymbol{\eta}' = [\boldsymbol{\mu}', \boldsymbol{\lambda}']$. When we mention a Lagrange multiplier vector $\boldsymbol{\eta}^*$ corresponding to a local minimizer $\mathbf{x}^*$, we will understand that its subcomponent $\boldsymbol{\lambda}^*$ is the unique vector in Range($\mathbf{S}$). In addition, let us group the equality constraints functions of (P_2) into one vector-valued function, that is, $\tilde{\mathbf{h}}(\mathbf{x})' = (\mathbf{h}(\mathbf{x})', \mathbf{x}'\mathbf{S}')$.

The convergence of the method of multipliers is based on the following result, which is an adaptation of the Proposition 2.4, page 108 of [2] so that it fits our setup.

Proposition 5.2: Let Assumptions 2.1 and 2.2 hold, let $(\mathbf{x}^*, \boldsymbol{\eta}^*)$, be a local minimizer-Lagrange multipliers pair of (P_2), and assume that $\mathbf{z}'\nabla^2_{\mathbf{xx}}\mathcal{L}_0(\mathbf{x}^*, \boldsymbol{\eta}^*)\mathbf{z} > 0$ for all $\mathbf{z} \in \text{TC}(\mathbf{x}^*, \boldsymbol{\Omega})$. In addition let $\bar{c}$ be a positive scalar such that $\nabla^2_{\mathbf{xx}}\mathcal{L}_{\bar{c}}(\mathbf{x}^*, \boldsymbol{\eta}^*) > 0$. There exists positive scalars $c_{\max}$, δ, ε and M such that:

(a) For all $(\boldsymbol{\eta}, c)$ in the set $D \subset \mathbb{R}^{N+n\bar{N}+1}$ defined by

$$D = \{(\boldsymbol{\eta}, c) \mid \|\mathbf{T}\boldsymbol{\eta} - \boldsymbol{\eta}^*\| < c\delta,\ \bar{c} \leq c \leq c_{\max}\}, \quad (14)$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - \mathbf{J} \end{bmatrix},$$

with $\mathbf{J}$ the orthogonal projection operator on Null($\mathbf{S}'$), the problem

$$\begin{aligned} \min \quad & \mathcal{L}_c(\mathbf{x}, \boldsymbol{\eta}) \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{S}(\mathbf{x}^*; \varepsilon) \end{aligned} \quad (15)$$

has a unique solution denoted by $\mathbf{x}(\boldsymbol{\eta}, c)$. The function $\mathbf{x}(\cdot, \cdot)$ is continuously differentiable in the interior of D, and for all $(\boldsymbol{\eta}, c) \in D$, we have

$$\|\mathbf{x}(\boldsymbol{\eta}, c) - \mathbf{x}^*\| \leq M\|\boldsymbol{\eta} - \boldsymbol{\eta}^*\|/c. \quad (16)$$

(b) For all $(\boldsymbol{\eta}, c) \in D$, we have

$$\|\tilde{\boldsymbol{\eta}}(\boldsymbol{\eta}, c) - \boldsymbol{\eta}^*\| \leq M\|\boldsymbol{\eta} - \boldsymbol{\eta}^*\|/c, \quad (17)$$

where

$$\tilde{\boldsymbol{\eta}}(\boldsymbol{\eta}, c) = \mathbf{T}\boldsymbol{\eta} + c\tilde{\mathbf{h}}[\mathbf{x}(\boldsymbol{\eta}, c)]. \quad (18)$$

(c) For all $(\boldsymbol{\eta}, c) \in D$, the matrix $\nabla^2_{\mathbf{xx}}\mathcal{L}_c[\mathbf{x}(\boldsymbol{\eta}, c), \boldsymbol{\eta}]$ is positive definite.□

The proof of this proposition is rather lengthy and is not included in the paper. Generally speaking, the proof follows the same lines as the proof of Proposition 2.4, [2]. However, since the local minimizer $\mathbf{x}^*$ is not regular, some changes in the statement of the proposition were needed compared to the original result, and consequently the proof had to be adapted accordingly. Compared to Proposition 2.4, page 108 of [2], our results have three main differences. The first difference consists of imposing an upper bound on c, namely $c_{\max}$. The reason we introduced $c_{\max}$ is to ensure that a certain Jacobian matrix that depends on c is invertible. Basically the entire proof of this proposition is based on the spectral properties of the Jacobian of the system of equations

$$\nabla \mathbf{F}(\mathbf{x}) + \nabla \tilde{\mathbf{h}}(\mathbf{x})\tilde{\boldsymbol{\eta}} = 0, \quad (19)$$

$$\tilde{\mathbf{h}}(\mathbf{x}) + t + \gamma(\boldsymbol{\eta}^* - \tilde{\boldsymbol{\eta}}) = 0, \quad (20)$$

with respect to $(\mathbf{x}, \tilde{\boldsymbol{\eta}})$ at the solution $(\mathbf{x}^*, \boldsymbol{\eta}^*)$, where $t = \frac{\mathbf{T}(\boldsymbol{\eta} - \boldsymbol{\eta}^*)}{c}$ and $\gamma = \frac{1}{c}$ (and consequently $\gamma \in [0, 1/\bar{c}]$). The Jacobian of (19)-(20) is given by

$$\begin{bmatrix} \nabla^2_{\mathbf{xx}}\mathcal{L}_0(\mathbf{x}^*, \boldsymbol{\eta}^*) & \nabla \tilde{\mathbf{h}}(\mathbf{x}^*) \\ \nabla \tilde{\mathbf{h}}(\mathbf{x}^*)' & \gamma \mathbf{I} \end{bmatrix}. \quad (21)$$

Using Proposition 5.1, it can be checked that for any $\gamma > 0$, the Jacobian defined in (21) is invertible. For $\gamma = 0$, however,

it can be checked that the nullspace of the matrix (21) is given by $\{(\mathbf{0}', \mathbf{0}', \mathbf{w}')' \mid \mathbf{w} \in \text{Null}(\mathbf{S}')\}$, and therefore, unlike in Proposition 2.4, [2], the Jacobian is no longer invertible at $\gamma = 0$. By choosing an arbitrarily large positive scalar $c_{\max}$ so that $\bar{c} \le c \le c_{\max}$, we in fact make sure that the matrix (21) is invertible for all considered values of c. Although finite, $c_{\max}$ can be made arbitrarily large. The price paid for this is the prevention of reaching a superlinear rate of convergence for Algorithm (A_1).

The second difference is the introduction of the operator $\mathbf{T}$. This operator acting on $\boldsymbol{\eta}$ ensures that $\tilde{\boldsymbol{\lambda}}(\boldsymbol{\eta}, c) \in \text{Range}(\mathbf{S})$ for all $(\boldsymbol{\eta}, c) \in D$, where $\tilde{\boldsymbol{\eta}} = (\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\lambda}})$. In defining the set D, the matrix $\mathbf{T}$ induces a neighborhood around $\boldsymbol{\eta}^*$ were only points $\boldsymbol{\eta} = (\boldsymbol{\mu}, \boldsymbol{\lambda})$, with $\boldsymbol{\lambda} \in \text{Range}(\mathbf{S})$ are considered. In this neighborhood, $\boldsymbol{\eta}^* = (\boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ is a unique Lagrange multiplier vector corresponding to the local minimizer $\mathbf{x}^*$.

The third difference is the definition of $\tilde{\boldsymbol{\eta}}$ in (18). Compared to the original statement[1] of Proposition 2.4, [2], we introduce the operator $\mathbf{T}$ that multiplies $\boldsymbol{\eta}$, to deal with the fact that $\mathbf{x}^*$ is not a regular local minimizer. As a consequence, we will have that $\tilde{\boldsymbol{\lambda}}(\boldsymbol{\eta}, c) \in \text{Range}(\mathbf{S})$ for all $(\boldsymbol{\eta}, c) \in D$, where $\tilde{\boldsymbol{\eta}} = (\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\lambda}})$.

Given a minimizer-Lagrange multiplier pair $(\mathbf{x}^*, \boldsymbol{\eta}^*)$ of (P_2), let us define the following matrix

$$\mathbf{M} = \left\{ \nabla \tilde{\mathbf{h}}(\mathbf{x}^*)' \left[\nabla^2_{\mathbf{xx}} \mathcal{L}_c(\mathbf{x}^*, \boldsymbol{\eta}^*) \right]^{-1} \nabla \tilde{\mathbf{h}}(\mathbf{x}^*) \right\}^{-1} - c\mathbf{I} \tag{22}$$

for any c for which $\nabla^2_{\mathbf{xx}} \mathcal{L}_c(\mathbf{x}^*, \boldsymbol{\eta}^*)$ is invertible. In addition, it can be shown that if $\left[\nabla^2_{\mathbf{xx}} \mathcal{L}_0(\mathbf{x}^*, \boldsymbol{\eta}^*) \right]^{-1}$ exist, then

$$\mathbf{M} = \left\{ \nabla \tilde{\mathbf{h}}(\mathbf{x}^*)' \left[\nabla^2_{\mathbf{xx}} \mathcal{L}_0(\mathbf{x}^*, \boldsymbol{\eta}^*) \right]^{-1} \nabla \tilde{\mathbf{h}}(\mathbf{x}^*) \right\}^{-1}, \tag{23}$$

respectively.

The following proposition characterizes the lower bound for c, that ensures that $\nabla^2_{\mathbf{xx}} \mathcal{L}_c(\mathbf{x}^*, \boldsymbol{\eta}^*)$ is positive definite.

Proposition 5.3 ([2]): Let Assumptions 2.1 and 2.2 hold, let $(\mathbf{x}^*, \boldsymbol{\eta}^*)$, be a local minimizer-Lagrange multipliers pair of (P_2) that satisfies $\mathbf{z}' \nabla^2_{\mathbf{xx}} \mathcal{L}_0(\mathbf{x}^*, \boldsymbol{\eta}^*) \mathbf{z} > 0$ for all $\mathbf{z} \in \text{TC}(\mathbf{x}^*, \boldsymbol{\Omega})$. For any scalar c, we have $\nabla^2_{\mathbf{xx}} \mathcal{L}_c(\mathbf{x}^*, \boldsymbol{\eta}^*) \succ 0$ if and only if $c > \max\{-\mathbf{e}_1, \ldots, -\mathbf{e}_{nN}\}$ (or equivalently $\mathbf{M} + c\mathbf{I} \succ 0$), where $\mathbf{e}_1, \ldots, \mathbf{e}_{nN}$ are the eigenvalues of $\mathbf{M}$.

The following result is the equivalent of Proposition 2.7 of [2]. Our result does not include the case of superlinear rate of convergence since we upper bound the scalar c and its statement is adapted so that it fits to the characteristics of Problem (P_2).

Proposition 5.4: Let Assumptions 2.1 and 2.2 hold, let $(\mathbf{x}^*, \boldsymbol{\eta}^*)$, with $\boldsymbol{\eta}^* = (\boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ be a local minimizer-Lagrange multipliers pair of (P_2) that satisfies $\mathbf{z}' \nabla^2_{\mathbf{xx}} \mathcal{L}_0(\mathbf{x}^*, \boldsymbol{\eta}^*) \mathbf{z} > 0$ for all $\mathbf{z} \in \text{TC}(\mathbf{x}^*, \boldsymbol{\Omega})$. In addition, let $\bar{c}$, δ and $c_{\max}$ be as in Proposition 5.2 with $\bar{c} > \max\{-2\mathbf{e}_1, \ldots, -2\mathbf{e}_{nN}\}$, where $\mathbf{e}_1, \ldots, \mathbf{e}_{nN}$ are the eigenvalues of $\mathbf{M}$ defined in (22). Then there exists δ_1 with $0 < \delta_1 \le \delta$ such that if $\{c_k\}$ and $\boldsymbol{\eta}_0$ satisfy

$$\|\mathbf{T}\boldsymbol{\eta}_0 - \boldsymbol{\eta}^*\| < \delta_1 c_0, \ \bar{c} \le c_k \le c_{k+1} \le c_{\max} \ \forall k, \tag{24}$$

then for the sequence $\{\boldsymbol{\eta}_k\}$ generated by

$$\boldsymbol{\eta}_{k+1} = \mathbf{T}\boldsymbol{\eta}_k + c_k \tilde{\mathbf{h}}[\mathbf{x}(\boldsymbol{\eta}_k, c_k)], \tag{25}$$

[1] $\boldsymbol{\eta}$ corresponds to λ in Proposition 2.4, [2].

we have $\boldsymbol{\eta}_k \to \boldsymbol{\eta}^*$ and $\mathbf{x}(\boldsymbol{\eta}_k, c_k) \to \mathbf{x}^*$. Furthermore if $\boldsymbol{\eta}_k \neq \boldsymbol{\eta}^*$ for all k, there holds

$$\limsup_{k \to \infty} \frac{\boldsymbol{\eta}_{k+1} - \boldsymbol{\eta}^*}{\boldsymbol{\eta}_k - \boldsymbol{\eta}^*} \le \max_{i=1\ldots nN} \left| \frac{\mathbf{e}_i}{\mathbf{e}_i + c_{\max}} \right|. \square \tag{26}$$

The previous proposition is the backbone of the convergence result for the Algorithm $(A1)$, as shown in the next section.

VI. Convergence analysis of Algorithm (A_1)

In this section we analyze the convergence properties of Algorithm (A_1). An inspection of the iteration (5), shows that it is a gradient descent method that can be used to find a local minimizer of the unconstrained optimization problem

$$\min_{\mathbf{x}} \mathcal{L}_c(\mathbf{x}, \boldsymbol{\mu}_k, \boldsymbol{\lambda}_k),$$

while the iteration (2)-(4) is the method of multipliers applied to Problem (P_2).

Algorithm (A_1) can be compactly written as

$$\mathbf{x}_k = \arg\min_{\mathbf{x}} \mathcal{L}_{c_k}(\mathbf{x}, \boldsymbol{\mu}_k, \boldsymbol{\lambda}_k), \ \mathbf{x}_0 = \mathbf{x}^0, \tag{27}$$

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + c_k \mathbf{h}(\mathbf{x}_k), \ \boldsymbol{\mu}_0 = \boldsymbol{\mu}^0, \tag{28}$$

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + c_k \mathbf{S}\mathbf{x}_k, \ \boldsymbol{\lambda}_0 = \boldsymbol{\lambda}^0. \tag{29}$$

Let $\mathbf{x}^* = \mathbb{1} \otimes x^*$ be a local minimizer of (P_2) with corresponding unique Lagrange multipliers vector $(\boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$, with $\boldsymbol{\lambda}^* \in \text{Range}(\mathbf{S})$ (consequently, $(x^*, \boldsymbol{\mu}^*)$ is a local minimizer, Lagrange multiplier pair of (P_1)).

In the following we show that under some conditions, Algorithm (A_1) ensures the convergence of each $x_{i,k}$ to x^* and the convergence of $\mu_{i,k}$ to μ_i^*, where $\boldsymbol{\mu}^* = (\mu_i^*)$. The algorithm does not guarantee convergence of $\lambda_{ij,k}$ to λ_{ij}^*, where $\boldsymbol{\lambda}^* = (\lambda_{ij}^*)$, for $j \in \mathcal{N}_i$ and $i = 1, \ldots, N$. In fact, if convergence is achieved, $\boldsymbol{\lambda}_k$ converges to the set $\boldsymbol{\lambda}^* + \text{Null}(\mathbf{S}')$. To simplify the convergence analysis, we make the following variable transformation: $\bar{\boldsymbol{\lambda}}_k = (\mathbf{I} - \mathbf{J})\boldsymbol{\lambda}_k$, where $\mathbf{J}$ is the projection operator on Null$(\mathbf{S}')$. Making the observation that $\mathbf{S}' = \mathbf{S}'(\mathbf{I} - \mathbf{J})$ and that $(\mathbf{I} - \mathbf{J})\bar{\boldsymbol{\lambda}}_k = (\mathbf{I} - \mathbf{J})\boldsymbol{\lambda}_k$, iterations (27)-(29) become

$$\mathbf{x}_k = \arg\min_{\mathbf{x}} \mathcal{L}_{c_k}(\mathbf{x}, \boldsymbol{\mu}_k, \bar{\boldsymbol{\lambda}}_k), \ \mathbf{x}_0 = \mathbf{x}^0, \tag{30}$$

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + c_k \mathbf{h}(\mathbf{x}_k), \ \boldsymbol{\mu}_0 = \boldsymbol{\mu}^0, \tag{31}$$

$$\bar{\boldsymbol{\lambda}}_{k+1} = (\mathbf{I} - \mathbf{J})\bar{\boldsymbol{\lambda}}_k + c_k \mathbf{S}\mathbf{x}_k, \ \bar{\boldsymbol{\lambda}}_0 = \bar{\boldsymbol{\lambda}}^0. \tag{32}$$

The following result gives conditions under which Algorithm (A_1) converges to a local minimizer x^* of Problem (P_1).

Corollary 6.1: Let Assumptions 2.1 and 2.2 hold, let (x^*, ψ^*) be a local minimizer-Lagrange multipliers pair of (P_1). In addition, let $\mathbf{x}^* = \mathbb{1} \otimes x^*$ be a local minimizer of (P_2) (as stated by Proposition 3.1), with unique Lagrange multipliers $(\boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ and $\boldsymbol{\lambda}^* \in \text{Range}(\mathbf{S})$. Assume also that $z' \left(\nabla^2 f_i(x^*) + \psi_i^* \nabla^2 h_i(x^*) \right) z > 0$ for all $z \in \text{TC}(x^*, \Omega)$ and let $\bar{c}$, $c_{\max}$ and δ be as in Proposition 5.4. Then there exists $0 < \delta_1 \le \delta$ so that if

$$\bar{c} > \max\{-2\mathbf{e}_1, \ldots, -2\mathbf{e}_{nN}\}, \tag{33}$$

where $\mathbf{e}_1, \ldots, \mathbf{e}_{nN}$ are the eigenvalues of $\mathbf{M}$ defined in (22), and the sequence $\{c_k\}$, $\boldsymbol{\mu}_0$ and $\boldsymbol{\lambda}_0$ satisfy

$$\left(\|\boldsymbol{\mu}_0 - \psi^*\|^2 + \|(\mathbf{I} - \mathbf{J})\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}^*\|^2 \right)^{1/2} < \delta_1 c_0, \tag{34}$$

$$\bar{c} \le c_k \le c_{k+1} \le c_{\max}, \tag{35}$$

then for the sequences $\{x_{i,k}\}$ and $\{\mu_{i,k}\}$ generated by the iteration (30)-(32), we have $x_{i,k} \to x^*$ and $\mu_{i,k} \to \psi_i^*$. Furthermore if $\mu_{i,k} \neq \psi_i^*$ for all k, the rate of convergence of $\{\mu_{i,k}\}$ is linear.

Proof: By Proposition 4.5 we have that $\boldsymbol{\mu}^* = \psi^*$. Using the definition of the Lagrangian function introduced in (9), we have $\nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*) = \text{diag}\left(\nabla^2 f_i(x^*) + \mu_i^* \nabla^2 h_i(x^*), i = 1, \ldots, N\right)$. In Proposition 4.3 we showed that $\text{TC}(\mathbf{x}^*, \boldsymbol{\Omega}) = \{\mathbb{1} \otimes h \mid h \in \text{TC}(x^*, \Omega)\}$, and therefore the assumption $x'\left[\nabla^2 f_i(x^*) + \mu_i^* \nabla^2 h_i(x^*)\right]x > 0$ for all $x \in \text{TC}(x^*, \Omega)$ is equivalent to $\mathbf{x}'\nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)\mathbf{x} > 0 \ \forall \mathbf{x} \in \text{TC}(\mathbf{x}^*, \boldsymbol{\Omega})$. Let δ_1 be as in Proposition 5.4. All assumptions of Proposition 5.4 are satisfied and the result follows. ∎

Remark 6.1: At each step of Algorithm (A_1), we use iteration (5) to obtain the solution of (2), and therefore the convergence is dependent on solving

$$\begin{aligned} \min \quad & \mathcal{L}_c(\mathbf{x}, \boldsymbol{\eta}_k) \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{S}(\mathbf{x}^*; \varepsilon). \end{aligned}$$

The solution of the above problem is well defined if it is "close enough" to the local minimizer $\mathbf{x}^*$. However, the unconstrained optimization problem may have multiple local minimizers. Thus for the algorithm to converge to the correct solution, $\mathbf{x}_k$ must remain in a neighborhood of the same local minimizer, at least after some time instant k. Practice showed that using $\mathbf{x}_k$ as starting point in (5) to compute $\mathbf{x}_{k+1}$ tends to ensure that the solutions of the unconstrained optimization problems remain in a neighborhood of the same local minimizer. In addition to starting closed enough from $\mathbf{x}^*$, appropriate step-sizes α_τ must be used so that (5) converges. A "sufficiently small" constant sequence $(\alpha_\tau = \alpha)$ or a slowly diminishing sequence $(\alpha_\tau \to 0, \sum_\tau \alpha_\tau = \infty)$ can be chosen. Conditions on the stepsize sequence that ensure convergence can be found in [1] (Propositions 1.2.3 and 1.2.4).□

Remark 6.2: Corollary 6.1 shows that we can use the method of multipliers to compute a local minimizer for the Problem (P_1). Note the change in the condition the initial values $\boldsymbol{\eta}_0$ must satisfy, compared to the original result, namely the projection of $\boldsymbol{\lambda}_0$ on Range($\mathbf{S}$). This change was necessary as a result of the lack of regularity of the local minimizer; lack of regularity that also prevented us from showing that Algorithm (A_1) can achieve a (theoretical) superlinear rate of convergence due to the upper-bound imposed on the sequence $\{c_k\}$.□

VII. Conclusions

We presented a distributed algorithm for solving an optimization problem with equality constrains; where 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. The algorithm was inspired by the methods of multipliers applied to a lifted optimization problem equivalent to the original one. We gave conditions for the (local) convergence of the algorithm and emphasized the changes needed for proving such results compared to the standard result, due to the distributed formulation. As a result of the loss of regularity of the local minimizers, we were not able to guarantee that the algorithm can reach superlinear rate of convergence, but can get arbitrarily close to it.

References

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

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

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.

[4] A. Cherukuri and J. Cortés. Distributed generator coordination for initialization and anytime optimization in economic dispatch. *IEEE Transactions on Control of Network Systems (Submitted)*, 2013.

[5] M.R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4:303–320, 1969.

[6] D. Jakovetic, M.F.J. Moura, and J. Xavier. Linear convergence rate of a class of distributed augmented lagrangian algorithms. *CoRR*, abs/1307.2482, 2013.

[7] D. Jakovetic, J. Xavier, and J.M.F. Moura. Cooperative convex optimization in networked systems: Augmented lagrangian algorithms with directed gossip communication. *Signal Processing, IEEE Transactions on*, 59(8):3889–3902, Aug.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

[13] I. Matei and J.S. Baras. Distributed algorithms for optimization problems with equality constraints. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 2352–2357, Dec 2013.

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

[15] 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.

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

[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] W. Shi, Q. Ling, K. Yuan, G Wu, and W. Yin. On the linear convergence of the admm in decentralized consensus optimization. *Signal Processing, IEEE Transactions on*, 62(7):1750–1761, April 2014.

[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] H. Terelius, U. Topcu, and R. M. Murray. Decentralized multi-agent optimization via dual decomposition. In *18th World Congress of the International Federation of Automatic Control (IFAC)*, August 2011.

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

[22] E. Wei and A. Ozdaglar. Distributed alternating direction method of multipliers. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 5445–5450, Dec 2012.

[23] M. Zhu and S. Martinez. An approximate dual subgradient algorithm for multi-agent non-convex optimization. *Automatic Control, IEEE Transactions on*, 58(6):1534–1539, June 2013.