Linear Convergence Rate of Class of Distributed Augmented Lagrangian Algorithms

Dušan Jakovetić, *Member, IEEE,* José M. F. Moura, *Fellow, IEEE,* and João Xavier, *Member, IEEE*

***Abstract*—We study distributed optimization where nodes cooperatively minimize the sum of their individual, locally known, convex costs $f_i(x)$'s, $x \in \mathbb{R}^d$ is global. Distributed augmented Lagrangian (AL) methods have good empirical performance on several signal processing and learning applications, but there is limited understanding of their convergence rates and how it depends on the underlying network. This paper establishes globally linear (geometric) convergence rates of a class of deterministic and randomized distributed AL methods, when the f_i's are twice continuously differentiable and have a bounded Hessian. We give explicit dependence of the convergence rates on the underlying network parameters. Simulations illustrate our analytical findings.**

Keywords: Distributed optimization, convergence rate, augmented Lagrangian, consensus.

I. Introduction

A. Motivation

We study distributed optimization over a N-node, connected, undirected network $\mathcal{G} = (\mathcal{V}, E)$, with $\mathcal{V}$ the set of nodes and E the set of edges. Node i has private cost function $f_i(x)$, $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$. We focus on iterative, distributed algorithms that solve the unconstrained problem:

$$\text{minimize} \quad f(x) := \textstyle\sum_{i=1}^N f_i(x), \tag{1}$$

while each node i communicates only with its neighbors. This is the setup in many applications, e.g., distributed inference, [1], or distributed source localization, [2], in sensor networks.

A popular approach to solve (1), e.g., [3], [4], [5], [6], [7], is through the augmented Lagrangian (AL) dual. The approach assigns a local copy $x_i \in \mathbb{R}^d$ of the global variable x in (1) to each node i, introduces the edge-wise constraints $\sqrt{W_{ij}}(x_i - x_j) = 0, \forall \{i,j\} \in E$ (W_{ij} a positive weight),[1] and forms an AL dual function by dualizing these constraints and adding the quadratic penalty $\frac{\rho}{2}\sum_{\{i,j\}\in E,\, i \leq j} W_{ij}\|x_i - x_j\|^2$, see, e.g., [8], Section V, for details[2]. Denote by $\lambda_{\{i,j\}} \in \mathbb{R}^d$ the dual variable that corresponds to the constraint on the edge $\{i,j\}$. Introducing the per-node aggregate dual variables $\mu_i := \sum_{j \in O_i} \sqrt{W_{ij}} \lambda_{\{i,j\}} \text{sign}(j-i)$ ($\text{sign}(0) := 1$), where O_i is the node i's neighborhood (including i), one obtains the following dual method to solve (1):

$$\begin{aligned} &(\, x_1(k+1), \cdots, x_N(k+1)\,) = \\ &\text{argmin}_{(x_1,\cdots,x_N)\in\mathbb{R}^{dN}} L_a\left(x_1, \cdots, x_N;\, \mu_1(k), \cdots, \mu_N(k)\right) \end{aligned} \tag{2}$$

$$\mu_i(k+1) = \mu_i(k) + \alpha \sum_{j \in O_i} W_{ij}\left(x_i(k+1) - x_j(k+1)\right), \tag{3}$$

where $\alpha > 0$ is the (dual) step-size, and $L_a : \mathbb{R}^{dN} \times \mathbb{R}^{dN} \rightarrow \mathbb{R}$, is the function:

$$\begin{aligned} L_a(x_1, \cdots, x_N; \mu_1, \cdots, \mu_N) &= \sum_{i=1}^N f_i(x_i) \\ + \sum_{i=1}^N \mu_i^\top x_i &+ \frac{\rho}{2} \sum_{\{i,j\}\in E,\, i\leq j} W_{ij}\, \|x_i - x_j\|^2. \end{aligned} \tag{4}$$

In (2) and (3), $x_i(k)$ and $\mu_i(k)$ are the node i's primal and dual variables, respectively. Dual updates (3) allow for distributed implementation, as each node i needs only the primal variables $x_j(k+1)$ from its immediate neighbors in the network. When $\rho = 0$, the primal update (2) decouples as well, and node i solves for $x_i(k+1)$ locally (without inter-neighbor communications.) When $\rho > 0$, the quadratic coupling term in (4) (in general) induces the need for inter-node communications to iteratively solve (2). Many known methods to solve (1) fall into the framework of (2)–(3); see, e.g., [9], [4], [5], [6], [7], [8]. These methods are used in various signal processing and learning applications, but, until recently, their convergence rates have not been analyzed.

B. Contributions

In this paper, we introduce an analytical framework to study the convergence rates of *distributed* AL methods of type (2)–(3) when problems (2) are solved inexactly. While the AL methods that we consider are variations on the existing methods, our analysis gives new results on the globally linear convergence rates of distributed AL algorithms and brings several important insights into the performance of distributed multi-agent optimization.

The work of the first and third authors was supported by: the Carnegie Mellon|Portugal Program under a grant from the Fundação de Ciência e Tecnologia (FCT) from Portugal; by FCT grants CMU-PT/SIA/0026/2009, FCT PTDC/EMS-CRO/2042/2012, and SFRH/BD/33518/2008 (through the Carnegie Mellon|Portugal Program managed by ICTI); by ISR/IST pluri-anual funding (POSC program, FEDER), and the work of the first and second authors was funded by AFOSR grant FA95501010291 and by NSF grant CCF1011903, while the first author was a doctoral and postdoctoral student within the Carnegie Mellon|Portugal Program. D. Jakovetić is with University of Novi Sad, BioSense Center, 21000 Novi Sad, Serbia. J. M. F. Moura is with Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA. J. Xavier is with Instituto de Sistemas e Robótica (ISR), Instituto Superior Técnico (IST), University of Lisbon, 1049-001 Lisbon, Portugal. Authors e-mails: djakovet@uns.ac.rs, moura@ece.cmu.edu, jxavier@isr.ist.utl.pt.

[1] We include also self-edges, i.e., $\{i,i\} \in E$, $\forall i$.

[2] Here, $\rho \geq 0$ is the penalty parameter and W_{ij} are the weights, collected in the $N \times N$ symmetric matrix W, where $W_{ij} > 0$ if $\{i,j\} \in E$, $i \neq j$, $W_{ii} := 1 - \sum_{j \neq i} W_{ij}$, and W is doubly stochastic.

Preprint submitted to IEEE Transactions on Automatic Control. Received: October 8, 2014 17:09:00 PST

We now explain our technical results. Let $x'(k+1) = (x'_1(k+1)^\top, ..., x'_N(k+1)^\top)^\top$ be the solution to (2) when the dual variables are fixed to $\mu(k) = (\mu_1^\top(k), ..., \mu_N^\top(k))$. Our framework handles arbitrary iterative method that solves (2), where the method's initial guess of $x'(k+1)$ (starting point) at iteration k is set to $x(k)$. Further, let $\|x(k+1) - x'(k+1)\| \leq \xi \, \|x(k) - x'(k+1)\|$, $\forall k$, $\xi \in (0,1)$, i.e., problem (2) is solved up to a certain accuracy such that the distance to the solution (in terms of Euclidean norm) is reduced ξ times with respect to the starting point $x(k)$. Assuming that the cost functions f_i's are twice continuously differentiable, with bounded Hessian ($h_{\min} I \preceq \nabla^2 f_i(x) \preceq h_{\max} I$, $\forall i$, $\forall x \in \mathbb{R}^d$, $h_{\min} > 0$), we give explicit conditions that relate the quantities ξ, $h_{\min}$, and $h_{\max}$, and the network's spectral gap $\lambda_2 = \lambda_2(\mathcal{L})$,[3] such that the distributed AL method converges to the solution of (1) at a *globally linear rate*. Furthermore, we explicitly characterize the achieved rate in terms of the above system parameters.

We apply and specialize our results to four iterative distributed AL methods that solve (1) that mutually differ in how (2) is solved. We refer to the four methods as: 1) deterministic Jacobi-type; 2) deterministic gradient; 3) randomized Gauss-Seidel-type; and 4) randomized gradient-type (see Section II for the algorithms' details.) We establish with all methods globally linear convergence rates in terms of the total number of per-node communications, and we explicitly characterize the rates in terms of the system parameters. Furthermore, with deterministic and randomized gradient variants, we establish the globally linear convergence rates in terms of the total number of per-node evaluations of gradients of the f_i's.

We now highlight several key contributions and implications of our results that distinguish our work from the existing literature on distributed multi-agent optimization.

1. We give a general framework to analyze distributed AL algorithms, and we establish linear convergence rates for a *wide class* of distributed AL methods. This contrasts with the existing work which typically studies a specific distributed method, like the distributed ADMM [10], [11]. In particular, this allows us to establish for the first time linear convergence rates of the distributed AL methods with *randomized* primal variable updates. We remark that, for certain specific methods, like the distributed ADMM, the literature gives tighter bounds than we do, as we explain below.

2. To our best knowledge, our results on deterministic and randomized *gradient* variants are the first that establish globally linear convergence rates for any distributed algorithm that solves (1), *simultaneously* in terms of per-node gradient evaluations and per-node communications.

3. We provide distributed methods (deterministic and randomized gradient variants) that involve only simple calculations (like the gradient-type methods in, e.g., [12]) but achieve significantly faster rates than [12]. That is, we show that through the AL mechanism much faster rates can be obtained compared with respect to standard distributed gradient methods [12], while maintaining the same communication cost and similar computational cost per iteration, and requiring additional knowledge on the system parameters. Namely, [13] (see also [14] for similar results) studies the method in [12] when the costs f_i's are strongly convex and have Lipschitz continuous gradients–the setup very similar to ours (We additionally require twice continuously differentiable costs.) Assuming that nodes know $h_{\max}$, it shows that the distance to the solution after k iterations is $O\left((1-\alpha\, c_2)^{k/2} + \frac{\alpha h_{\max}}{\lambda_2}\right)$, where α is the step-size and $c_2 = h_{\max} h_{\min}/(h_{\max} + h_{\min})$. From these results, it follows that, to achieve ϵ-accuracy, we need $O\left(\frac{\gamma \log(1/\epsilon)}{\epsilon\, \lambda_2}\right)$ per-node communications and per-node gradient evaluations, where $\gamma = h_{\max}/h_{\min}$ is the condition number. In contrast, we assume with our deterministic gradient that nodes know $\lambda_2, h_{\min}$, and $h_{\max}$, and we show that (ignoring terms logarithmic in N, λ_2, and γ) the ϵ-accuracy is achieved in $O\left(\frac{\gamma \log(1/\epsilon)}{\lambda_2}\right)$ per-node communications and per-node gradient evaluations.

C. Related work

We now further relate our work with the existing literature. We first consider the literature on distributed multi-agent optimization, and then we consider the work on the conventional, centralized optimization.

Distributed multi-agent optimization. Many relevant works on this and related subjects have recently appeared. Reference [15] considers (1) over generic networks as we do, under a wide class of generic convex functions. The reference shows $O(1/\mathcal{K})$ rate of convergence in the number of per-node communications for a distributed ADMM method. It is important to note that, differently from our paper, [15] considers generic costs for which even in a centralized setting linear rates are not achievable. Reference [16] considers both resource allocation problems and (1) and develops accelerated dual gradient methods which are different than our methods. It gives the methods' convergence factors as $1 - \Omega\left(\sqrt{\frac{\lambda_{\min}(AA^\top)}{\gamma\, \lambda_{\max}(AA^\top)}}\right)$, where A is the edge-node incidence matrix and $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the minimal non-zero and maximal eigenvalues, respectively.[4] The rates in [16] are better than the rates that we establish for our methods. Reference [16] assumes that each node exactly solves certain local optimization problems and is not concerned with establishing the rates in terms of the number of gradient evaluations ([16] corresponds to exact dual methods.) Another difference is that the methods in [16] are based on the ordinary dual–not AL dual. Reference [17] analyzes distributed ADMM for the consensus problem–the special case when $f_i : \mathbb{R} \to \mathbb{R}$ is $f_i(x) = (x - a_i)^2$, $a_i \in \mathbb{R}$. It establishes the global convergence factor $1 - \Omega(\sqrt{\lambda_2(\mathcal{L})})$. When we specialize our result to the problem studied in [17], their convergence factor bound is tighter than ours. Finally, references [10], [11] analyze a distributed ADMM method therein when the costs are strongly convex and have Lipschitz continuous gradients. The method in [10], [11] corresponds to our deterministic Jacobi-type variant when $\tau = 1$. With

[3] The spectral gap $\lambda_2(\mathcal{L})$ is the second smallest eigenvalue of the weighted Laplacian matrix $\mathcal{L} := I - W$.

[4] For two positive sequences η_n and χ_n, $\eta_n = \Omega(\chi_n)$ means that $\liminf_{n\to\infty} \frac{\eta_n}{\chi_n} > 0$.

respect to our results, the bounds in [10], [11] are tighter than ours for the method they study.

References [18], [19], [20], [21] study distributed primal-dual methods that resemble ours when the number of inner iterations τ is set to one (but their methods are not the same.) These works do not analyze the convergence rates of their algorithms.

Centralized optimization. Our work is also related to studies of the AL and related algorithms in conventional, centralized optimization. There is a vast literature on the subject, and many authors considered inexact primal minimizations (see [22], [23], [24] and the references listed in the following paragraphs.) Before detailing the existing work, we point to main differences of this paper with respect to usual studies in the literature. First, when analyzing inexact AL methods, the literature usually assumes that the primal problems use arbitrary initialization. In contrast, we initialize the inner primal algorithm with the previous primal variable. Consequently, our results and the results in the literature are different, the algorithms in the literature typically be convergent only to a solution neighborhood, e.g. [22], [23]. Second, except for recent papers, e.g., [22], [23], the analysis of inexact AL is usually done with respect to dual sub-optimality. In contrast, we are interested in the primal sub-optimality measures. Third, convergence rates are usually established at the outer iteration level, while we–besides the outer iterations level–establish the rates in the number of *inner* iterations.

In summary, we establish *primal sub-optimality* globally linear convergence rates in the number of *inner iterations* (overall number of iterations) for our AL methods; such studies seem not abundant in the literature.

We now detail the literature and divide it into four classes: 1) ADMM algorithms; 2) AL algorithms; 3) saddle point algorithms; and 4) Jacobi/Gauss-Seidel algorithms. We also point to several interesting connections among different methods.

ADMM algorithms. The ADMM method has been proposed in the 70s [25], [26] and has been since then extensively studied. References [27], [28], [29] show locally linear or superlinear convergence rates of AL methods. Reference [24] analyzes convergence of the ADMM method using the theory of maximal set monotone operators, and it studies its convergence under inexact primal minimizations. Recently, [30], [31] show that the ADMM method converges globally linearly, for certain more general convex costs than ours. (The most related work to ours on ADMM is actually the work on distributed ADMM in [10], [11] that we have already commented on above.)

AL algorithms. Lagrangian duality is classical and a powerful machinery in optimization; see, e.g. [32] for general theory, and, e.g., [33], for applications in combinatorial optimization and unit-commitment problems. The method of multipliers based on the augmented Lagrangian has been proposed in the late 60s [34], [35]. The convergence of the algorithm has been extensively studied, also under inexact primal minimizations. References [27], [28], [29] show *locally* linear or superlinear convergence rates of AL methods. The work [22] analyzes the inexact AL method when the primal and dual variables are updated using inexact fast gradient schemes. This paper finds the total number of the *inner* iterations needed to achieve an ϵ-accurate primal solution. Reference [23] studies AL dual standard and fast gradient methods when the primal problems are solved inexactly, up to a certain accuracy ϵ_{in}. The reference finds the number of outer iterations and the required accuracy ϵ_{in} to obtain an ϵ_{out}-suboptimal primal solution.

Saddle point algorithms. This thread of the literature considers iterative algorithms to solve saddle point problems. We divide the saddle point algorithms into two types. The first type of algorithms performs at each iteration only one gradient step with respect to the primal variables. The second type of algorithms solves at each iteration an optimization problem, like it is done with the AL method in (2). We now consider the first type of methods. A classical method dates back to the 50s [36]. Our distributed gradient AL, when the number of inner iterations is set to $\tau = 1$, is similar to this algorithm. Reference [36] analyzes stability of the method in continuous time, while [37], [38] analyzes the method's convergence under diminishing step-sizes. Different versions of the method are considered and analyzed in [39]. More recently, reference [40] studies similar algorithms for a wide class of non-differentiable (in general) cost functions and gives sub-linear rates to a neighborhood of a saddle point (The sub-linear rate is due to the wide function class assumed). In summary, although one of our algorithms generally falls into the framework of this class of methods, we could not find the results in the literature that are equivalent to ours.

We now focus on the second type of methods. The classical method is the Arrow-Hurwitz-Uzawa method (also known as Uzawa method), see, e.g., [41]. The algorithm has been thoroughly analyzed and several modifications have been proposed, e.g., [41], [42], [43], [44], [45]. In fact, our inexact distributed AL method is (an inexact version of) the Arrow-Hurwitz-Uzawa method, applied to a specific saddle point system (see ahead (19)–(21).) This in particular means that the AL algorithm on the dual of (1), given by (2)–(3), is analogous to the Arrow-Hurwitz-Uzawa method on a specific saddle point problem (19)–(21). Reference [43] analyzes an exact method therein and establishes its convergence rates. References [42], [45] analyze the inexact methods therein for linear saddle point problems (which corresponds to quadratic cost functions), while references [41], [44] analyze inexact methods therein for non-linear saddle point problems (which corresponds to more general cost functions.) Our analysis is in the spirit closest to this thread of works. Although (19)–(21) is similar to the classical setup, we could not find in the literature results equivalent to ours.

Jacobi/Gauss-Seidel algorithms. Our work is also related to studies of Jacobi/Gauss-Seidel algorithms, in the following sense. Certain distributed AL methods that we consider solve the *inner problems* (2) via iterative Gauss-Seidel/Jacobi algorithms. In other words, we employ the Jacobi/Gauss-Seidel methods at the inner iteration level. Jacobi and Gauss-Seidel methods have been studied for a long time, e.g., [46], [47], [48], [49], [50], [51], [52], [53]. The methods have been studied both in the synchronous updates setting, e.g., [46], [48], and in the asynchronous updates setting, e.g., [47], [48], [49], [50], [51], [52], [53], in more general setups than the setup that we consider. Reference [46] presents, e.g., global convergence for Jacobi and Gauss-Seidel methods (with cyclic order of variable updates) for solving nonlinear systems $F(x)=0$, $F:\mathbb{R}^N \mapsto \mathbb{R}^N$, where $F(x)=Ax+\phi(x)$, A is an M-matrix and ϕ is a diagonal, isotone mapping (see Theorems 13.1.3. and 13.1.5 in [46]). The cyclic Jacobi and Gauss-Seidel methods are known to converge at globally linear rates, when the gradient of the map F is a diagonally dominant (positive definite) matrix; see [48], Proposition 2.6. Reference [47] studies asynchronous multi-node[5] iterative methods including Gauss-Seidel and Jacobi, in the presence of bounded inter-node communication delays. It uses Lyapunov theory to establish global and local convergence (stability) of asynchronous iterative methods under various conditions. For example, it is shown that an asynchronous iterative scheme converges if the local nodes' update maps are block Lipschitz continuous, and if the corresponding matrix of Lipschitz constants is Schur-stable; see Theorem 4.4.4 in [47], other results in Chapter 4, and references therein. In contrast with the above existing results, convergence of Jacobi/Gauss-Seidel algorithms in general settings is not our main concern; instead, we are interested in the overall AL algorithm with Jacobi/Gauss-Seidel type inner algorithms. In contradistinction with the literature, we consider certain Gauss-Seidel and Jacobi-type methods for the special case of minimizing (4); exploiting this special structure, we derive *explicit convergence factors* of the updates. This allows us to explicitly determine the required number of inner (Jacobi/Gauss-Seidel-type) iterations τ that ensure linear convergence of the overall AL distributed schemes (See Theorem 1 and Lemmas 5–8 for details).

Paper organization. Section II details our network and optimization models and presents distributed AL methods. Section III presents our analytical framework for the analysis of inexact AL and proves the generic result on its convergence rate. Section IV specializes this result for the four considered distributed methods. Section V provides simulations with l_2-regularized logistic losses. Finally, we conclude in Section VI.

Notation. Denote by: $\mathbb{R}^d$ the d-dimensional real space; a_l the l-th entry of vector a; A_{lm} or $[A]_{lm}$ the (l,m) entry of A; $A^\top$ the transpose of A; $\otimes$ the Kronecker product of matrices; I, 0, 1, and e_i, respectively, the identity matrix, the zero matrix, the column vector with unit entries, and the i-th column of I; J the $N\times N$ ideal consensus matrix $J:=(1/N)1\,1^\top$; $\|\cdot\|_l$ the vector (respectively, matrix) l-norm of its vector (respectively, matrix) argument; $\|\cdot\|=\|\cdot\|_2$ the Euclidean (respectively, spectral) norm of its vector (respectively, matrix) argument; $\lambda_i(\cdot)$ the i-th smallest eigenvalue; $A\succ 0$ means A is positive definite; $\mathrm{diag}(a)$ the diagonal matrix with the diagonal equal to vector a; $\lfloor a \rfloor$ the integer part of a real scalar a; $\nabla\phi(x)$ and $\nabla^2\phi(x)$ the gradient and Hessian at x of a twice differentiable function $\phi:\mathbb{R}^d\rightarrow\mathbb{R}$, $d\geq 1$; $\mathbb{P}(\cdot)$ and $\mathbb{E}[\cdot]$ the probability and expectation, respectively; and $\mathcal{I}(\mathcal{A})$ the indicator of event $\mathcal{A}$. For two positive sequences η_n and χ_n, $\eta_n=O(\chi_n)$ means that $\limsup_{n\rightarrow\infty}\frac{\eta_n}{\chi_n}<\infty$; $\eta_n=\Omega(\chi_n)$ means that $\liminf_{n\rightarrow\infty}\frac{\eta_n}{\chi_n}>0$; and $\eta_n=\Theta(\chi_n)$ means that $\eta_n=O(\chi_n)$ and $\eta_n=\Omega(\chi_n)$.

II. Distributed Augmented Lagrangian Algorithms

The network and optimization models are in Subsection II-A, deterministic distributed AL methods are in Subsection II-B, while randomized methods are in Subsection II-C.

A. Optimization and network models

Model. We consider distributed optimization where N nodes solve the unconstrained problem (1). The function $f_i:\mathbb{R}^d\rightarrow\mathbb{R}$, known only to node i, has the following structure.

Assumption 1 (Optimization model) The functions $f_i:\mathbb{R}^d\mapsto\mathbb{R}$ are convex, twice continuously differentiable with bounded Hessian, i.e., there exist $0<h_{\min}\leq h_{\max}<\infty$, such that, for all i:

$$h_{\min}\,I \preceq \nabla^2 f_i(x) \preceq h_{\max}\,I,\ \ \forall x\in\mathbb{R}^d. \tag{5}$$

Under Assumption 1, problem (1) is solvable and has the unique solution $x^\star$. Denote by $f^\star=\inf_{x\in\mathbb{R}^d} f(x)=f(x^\star)$ the optimal value. Further, Assumption 1 implies Lipschitz continuity of the ∇f_i's and strong convexity of the f_i's, i.e., for all i, $\forall x,y\in\mathbb{R}^d$:

$$\|\nabla f_i(x)-\nabla f_i(y)\|\leq h_{\max}\,\|x-y\|,$$
$$f_i(y)\geq f_i(x)+\nabla f_i(x)^\top\,(y-x)+\frac{h_{\min}}{2}\|x-y\|^2.$$

Communication model. We associate with (1) a network $\mathcal{V}$ of N nodes, described by the graph $\mathcal{G}=(\mathcal{V},E)$, where $E\subset\mathcal{V}\times\mathcal{V}$ is the set of edges. (We include self-edges: $\{i,i\}\in E$, $\forall i$.)

Assumption 2 (Network model) The graph $\mathcal{G}$ is connected and undirected.

Weight matrix and weighted Laplacian. Assign to graph $\mathcal{G}$ a symmetric, stochastic (rows sum to one and all the entries are non-negative), $N\times N$ weight matrix W, with, for $i\neq j$, $W_{ij}>0$ if and only if $\{i,j\}\in E$, and $W_{ii}=1-\sum_{j\neq i}W_{ij}$. Let also $\widetilde{W}:=W-J$. (See (4) for the role of W.) We require W to be positive definite and its second largest eigenvalue $\lambda_{N-1}(W)<1$. Let $\mathcal{L}:=I-W$ the weighted graph Laplacian matrix, with $\lambda_2(\mathcal{L})=1-\lambda_{N-1}(W)\in[0,1)$ the network spectral gap that measures how well connected the network is. For example, for a chain N-node network, $\lambda_2(\mathcal{L})=\Theta\left(\frac{1}{N^2}\right)$, while, for expander graphs, it stays bounded away from zero as N grows.

Global knowledge assumptions. We summarize the global knowledge on the system parameters required by our algorithms beforehand at all nodes. They all require (a lower bound on) the Hessian lower bound $h_{\min}$, (an upper bound on) the Hessian upper bound $h_{\max}$, and (a lower bound) on the network spectral gap $\lambda_2(\mathcal{L})$. In addition, the two randomized methods require (an upper bound) on the number of nodes N. Further, each node i initializes its dual variable $\mu_i(0)$ to zero. This is essential for the algorithm's convergence.

[5] Reference [47] assumes all-to-all inter-node communications subject to bounded delays.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2014.2363299, IEEE Transactions on Automatic Control

We assume that all nodes initialize their primal variables to same values, i.e., $x_i(0) = x_j(0)$, $\forall i, j$; e.g., these are set to zero. Equal primal variable initialization is not necessary for convergence but allows for simplified expressions in the analysis. In addition, each node knows its neighborhood set O_i and assigns beforehand the weights W_{ij}, $j \in O_i$. We refer to [54] on how all the above global knowledge can be acquired in a distributed way. Finally, with all our methods, all nodes use the same algorithm parameters: the dual step-size α, the AL penalty ρ, the number of inner iterations τ, and the primal step-size β (with gradient algorithm variants). As we will see in Sections III and IV, the parameters α, β, ρ, and τ need to be appropriately set to ensure convergence; for setting the latter parameters, nodes require knowledge of (bounds on) $h_{\min}$, $h_{\max}$, and $\lambda_2(\mathcal{L})$, and also N with the randomized methods.

B. Deterministic Methods

We present two variants of deterministic distributed AL algorithms of type (2)–(3). They differ in step (2). Both methods solve (2) through inner iterations, indexed by s, and perform (3) in the outer iterations, indexed by k. With the first variant, nodes update their primal variables via a Jacobi-type method; with the second variant, they use a gradient descent method on $L_a(\cdot\,; \mu(k))$. At outer iterations k, with both variants, nodes update the dual variables via the dual gradient ascent method (while the primal variables are fixed).

Jacobi-type primal updates. We detail the first algorithm variant. Later, to present other variants, we indicate only the differences with respect to this one. Denote by: $x_i(k,s)$ the node i's primal variable at the inner iteration s and outer iteration k; and $\mu_i(k)$ the node i's dual variable at the outer iteration k. Further, as in (2)–(3), denote by $x_i(k+1)$ the node i's primal variable at the end of the k-th outer iteration. We relate the primal variables at the inner and outer iterations: $x_i(k, s=0) := x_i(k)$, and $x_i(k+1) := x_i(k, s=\tau)$. In addition, nodes maintain a weighted average of their own and the neighbors' primal variables $\overline{x}_i(k,s) := \sum_{j \in O_i} W_{ij}\, x_j(k,s)$, and $\overline{x}_i(k) := \sum_{j \in O_i} W_{ij}\, x_j(k)$. Recall that O_i is the neighborhood set of node i, including node i.

The algorithm has, as tuning parameters, the weight matrix W, the number of inner iterations per outer iteration τ, the AL penalty parameter $\rho \geq 0$, and the dual step-size $\alpha > 0$. The algorithm is in Algorithm 1. Algorithm 1 has outer iterations k (step 3) and inner iterations s (step 2). At inner iteration s, $s = 0, \cdots, \tau - 1$, node i solves the local optimization problem (6) to obtain $x_i(k, s+1)$, broadcasts $x_i(k, s+1)$ to all its neighbors $j \in O_i - \{i\}$, receives $x_j(k, s+1)$, for all $j \in O_i - \{i\}$; and computes $\overline{x}_i(k, s+1)$ via (7). At outer iteration k, node i updates $\mu_i(k)$ via (8). (Note that (8) is equivalent to (3).) Each inner iteration requires one (d-dimensional) broadcast transmission per node, while the outer (dual) iterations do not require communication. Overall, node i performs τ broadcast transmissions per k.

Algorithm 1 AL with Jacobi-type updates

```
1: (Initialization) Node $i$ sets $k = 0$, $x_i(k=0) \in \mathbb{R}^d$, $\overline{x}_i(k=0) = x_i(0)$, and $\mu_i(k=0) = 0$.
2: (Inner iterations) Node cooperatively run the Jacobi-type method for $s = 0, 1, \cdots, \tau - 1$, with $x_i(k, s=0) := x_i(k)$ and $\overline{x}_i(k, s=0) := \overline{x}_i(k)$:
       $x_i(k, s+1) = \mathrm{argmin}_{x_i \in \mathbb{R}^d} \big( f_i(x_i) + \left(\mu_i(k) - \rho \overline{x}_i(k,s)\right)^\top x_i + \frac{\rho \|x_i\|^2}{2} \big)$   (6)
       $\overline{x}_i(k, s+1) = \sum_{j \in O_i} W_{ij} x_j(k, s+1)$,   (7)
   and set $x_i(k+1) := x_i(k, s=\tau)$, $\overline{x}_i(k+1) = \overline{x}_i(k, s=\tau)$.
3: (Outer iteration) Node $i$ updates the dual variable $\mu_i(k)$:
       $\mu_i(k+1) = \mu_i(k) + \alpha \left( x_i(k+1) - \overline{x}_i(k+1) \right)$.   (8)
4: Set $k \mapsto k+1$ and go to step 2.
```

Gradient primal updates. This algorithm variant is very similar to the Jacobi-type variant. It replaces in the Jacobi variant, Algorithm 1, the Jacobi-type update (6) with the gradient descent update on $L_a(;\mu(k))$ in (4). After algebraic manipulations, obtain the update:

$$
\begin{aligned}
x_i(k, s+1) &= (1 - \beta\, \rho)\; x_i(k,s) + \beta\, \rho\, \overline{x}_i(k,s) \\
&\quad - \beta\, \left(\mu_i(k) + \nabla f_i(x_i(k,s)) \right), \qquad (9)
\end{aligned}
$$

where $\beta > 0$ is the (primal) step-size parameter. Hence, in addition to W, α, and ρ, the gradient primal update algorithm has an additional tuning parameter β.

C. Randomized Methods

We introduce two variants of the randomized distributed AL methods of type (2)–(3). Both utilize the same communication protocol, but they differ in the way primal variables are updated. Like the deterministic counterparts, they both update the dual variables at the outer iterations k, and they update the primal variables at the inner iterations s. At each inner iteration s, one node, say i, is selected uniformly at random from the set of nodes $\{1, 2, \cdots, N\}$. Upon selection, node i updates its primal variable and broadcasts it to all its neighbors. We now detail the time and communication models. The outer iterations occur at discrete time steps of the physical time; k-th outer iteration occurs at time $\tau\, k$, $k = 1, 2, \cdots$, i.e., every τ time units. We assume that all nodes have synchronized clocks for the dual variable updates (dual variable clocks). Each node i has another clock (primal variable clock) that ticks according to a Poisson process with rate 1; on average, there is one tick of node i in the time interval of width 1. Whenever node i's Poisson clock ticks, node i updates its primal variable and broadcasts it to neighbors. The Poisson process clocks are independent. Consider the Poisson process clock that ticks whenever one of the nodes' clocks ticks. This process is a rate-N Poisson process. Hence, in the time interval of length τ, there are on average $\tau\, N$ ticks (primal updates), out of which τ on average are done by i. One primal update here corresponds to an update of a *single* node. Thus, roughly, N updates (ticks) here correspond to one update (inner) iteration of the deterministic algorithm.

More formally, let $(\Theta, \mathcal{F}, \mathbb{P})$ be a probability space. Let $\{\mathcal{T}_i(a, b]\}_{0 \leq a \leq b < \infty}$ be a Poisson process with rate 1, $i =$

$1, \cdots, N$. (This is the node i's clock for primal variables.) Thus, for a fixed a, b, $\mathcal{T}_i(a, b] : \Theta \rightarrow \mathbb{R}$, $\mathcal{T}_i(a, b] = \mathcal{T}_i((a, b]\,; \omega)$, $\omega \in \Theta$, is a Poisson random variable with mean $(b-a)$. Assume the processes $\mathcal{T}_i$ are independent. Let $\mathcal{T}$ be a Poisson process defined by $\mathcal{T}(a, b] := \sum_{i=1}^{N} \mathcal{T}_i(a, b]$. Define the random variable $\tau(k) := \mathcal{T}(k\tau, (k+1)\tau]$ (the number of ticks across all nodes in the k-the outer iteration.) Consider the events $\mathcal{A}_{k,j} := \{\omega \in \Theta : \tau(k; \omega) = j\}$, $j = 0, 1, 2, \cdots$.For $j \geq 1$, define the maps: $\hat{\imath}(k, s) : \mathcal{A}_{k,j} \rightarrow \{1, 2, \cdots, N\}$, $s = 0, \cdots, j-1$, by $\hat{\imath}(k, s; \omega) = i$, if the $(s+1)$-th tick of $\mathcal{T}$ in the interval $(k\tau, (k+1)\tau]$ comes from node i's clock $\mathcal{T}_i$.

We present two variants of the randomized distributed AL algorithm: one updates the primal variables via a Gauss-Seidel-type method and the other replaces the Gauss-Seidel updates by gradient-type updates.

Gauss-Seidel-type updates. The dual variables are updated (instantaneously) at times $k\tau$, $k = 1, 2, \cdots$. We denote by $x_i(k) := x_i(k\tau)$ the node i's primal variable at time $k\tau$, $k = 0, 1, \cdots$ Further, consider $\omega \in \mathcal{A}_{k,j}$: the total number of ticks $\tau(k)$ of $\mathcal{T}$ in the interval $(k\tau, (k+1)\tau]$ equals j, and hence we have j inner iterations (ticks) at the outer iteration k. For any $\omega \in \mathcal{A}_{k,j}$, we denote by $x_i(k, s)$ the node i's variable after the s-th inner iteration, $s = 1, \cdots, j$, $j \geq 1$. Also, let $x_i(k, 0) := x_i(k)$, and, for $\omega \in \mathcal{A}_{k,j}$, $x_i(k, \tau(k) = j) := x_i(k+1)$. Each node maintains: 1) the primal variable $x_i(k)$; 2) the dual variable $\mu_i(k) := \mu_i(k\tau)$; 3) the (weighted) sum of the neighbors' variables $\overline{x}_i(k) := \sum_{j \in O_i} W_{ij} x_j(k)$; and 4) the analogous intermediate variables $x_i(k, s)$ and $\overline{x}_i(k, s)$ during the inner iterations s. The algorithm is Algorithm 3. For all i, and arbitrary fixed k, s, Algorithm 3 defines $x_i(k, s) = x_i(k, s; \omega)$ for any outcome $\omega \in \cup_{t=s}^{\infty} \mathcal{A}_{k,t}$. We formally define $x_i(k, s; \omega) = 0$, for any $\omega \in \Theta$, $\omega \notin \cup_{t=s}^{\infty} \mathcal{A}_{k,t}$. Thus, the random variable $x_i(k, s)$ is defined as in Algorithm 3 for $\omega \in \cup_{t=s}^{\infty} \mathcal{A}_{k,t}$, and $x_i(k, s; \omega) = 0$, for $\omega \notin \cup_{t=s}^{\infty} \mathcal{A}_{k,t}$.

Algorithm 2 Randomized distributed AL with Gauss-Seidel-type updates

```
1: (Initialization) Node $i$ sets $k = 0$, $x_i(k = 0) \in \mathbb{R}^d$, $\overline{x}_i(k = 0) = x_i(k = 0)$, and $\mu_i(k = 0) = 0$.
2: (Inner iterations) Set $x_i(k, s = 0) := x_i(k)$, $\overline{x}_i(k, s = 0) := \overline{x}_i(k)$, and $s = 0$. If $\omega \in \Theta$ is such that $\tau(k) = \tau(k; \omega) > 0$, then, for $s = 0, 1, \cdots, \tau(k) - 1$, do (else, if $\tau(k; \omega) = 0$, then go to step 3):
   Update the inner variables $x_j(k, s)$, $j = 1, \cdots, N$, by :
   $x_j(k, s+1) = \begin{cases} \mathrm{argmin}_{x_j \in \mathbb{R}^d} \left(f_j(x_j) + (\mu_j(k) - \rho \overline{x}_j(k, s))^\top x_j + \frac{\rho \|x_j\|^2}{2}\right) \\ j = \hat{\imath}(k, s) \\ x_j(k, s+1) = x_j(k, s) \\ \text{else.} \end{cases}$   (10)
   Update the variables $\overline{x}_j(k, s)$, $j = 1, \cdots, N$, by :
   $\overline{x}_j(k, s+1) = \begin{cases} \sum_{l \in \Omega_j} W_{jl} x_l(k, s+1) & j \in O_i\text{: } i = \hat{\imath}(k, s) \\ \overline{x}_j(k, s+1) = \overline{x}_j(k, s) \text{else;} \end{cases}$   (11)
   and all nodes $j = 1, \cdots, N$ set $x_j(k+1) := x_j(k, s = \tau(k))$, $\overline{x}_j(k+1) = \overline{x}_j(k, s = \tau(k))$.
3: (Outer iteration) All nodes $j$ update the dual variables $\mu_j(k)$ via:
   $\mu_j(k+1) = \mu_j(k) + \alpha\, (x_j(k+1) - \overline{x}_j(k+1))$.   (12)
4: Set $k \mapsto k+1$ and go to step 2.
```

Gradient primal updates. This algorithm variant is the same as Algorithm 3, except that step (10) is replaced by the following:

$$x_j(k, s+1) = \begin{cases} (1 - \beta\rho)\; x_j(k, s) + \beta\rho \overline{x}_j(k, s) - \beta\left(\mu_j(k) + \nabla f_j(x_j(k, s))\right) \\ \text{for } j = \hat{\imath}(k, s) \\ x_j(k, s+1) = x_j(k, s) \\ \text{else.} \end{cases} \tag{13}$$

Here, $\beta > 0$ is the (primal) step-size parameter.

III. Analysis of inexact augmented Lagrangian methods

In this Section, we introduce our framework for the analysis of inexact AL algorithms (2)–(3). Subsection III-A states our result, while Subsection III-B proves the result through several auxiliary Lemmas. In Section IV, we apply these results to each of the four distributed algorithms.

A. Inexact AL algorithm: Convergence rate

We consider an inexact version of algorithm (2)–(3). Introduce compact notation, and denote by $x(k) := (x_1(k)^\top, ..., x_N(k)^\top)^\top$, and $\mu(k) := (\mu_1(k)^\top, ..., \mu_N(k)^\top)^\top$. Recall the function in (4). For any $\mu \in \mathbb{R}^{Nd}$, denote by $x'(\mu) := \arg\min_{x \in \mathbb{R}^{dN}} L_a(x; \mu)$. The latter quantity is well-defined as the function $L_a(\cdot; \mu)$ is strongly convex in x, for any μ. Recall the weighted Laplacian matrix $\mathcal{L} = I - W$. We consider the following inexact AL method that updates the primal variable $x(k)$ and the dual variable $\mu(k)$ over iterations $k = 0, 1, ...$ The primal variable is initialized to $x(0) = (x_1(0)^\top, ..., x_N(0)^\top)^\top$, with $x_i(0) = x_1(0)$, $\forall i$, $x_1(0) \in \mathbb{R}^d$ arbitrary, and the dual $\mu(0) = 0$. For $k = 0, 1, ...$, given $x(k)$, $\mu(k)$, perform the following update:

$$x(k+1)\,\text{be any point such that :} \tag{14}$$

$$\|x(k+1) - x'(\mu(k))\| \leq \xi\, \|x(k) - x'(\mu(k))\|$$

$$\mu(k+1) = \mu(k) + \alpha\, (\mathcal{L} \otimes I)\, x(k+1). \tag{15}$$

Update (15) is (3) rewritten in a compact form. (Here $\mathcal{L} \otimes I$ is the Kronecker product of $\mathcal{L}$ and the $d \times d$ identity matrix.) In (14), the constant $\xi \in [0, 1)$. Update (14) is an inexact version of (2). Note that $x'(\mu(k))$ corresponds to the exact AL update. We require that $x(k+1)$ be close to $x'(\mu(k))$; more precisely, $x(k+1)$ be ξ times closer to $x'(\mu(k))$ than $x(k)$. The motivation for this condition is the following. Given $\mu(k)$, we seek the new primal variable (ideally $x'(\mu(k))$) via an iterative method, initialized by the previous primal variable $x(k)$. We stop the iterative method as soon as (14) is fulfilled.[6]

[6] As we will see in Section IV, with our distributed methods we do not verify the termination condition in (14) on-the-fly. Instead, given a desired ξ and the network and function parameters, we set beforehand the number of inner iterations τ such that (14) is automatically fulfilled.

We now present our generic Theorem on (14)–(15). We apply it on the four distributed methods in Section IV. Denote by $D_x := \|x_1(0) - x^\star\|$, and $D_\mu := \left(\frac{1}{N} \sum_{i=1}^N \|\nabla f_i(x^\star)\|^2 \right)^{1/2}$.

Theorem 1 Consider algorithm (14)–(15), and let Assumptions 1 and 2 hold. Further, let the algorithm and network parameters satisfy:

$$\alpha \leq h_{\min} + \rho \text{ and } \xi < \frac{1}{3} \frac{\lambda_2(\mathcal{L})\, h_{\min}}{\rho + h_{\max}}. \tag{16}$$

Then, $\forall i$, $x_i(k)$ generated by (14)–(15) converges linearly to the solution $x^\star$ of (1), with convergence factor:

$$r := \max \left\{ \frac{1}{2} + \frac{3}{2}\xi, \left(1 - \frac{\alpha \lambda_2(\mathcal{L})}{\rho + h_{\max}}\right) + \frac{3\alpha}{h_{\min}}\xi \right\} < 1. \tag{17}$$

It holds:

$$\|x_i(k) - x^\star\| \leq r^k \sqrt{N} \max \left\{ D_x, \frac{2 D_\mu}{\sqrt{\lambda_2(\mathcal{L})} h_{\min}} \right\}. \tag{18}$$

Theorem 1 establishes that the inexact AL method converges to the *primal* solution at the globally linear rate in the number of *outer* iterations, provided that ξ is sufficiently small, and it quantifies the achieved rate as well as how small ξ should be. We emphasize the interesting effect of constant D_μ. It measures how difficult it is to solve (1) by distributed methods like (2)–(3)–the larger, the more difficult the problem is. If, at an extreme, the f_i's all have the same minimizer, say $y^\star$, then $y^\star$ is also the minimizer of (1) ($y^\star = x^\star$.) Such problem is "easy," because nodes do not need to communicate with others to obtain the global solution to (1)–"easyness" of the problem agrees with the value $D_\mu = 0$. On the other hand, if the local minimizers (of the f_i's), say $y_i^\star$'s, are very different, then they may be very different from $x^\star$. Hence, node i needs to communicate with others to recover $x^\star$. This agrees with D_μ large in such scenarios. (See Lemma 2 that relates D_μ to the dual optimum.)

B. Auxiliary results and proof of Theorem 1

We now prove Theorem 1 by introducing several auxiliary objects and results. We base our analysis on the following nonlinear saddle point system of equations:

$$\nabla F(x) + \mu + \rho\, (\mathcal{L} \otimes I)\, x = 0 \tag{19}$$

$$(\mathcal{L} \otimes I) x = 0 \tag{20}$$

$$(1 \otimes I)^\top \mu = 0. \tag{21}$$

In (19), $\rho \geq 0$ is the AL penalty parameter, and $F : \mathbb{R}^{N\,d} \mapsto \mathbb{R}$ is defined by $F(x) = F(x_1, \cdots, x_N) = f_1(x_1) + f_2(x_2) + \cdots + f_N(x_N)$. In (19), $x, \mu \in \mathbb{R}^{N\,d}$ are the primal and dual variables, whose i-th d-dimensional blocks correspond to node i's primal and dual variables, respectively. In (19)–(21) and in subsequent text, Kronecker products $a \otimes b$ are always such that the left object a is of size either $N \times 1$ or $N \times N$, while the right object is of size $d \times 1$ or $d \times d$. Henceforth, to simplify notation, we do not designate the objects' dimensions. The next Lemma shows that solving (19)–(21) solves (1) at each node i.

Lemma 2 Consider optimization problem (1) and the nonlinear system (19)–(21) , and let Assumptions 1 and 2 hold. Then, there exists unique $(x^\bullet, \mu^\bullet) \in \mathbb{R}^{Nd} \times \mathbb{R}^{Nd}$ that satisfies (19)–(21), with $x^\bullet = 1 \otimes x^\star$, where $x^\star$ is the solution to (1) and $\mu^\bullet = -\nabla F(1 \otimes x^\star)$.

Proof: First show $x^\bullet = 1 \otimes x^\star$ and $\mu^\bullet = -\nabla F(1 \otimes x^\star)$ solve (19)–(21). Consider (20). We have $(\mathcal{L} \otimes I)x^\bullet = (\mathcal{L} \otimes I)(1 \otimes x^\star) = (\mathcal{L}\, 1) \otimes (I\, x^\star) = 0$, since $1/\sqrt{N}$ is the (unique) unit-norm eigenvector with eigenvalue 0 of the Laplacian for a connected network. Next:

$$(1 \otimes I)^\top \mu^\bullet = -\sum_{i=1}^N \nabla f_i(x^\star) = 0.$$

The right equality holds because $x^\star$ is the solution to (1). Finally, because $(\mathcal{L} \otimes I)x^\bullet = 0$ (already shown) and $\nabla F(x^\bullet) = -\mu^\bullet$, we have ($x^\bullet = 1 \otimes x^\star$, $\mu^\bullet = -\nabla F(1 \otimes x^\star)$) satisfy (19)–(21). The uniqueness is by the uniqueness of the solution to (1) due to strong convexity. ■

Next, introduce the following maps $\Phi : \mathbb{R}^{Nd} \mapsto \mathbb{R}^{Nd}$, $\Psi : \mathbb{R}^{Nd} \mapsto \mathbb{R}^{Nd}$, and $\Phi_i : \mathbb{R}^d \mapsto \mathbb{R}^d$, $i = 1, ..., N$:

$$\Phi(x) := \nabla F(x) + \rho\, I\, x \tag{22}$$

$$\Psi(x) := \nabla F(x) + \rho\, \mathcal{L}\, x \tag{23}$$

$$\Phi_i(x) := \nabla f_i(x) + \rho\, x. \tag{24}$$

Further, define the maps: $\Phi^{-1} : \mathbb{R}^{Nd} \to \mathbb{R}^{Nd}$, $\Psi^{-1} : \mathbb{R}^{Nd} \to \mathbb{R}^{Nd}$, and $\Phi_i^{-1} : \mathbb{R}^d \to \mathbb{R}^d$ by:

$$\Phi^{-1}(\mu) := \mathrm{argmin}_{y \in \mathbb{R}^{Nd}} \Big(F(y) - \mu^\top y + \frac{\rho}{2} \|y\|^2 \Big) \tag{25}$$

$$\Psi^{-1}(\mu) := \mathrm{argmin}_{y \in \mathbb{R}^{Nd}} \Big(F(y) - \mu^\top y + \frac{\rho}{2} y^\top \mathcal{L} y \Big) \tag{26}$$

$$\Phi_i^{-1}(\mu) := \arg\min{}_{y \in \mathbb{R}^d} \left(f_i(y) - \mu_i^\top\, y + \frac{\rho}{2}\, \|y\|^2 \right). \tag{27}$$

The cost function in (26) is precisely $L_a(y; -\mu)$ in (4). For any $\mu \in \mathbb{R}^{Nd}$, these maps are well-defined by Assumption 1 (This assumption ensures that there exists a unique solution in the minimizations in (25)–(27), as the costs in (25)–(27) are all strongly convex.) Next, we have:

$$\nabla F(\Phi^{-1}(\mu)) + \rho\, I \Phi^{-1}(\mu) = \mu = \Phi(\Phi^{-1}(\mu)),$$

where the left equality is by the first order optimality conditions, from (25), and the right equality is by definition of Φ in (22). Thus, the map Φ^{-1} is the inverse of Φ. Likewise, the map Ψ^{-1} (Φ_i^{-1}) is the inverse of Ψ (Φ_i). By the inverse function theorem, e.g., [55], the maps $\Phi^{-1} : \mathbb{R}^{Nd} \to \mathbb{R}^{Nd}$, $\Psi^{-1} : \mathbb{R}^{Nd} \to \mathbb{R}^{Nd}$, and $\Phi_i^{-1} : \mathbb{R}^d \to \mathbb{R}^d$ are continuously differentiable, with derivatives:

$$\nabla \Phi^{-1}(\mu) = \left(\nabla^2 F(\Phi^{-1}(\mu)) + \rho\, I \right)^{-1} \tag{28}$$

$$\nabla \Psi^{-1}(\mu) = \left(\nabla^2 F(\Psi^{-1}(\mu)) + \rho\, (\mathcal{L} \otimes I) \right)^{-1} \tag{29}$$

$$\nabla \Phi_i^{-1}(\mu) = \left(\nabla^2 f_i(\Phi_i^{-1}(\mu)) + \rho\, I \right)^{-1}. \tag{30}$$

Note that invertibility is assured because $\nabla^2 F(x)$ and $\nabla^2 f_i(x_i)$ are positive definite, $\forall x \in \mathbb{R}^{Nd}$, $\forall x_i \in \mathbb{R}^d$, and so are the matrices in (28)–(30). Using the following identity for a continuously differentiable map $h : \mathbb{R}^{Nd} \to \mathbb{R}^{Nd}$,

$\forall u, v \in \mathbb{R}^{Nd}$:

$$h(u) - h(v) = \left[\int_0^1 \nabla h(v + z(u-v))dz\right](u-v), \tag{31}$$

we obtain the following useful relations:

$$\Phi^{-1}(\mu_1) - \Phi^{-1}(\mu_2) = R_\Phi(\mu_1,\mu_2)\,(\mu_1 - \mu_2), \tag{32}$$
$$R_\Phi(\mu_1,\mu_2) := \int_{z=0}^{1} \nabla\Phi^{-1}(\mu_1 + z(\mu_2 - \mu_1))\,dz$$
$$\Psi^{-1}(\mu_1) - \Psi^{-1}(\mu_2) = R_\Psi(\mu_1,\mu_2)(\mu_1 - \mu_2), \tag{33}$$
$$R_\Psi(\mu_1,\mu_2) := \int_{z=0}^{1} \nabla\Psi^{-1}(\mu_1 + z(\mu_2 - \mu_1))\,dz$$
$$\Phi_i^{-1}(\mu_1) - \Phi_i^{-1}(\mu_2) = R_{\Phi,i}(\mu_1,\mu_2)\,(\mu_1 - \mu_2), \tag{34}$$
$$R_{\Phi,i}(\mu_1,\mu_2) := \int_{z=0}^{1} \nabla\Phi_i^{-1}(\mu_1 + z(\mu_2 - \mu_1))dz.$$

By Assumption 1: $h_{\min} I \preceq \nabla^2 F(x) \preceq h_{\max} I$, $\forall x \in \mathbb{R}^{Nd}$. Using the latter, (28), (29), (31), and $\mathcal{L} = I - W$, $0 \preceq \mathcal{L} \preceq I$ ($W \succ 0$, symmetric, stochastic), we obtain the following properties of the $(Nd) \times (Nd)$ matrices $R_\Phi(\mu_1,\mu_2)$ and $R_\Psi(\mu_1,\mu_2)$, and $d \times d$ matrices $R_{\Phi,i}(\mu_1,\mu_2)$:

$$\frac{1}{h_{\max}+\rho} I \;\preceq\; R_\Phi(\mu_1,\mu_2) \preceq \frac{1}{h_{\min}+\rho} I, \quad \forall \mu_1,\mu_2 \in \mathbb{R}^{Nd} \tag{35}$$

$$\frac{1}{h_{\max}+\rho} I \;\preceq\; R_\Psi(\mu_1,\mu_2) \preceq (h_{\min} I + \rho(\mathcal{L}\otimes I))^{-1}, \quad \forall \mu_1,\mu_2 \in \mathbb{R}^{Nd} \tag{36}$$

$$\frac{1}{h_{\max}+\rho} I \;\preceq\; R_{\Phi,i}(\mu_1,\mu_2) \preceq \frac{1}{h_{\min}+\rho} I, \quad \forall \mu_1,\mu_2 \in \mathbb{R}^{d}. \tag{37}$$

The right inequality in (36) holds because, $\forall \mu$, $\nabla^2 F(\Psi^{-1}(\mu)) + \rho\,(\mathcal{L}\otimes I) \succeq h_{\min} I + \rho\,(\mathcal{L}\otimes I)$ (due to Assumption 1), and so $[\,\nabla^2 F(\Psi^{-1}(\mu)) + \rho\,(\mathcal{L}\otimes I)\,]^{-1} \preceq [\,h_{\min} I + \rho\,(\mathcal{L}\otimes I)\,]^{-1}$.

Denote by $\widetilde{x}(k) := x(k) - x^\bullet$ and $\widetilde{\mu}(k) := \mu(k) - \mu^\bullet$ the primal and dual errors, respectively. Also, write $x'(k+1) := x'(\mu(k))$, to simplify notation. We now state and prove several Lemmas that allow us to prove Theorem 1. We prove these lemmas assuming $d = 1$, to avoid further extensive use of Kronecker products; the proofs extend to generic $d > 1$. We first upper bound the primal error $\|\widetilde{x}(k+1)\|$.

Lemma 3 (Primal error) Let Assumptions 1, 2 hold. Then, for $k = 0, 1, \cdots$

$$\|\widetilde{x}(k+1)\| \le \xi\|\widetilde{x}(k)\| + \frac{1}{h_{\min}}\,(1+\xi)\,\|\widetilde{\mu}(k)\|.$$

Proof: Write $\widetilde{x}(k+1) = (x(k+1) - x'(k+1)) + (x'(k+1) - x^\bullet)$. Then, $\|\widetilde{x}(k+1)\| \le \|x(k+1) - x'(k+1)\| + \|x'(k+1) - x^\bullet\|$. From (14), we know that $\|x(k+1) - x'(k+1)\| \le \xi\|x(k) - x'(k+1)\|$. The latter is further upper bounded as: $\|x(k) - x'(k+1)\| \le \xi\|x(k) - x^\bullet + x^\bullet - x'(k+1)\| \le \xi\|\widetilde{x}(k)\| + \xi\|x^\bullet - x'(k+1)\|$. Hence,

$$\|\widetilde{x}(k+1)\| \le \xi\|\widetilde{x}(k)\| + (1+\xi)\|x'(k+1) - x^\bullet\|. \tag{38}$$

It remains to upper bound $\|x'(k+1) - x^\bullet\|$. Note that $x^\bullet = \Psi^{-1}(-\mu^\bullet)$, and $x'(k+1) = \Psi^{-1}(-\mu(k))$. Using the latter and (33), we obtain:

$$\begin{aligned} x'(k+1) - x^\bullet &= \Psi^{-1}(-\mu(k)) - \Psi^{-1}(\mu^\bullet) \\ &= -R_\Psi(k)\,(\mu(k) - \mu^\bullet), \end{aligned} \tag{39}$$

with $R_\Psi(k) := R_\Psi(-\mu(k), -\mu^\bullet)$. This, with (36), and $\widetilde{\mu}(k) = \mu(k) - \mu^\bullet$, gives:

$$\|x'(k+1) - x^\bullet\| \le \frac{1}{h_{\min}}\,\|\widetilde{\mu}(k)\|. \tag{40}$$

The result follows from (38) and (40). ■

Since our final goal is to bound the primal error, rather than bounding $\widetilde{\mu}(k) = \mu(k) - \mu^\bullet$, it turns out to be more useful to bound a certain transformed dual quantity. Represent the weighted Laplacian matrix $\mathcal{L}$ through its (reduced) eigen-decomposition (we do not include the pair $(0, q_1)$) $\mathcal{L} = Q\widehat{\Lambda}Q^\top = \sum_{i=2}^{N} \lambda_i\, q_i q_i^\top$, where (λ_i, q_i) is the i-th eigenvalue, eigenvector pair ($\lambda_i > 0$, for all $i = 2, \cdots, N$); $Q = [q_2, \cdots, q_N]$; and $\widehat{\Lambda} = \operatorname{diag}(\lambda_2, \cdots, \lambda_N)$. Instead of bounding the dual error, we bound the norm of $\widetilde{\mu}''(k) \in \mathbb{R}^{N-1}$ that we define:

$$\widetilde{\mu}'(k) := Q^\top\widetilde{\mu}(k) \in \mathbb{R}^{N-1} \text{ and } \widetilde{\mu}''(k) := \widehat{\Lambda}^{-1/2}\widetilde{\mu}'(k). \tag{41}$$

Lemma 4 (Dual error) Let $\alpha \le h_{\min} + \rho$, and let Assumptions 1 and 2 hold. Then, for all $k = 0, 1, \cdots$

$$\|\widetilde{\mu}''(k+1)\| \le \left[\left(1 - \frac{\alpha\lambda_2(\mathcal{L})}{h_{\max}+\rho}\right) + \frac{\alpha}{h_{\min}}\xi\right]\|\widetilde{\mu}''(k)\| + \alpha\xi\|\widetilde{x}(k)\|.$$

Proof: Because $\mathcal{L}x^\bullet = \mathcal{L}x^\star\,1 = 0$:

$$\mathcal{L}x(k+1) = \mathcal{L}(x(k+1) - x'(k+1)) + \mathcal{L}(x'(k+1) - x^\bullet).$$

Using this and subtracting $\mu^\bullet$ from both sides of (15):

$$\begin{aligned} \widetilde{\mu}(k+1) = \widetilde{\mu}(k) &+ \alpha\,\mathcal{L}(x'(k+1) - x^\bullet) \\ &+ \alpha\,\mathcal{L}(x(k+1) - x'(k+1)). \end{aligned} \tag{42}$$

Further, using (39), we get:

$$\widetilde{\mu}(k+1) = (I - \alpha\mathcal{L}R_\Psi(k))\widetilde{\mu}(k) + \alpha\mathcal{L}(x(k+1) - x'(k+1)). \tag{43}$$

Now, recall $\widetilde{\mu}'(k)$ in (41). It is easy to see that:

$$\|\widetilde{\mu}'(k)\| = \|\widetilde{\mu}(k)\|, \quad QQ^\top\widetilde{\mu}(k) = \widetilde{\mu}(k). \tag{44}$$

Indeed, note that $1^\top\mu(k) = 1^\top\mu(k-1) + \alpha 1^\top\mathcal{L}x(k) = 1^\top\mu(k-1) = \cdots = 1^\top\mu(0) = 0$, because $\mu(0) = 0$ (by assumption.) Also, $1^\top\mu^\bullet = 0$ (see Lemma 2.) Therefore, $1^\top\widetilde{\mu}(k) = 0$, $\forall k$. Now, as $q_1 = \frac{1}{\sqrt{N}}1$, we have $QQ^\top\widetilde{\mu}(k) = \sum_{i=2}^{N} q_i q_i^\top\widetilde{\mu}(k) = \sum_{i=1}^{N} q_i q_i^\top\widetilde{\mu}(k) = \widetilde{\mu}(k)$; thus, the second equality in (44). For the first equality in (44), observe that: $\|\widetilde{\mu}'(k)\|^2 = (\widetilde{\mu}'(k))^\top\widetilde{\mu}'(k) = \widetilde{\mu}(k)^\top QQ^\top\widetilde{\mu}(k) = \|\widetilde{\mu}(k)\|^2$.

Next, multiplying (43) from the left by $Q^\top$, expressing $\mathcal{L} = Q\widehat{\Lambda}Q^\top$, and using (44), obtain:

$$\begin{aligned} \widetilde{\mu}'(k+1) = &\left(\,I - \alpha\,\widehat{\Lambda}\,Q^\top R_\Psi(k)\,Q\,\right)\,\widetilde{\mu}'(k) \\ &+ \alpha\,\widehat{\Lambda}\,Q^\top\,(x(k+1) - x'(k+1)). \end{aligned} \tag{45}$$

Further, recall $\widetilde{\mu}''(k)$ in (41). Multiplying (45) from the left by $\widehat{\Lambda}^{-1/2}$, we obtain:

$$\widetilde{\mu}''(k+1) = \left(I - \alpha\, \widehat{\Lambda}^{1/2}\, Q^\top R_\Psi(k)\, Q \widehat{\Lambda}^{1/2} \right)\, \widetilde{\mu}''(k) \\ + \alpha\, \widehat{\Lambda}^{1/2}\, Q^\top\, (x(k+1) - x'(k+1)). \tag{46}$$

Next, using variational characterizations of minimal and maximal eigenvalues, we can verify:

$$\frac{\lambda_2}{h_{\max} + \rho}\, I \preceq \widehat{\Lambda}^{1/2}\, Q^\top R_\Psi(k)\, Q \widehat{\Lambda}^{1/2} \preceq \frac{1}{h_{\min} + \rho}\, I. \tag{47}$$

The right inequality in (47) holds because of the following. First, use the right inequality in (36) to show $\widehat{\Lambda}^{1/2}\, Q^\top R_\Psi(k)\, Q \widehat{\Lambda}^{1/2} \preceq \widehat{\Lambda}^{1/2}\, Q^\top [\, h_{\min} I + \rho \mathcal{L}\,]^{-1}\, Q \widehat{\Lambda}^{1/2}$. (Note that $\widehat{\Lambda}$ is $(N-1) \times (N-1)$, Q is $N \times (N-1)$, and $[\, h_{\min} I + \rho \mathcal{L}\,]^{-1}$ is $N \times N$.) Next, decompose the $N \times N$ matrix $[\, h_{\min} I + \rho \mathcal{L}\,]^{-1}$ via the $(N \times N)$ eigenvalue decomposition, and use orthogonality of the eigenvectors of $\mathcal{L}$ to show that the $((N-1) \times (N-1))$ matrix: $\widehat{\Lambda}^{1/2}\, Q^\top R_\Psi(k)\, Q \widehat{\Lambda}^{1/2} \preceq \widehat{\Lambda}^{1/2} [\, h_{\min} I + \rho \widehat{\Lambda}\,]^{-1} \widehat{\Lambda}^{1/2}$. The maximal eigenvalue of $\widehat{\Lambda}^{1/2} [\, h_{\min} I + \rho \widehat{\Lambda}\,]^{-1} \widehat{\Lambda}^{1/2}$ is $\frac{1}{h_{\min}/\lambda_N(\mathcal{L}) + \rho} \leq \frac{1}{h_{\min} + \rho}$. Next, by Assumption, $\alpha \leq h_{\min} + \rho$, and so:

$$\| I - \alpha\, \widehat{\Lambda}^{1/2}\, Q^\top R_\Psi(k)\, Q \widehat{\Lambda}^{1/2} \| \leq 1 - \frac{\alpha\, \lambda_2}{h_{\max} + \rho}. \tag{48}$$

Using (48), $\|\widehat{\Lambda}^{1/2}\| \leq 1$ (as $0 \preceq \mathcal{L} \preceq I$), $\|Q\| = 1$, and Lemma 3, we get:

$$\|\widetilde{\mu}''(k+1)\| \leq \left(1 - \frac{\alpha\, \lambda_2}{h_{\max} + \rho} \right) \|\widetilde{\mu}''(k)\| \\ + \alpha \xi\, \|\widetilde{x}(k)\| + \alpha \xi \frac{\|\widetilde{\mu}(k)\|}{h_{\min}}.$$

Finally, using $\|\widetilde{\mu}(k)\| = \|\widetilde{\mu}'(k)\| = \|\widehat{\Lambda}^{1/2} \widetilde{\mu}''(k)\| \leq \|\widetilde{\mu}''(k)\|$, we obtain the desired result. ■

We are now ready to prove Theorem 1.

Proof of Theorem 1: Introduce $\nu(k) := \frac{2}{h_{\min}} \|\widetilde{\mu}''(k)\|$. Further, denote by $c_{11} := \xi$, $c_{12} := \frac{1}{2}\,[1 + \xi\,]$; $c_{21} := \frac{2\,\alpha}{h_{\min}} \xi$, and $c_{22} := \left(1 - \frac{\alpha\, \lambda_2}{h_{\max} + \rho} \right) + \frac{\alpha}{h_{\min}} \xi$. Using $\|\widetilde{\mu}(k)\| \leq \|\widetilde{\mu}''(k)\|$, Lemma 3, and Lemma 4, we obtain:

$$\max \left\{ \|\widetilde{x}(k+1)\|,\, \nu(k+1) \right\} \leq r\, \max \left\{ \|\widetilde{x}(k)\|,\, \nu(k) \right\},$$

with $r = \max \left\{ c_{11} + c_{12},\, c_{21} + c_{22} \right\}$. Unwinding the recursion, using $\|\widetilde{x}(k)\| \leq \max\{\|\widetilde{x}(k)\|,\, \nu(k)\}$, $\nu(0) = \frac{2}{h_{\min}} \|\widehat{\Lambda}^{-1/2} Q^\top \widetilde{\mu}(0)\| = \frac{2}{h_{\min}} \|\widehat{\Lambda}^{-1/2}\, Q^\top\, (-\nabla F(x^\star\, 1))\| \leq \frac{2}{h_{\min} \sqrt{\lambda_2}} \sqrt{N} D_\mu$, obtain (18).

It remains to show that $r < 1$ if conditions (16) hold.Note that: $c_{11} + c_{12} = \frac{1}{2} + \frac{3}{2} \xi$, and so $c_{11} + c_{12} < 1$ if: $\xi < \frac{1}{3}$. Next, note that: $c_{21} + c_{22} = \left(1 - \frac{\alpha\, \lambda_2}{\rho + h_{\max}} \right) + \frac{3\alpha}{h_{\min}} \xi$, and so $c_{21} + c_{22} < 1$ if: $\xi < \frac{1}{3} \left(\frac{h_{\min} \lambda_2}{\rho + h_{\max}} \right)$. Combining the last two conditions, obtain $r < 1$ if conditions (16) hold.The proof is complete. ■

IV. Analysis of distributed augmented Lagrangian methods

In this Section, we specialize our results from Section III to each of the four distributed AL algorithm variants. More precisely, we characterize the quantity ξ in (14) with each method. This, with Theorem 1, allows us to establish convergence rates in the inner iterations.

With each of the four variants, we use compact notation: $x(k) = (x_1(k)^\top, ..., x_N(k)^\top)^\top$, $\mu(k) = (\mu_1(k)^\top, ..., \mu_N(k)^\top)^\top$, and $x(k,s) = (x_1(k,s)^\top, ..., x_N(k,s)^\top)^\top$. We start with the deterministic Jacobi-type variant. For the proofs of the results in current Section, we let $d = 1$ for notation simplicity, but they extend to a generic $d > 1$.

Lemma 5 (Deterministic Jacobi-type) Consider the distributed AL algorithm with deterministic Jacobi-type primal updates and τ inner iterations. Further, let Assumptions 1 and 2 hold. Then, for all $k = 0, 1, \cdots$:

$$\|x(k+1) - x'(k+1)\| \leq \left(\frac{\rho}{\rho + h_{\min}} \right)^\tau \|x(k) - x'(k+1)\|.$$

Proof: Recall that $x'(k+1) = \arg\min_{x \in \mathbb{R}^N} L_a(x; \mu(k))$. From the corresponding first order optimality conditions, we have: $\nabla F(x'(k+1)) + \rho\, \mathcal{L}\, x'(k+1) = -\mu(k)$. Hence, using $\mathcal{L} = I - W$ and the definition of Φ in (22):

$$x'(k+1) = \Phi^{-1}\, (\, \rho\, W x'(k+1) - \mu(k)\,). \tag{49}$$

Fix s, $0 \leq s \leq \tau - 1$. Next, from Algorithm 1 and definition of Φ:

$$x(k, s+1) \;\;=\;\; \Phi^{-1}\, (\, \rho\, W\, x(k,s) - \mu(k)\,)\,; \tag{50}$$

Subtracting $x'(k+1)$ from both sides of (50), and using (49) and (32):

$$x(k,s+1) - x'(k+1) = R_\Phi(s) \rho W (x(k,s) - x'(k+1)),$$

where $R_\Phi(s) := R_\Phi(\rho W x(k,s) - \mu(k), \rho W x'(k+1) - \mu(k))$. Using (35) and $\|W\| = 1$, obtain:

$$\|x(k,s+1) - x'(k+1)\| \leq \left(\frac{\rho}{\rho + h_{\min}} \right) \|x(k,s) - x'(k+1)\|.$$

Applying this for $s = 0, 1, \cdots, \tau - 1$, using $x(k,\tau) = x(k+1)$, $x(k,0) = x(k)$, get:

$$\|x(k+1) - x'(k+1)\| \leq \left(\frac{\rho}{\rho + h_{\min}} \right)^\tau \|x(k) - x'(k+1)\|. \tag{51}$$

■

The immediate corollary of Lemma 5 is that, for the distributed AL algorithm with Jacobi-type primal updates, Theorem 1 holds with $\xi := \left(\frac{\rho}{\rho + h_{\min}} \right)^\tau$. In other words, if the conditions on the system parameters in Theorem 1 hold, the distributed AL algorithm converges linearly in the outer iterations. Furthermore, as the number of inner iterations is fixed and equals τ, the algorithm also converges linearly in the number of inner iterations, and hence in the number of per-node communications, with the convergence factor $r^{1/\tau}$. Note that, for any choice of $\rho \geq 0$, we can choose α and τ such that linear convergence is assured. Recall the f_i's condition number $\gamma = h_{\max}/h_{\min}$. Setting $\rho = h_{\max}$, $\alpha = h_{\min} + \rho$, and $\tau = \left\lceil \frac{\log(12(\gamma+1)/\lambda_2)}{\log(1+1/\gamma)} \right\rceil$, we obtain the convergence factor at outer iterations $r = 1 - \Omega(\lambda_2)$. Hence, interestingly, we can eliminate the negative effect of the condition number γ

at the outer iterations level. Of course, we pay a price at the inner iterations level, where the convergence factor is, for λ_2 bounded away from one, $r^{1/\tau} = 1 - \Omega\left(\frac{\lambda_2}{\gamma\,\log(\gamma/\lambda_2)}\right)$.

We remark that, for a reasonable choice of the step-size α and the AL penalty ρ, e.g., $\alpha = \rho = h_{\min}$, our results do not guarantee linear convergence for $\tau = 1$. (Hence, we do not guarantee convergence either, for $\tau = 1$.) However, we know from the literature that, for any choice of $\alpha = \rho > 0$, and a certain choice of W (see [10]), the algorithm with Jacobi-type updates and $\tau = 1$ (a distributed ADMM) converges globally linearly to the primal solution [10]. This, in particular, means that, for $\tau = 1$, $\alpha = \rho > 0$, and W in [10], the algorithm always converges, and always at a globally linear rate.

We now consider the deterministic gradient variant.

Lemma 6 (Deterministic gradient) Consider the distributed AL algorithm with deterministic gradient primal updates with τ inner iterations and the primal step-size $\beta \leq 1/(h_{\max} + \rho)$. Further, let Assumptions 1 and 2 hold. Then, for all $k = 0, 1, \cdots$:

$$\|x(k+1) - x'(k+1)\| \leq (1 - \beta\, h_{\min})^{\tau}\ \|x(k) - x'(k+1)\|\,.$$

Proof: Using $\mathcal{L} = I - W$ and compact notation, the update (9) is rewritten as:

$$x(k,s+1)=x(k,s)-\beta(\rho\mathcal{L}x(k,s)+\mu(k)+\nabla F(x(k,s))). \tag{52}$$

This is the gradient descent on $L_a(\cdot;\mu(k))$ in (4). As $x'(k+1)$ satisfies $\rho\,\mathcal{L}\,x'(k+1) + \mu(k) + \nabla F(x'(k+1)) = 0$, we have:

$$x'(k+1)=x'(k+1)-\beta(\rho\mathcal{L}x'(k+1)+\mu(k)+\nabla F(x'(k+1))). \tag{53}$$

Further, by Assumption 1, $\nabla F : \mathbb{R}^N \to \mathbb{R}^N$ is continuously differentiable, and it holds:

$$\begin{aligned} & \nabla F(x(k,s)) - \nabla F(x'(k+1)) = \\ & \left[\int_{z=0}^{1} \nabla^2 F\left(x'(k+1) + z(x(k,s) - x'(k+1))\right) dz\right] \\ \times\quad & (x(k,s) - x'(k+1)) \\ =:\quad & H_F\,(s)\,(x(k,s) - x'(k+1)). \end{aligned} \tag{54}$$

Further, by Assumption 1, the matrix $H_F\,(s)$ satisfies:

$$h_{\min}\, I \preceq H_F(s) \preceq h_{\max}\, I. \tag{55}$$

Using (54), and subtracting (53) from (52), we obtain:

$$\begin{aligned} & x(k,s+1) - x'(k+1) = (I - \beta\,\rho\,\mathcal{L} - \beta\,H_F(s)) \\ & \times (x(k,s) - x'(k+1)). \end{aligned} \tag{56}$$

Consider the matrix $(I - \beta\,\rho\,\mathcal{L} - \beta\,H_F(s))$. As $\beta \leq \frac{1}{\rho + h_{\max}}$ (by assumption), using (55) and $0 \preceq \mathcal{L} \preceq I$, get: $(I - \beta\,\rho\,\mathcal{L} - \beta\,H_F(s)) \succeq 0$. Thus, $\|I - \beta\,\rho\,\mathcal{L} - \beta\,H_F(s)\| \leq 1 - \lambda_1\,(\beta\,\rho\,\mathcal{L} + \beta\,H_F(s)) \leq 1 - \beta\,h_{\min}$. Applying this bound to (56), obtain the inequality:

$$\|x(k,s+1)-x'(k+1)\|\leq(1-\beta h_{\min})\,\|x(k,s)-x'(k+1)\|. \tag{57}$$

Applying (57) for $s = 0, \cdots, \tau - 1$, using $x(k, s=0) = x(k)$, and $x(k, s=\tau) = x(k+1)$, we obtain the desired result. ■ The immediate corollary of Lemma 6 is that Theorem 1 holds for the deterministic gradient variant, with $\xi = (1 - \beta\,h_{\min})^{\tau}$. Hence, under conditions of Theorem 1, the algorithm converges linearly in the number of inner iterations, with the convergence factor $r^{1/\tau}$. This implies the linear convergence both in the number of per-node communications and in the number of per-node gradient evaluations (gradients of f_i's). Setting $\rho = h_{\max}$, $\alpha = h_{\min} + \rho$, $\beta = \frac{1}{h_{\max}+\rho}$, and: $\tau = \left\lceil \frac{\log(12(1+\gamma)/\lambda_2)}{\log\left(1+\frac{1}{2\gamma-1}\right)} \right\rceil$, gives the convergence factor in the inner iterations (for λ_2 bounded away from one) as $r^{1/\tau} = 1 - \Omega\left(\frac{\lambda_2}{\gamma\,\log(\gamma/\lambda_2)}\right)$.

Note that, for reasonable choices of α, β, and ρ, e.g., $\alpha = \rho = h_{\min}$, $\beta = 1/(\rho + h_{\max})$, our results do not guarantee convergence nor linear convergence rates when we set $\tau = 1$. Reference [20] establishes global convergence of a similar algorithm for $\tau = 1$, $\rho = 0$, and a sufficiently small α and β. An interesting research direction is to explore whether there is a boundary between stability results and global linear rates. In other words, setting $\tau = 1$, an open problem is whether for certain choices of α, β, and ρ the algorithm converges, but at globally sub-linear rates. (Recall that this scenario does not occur with the Jacobi-type variant.) Another important open problem is to research whether, for $\tau = 1$, there exists a choice of α, β, and ρ that ensures globally linear rates.

Recall the random model in Subsection II-C and the randomized Gauss-Seidel-type method.

Lemma 7 (Randomized Gauss-Seidel-type) Consider the distributed AL algorithm with randomized Gauss-Seidel-type primal updates, where the expected number of inner iterations equals τ. Further, let Assumptions 1 and 2 hold. Then, for all $k = 0, 1, \cdots$:

$$\mathbb{E}\left[\,\|x(k+1) - x'(k+1)\|\,\right] \leq e^{-\eta\,\tau}\,\mathbb{E}\left[\,\|x(k) - x'(k+1)\|\,\right],$$

where

$$\eta := N\left\{1 - \left[1 - \frac{1}{N}\left(1 - \frac{\rho^2}{(\rho + h_{\min})^2}\right)\right]^{1/2}\right\}. \tag{58}$$

Proof: Fix some k, fix some $j = 1, 2, ...,$ and take $\omega \in \mathcal{A}_{k,j}$. Thus, $\tau(k) = \tau(k;\omega) = j$ and there are j inner iterations. Fix some s, $s \in \{0, 1, ..., j-1\}$, and suppose that $\widehat{\imath}(k,s) = i$ (node i is activated.) We have that $x_i(k, s+1)$ satisfies the following:

$$x_i(k,s+1) = \Phi_i^{-1}\left(\sum_{j\in O_i} \rho\,W_{ij}\,x_j(k,s) - \mu_i(k)\right).$$

On the other hand, we know that $x'_i(k+1)$ satisfies:

$$x'_i(k+1) = \Phi_i^{-1}\left(\sum_{j\in O_i} \rho\,W_{ij}\,x'_j(k+1) - \mu_i(k)\right).$$

Subtracting the above equalities, and using (37), letting

$$\begin{aligned} R_{\Phi,i}(s) := R_{\Phi,i}(\rho & \sum_{j\in O_i} W_{ij}\,x_j(k,s) - \mu_i(k)\,, \\ \rho & \sum_{j\in O_i} W_{ij}\,x'_j(k+1) - \mu_i(k)), \end{aligned}$$

and squaring the equality, we obtain:

$$
\begin{aligned}
& (x_i(k,s+1) - x_i'(k+1))^2 \\
= \;& (R_{\Phi,i}(s))^2 \, \rho^2 \left(\sum_{j \in O_i} W_{ij} \, (x_j(k,s) - x_j'(k+1)) \right)^2 \\
\leq \;& \left(\frac{\rho}{\rho + h_{\min}} \right)^2 \sum_{j \in O_i} W_{ij} \, (x_j(k,s) - x_j'(k+1))^2 \quad (59) \\
= \;& \delta^2 \sum_{j=1}^{N} W_{ij} \; (x_j(k,s) - x_j'(k+1))^2. \quad (60)
\end{aligned}
$$

Here, (59) further uses: 1) convexity of the quadratic function $u \mapsto u^2$; 2) the fact that $\sum_{j \in O_i} W_{ij} = 1$; and 3) the fact that the W_{ij}'s are nonnegative. Also, (60) introduces notation: $\delta := \frac{\rho}{\rho + h_{\min}}$, and uses the fact that $W_{ij} = 0$ if $\{i,j\} \notin E$ and $i \neq j$. As node i is selected, the remaining quantities $x_j(k,s)$, $j \neq i$, remain unchanged; i.e., $x_j(k,s+1) - x_j'(k+1) = x_j(k,s) - x_j'(k+1)$, $j \neq i$. Squaring the latter equalities, adding them up for all $j \neq i$, and finally adding them to (60), we obtain:

$$
\begin{aligned}
& \|x(k,s+1) - x'(k+1)\|^2 \\
\leq \;& \|x(k,s) - x'(k)\|^2 \\
+ \;& \delta^2 \sum_{j=1}^{N} W_{ij} \, (x_j(k,s) - x_j'(k+1))^2 \\
- \;& (x_i(k,s) - x_i'(k+1))^2, \quad (61)
\end{aligned}
$$

for any $\omega \in \mathcal{A}_{k,j}$ such that $\widehat{\imath}(k,s) = i$.

We now compute conditional expectation of $\|x(k,s+1) - x'(k+1)\|^2$, conditioned on $\tau(k) = j$, $x(k) = x(k,0)$, $\mu(k)$, and $x(k,1), ..., x(k,s)$, $s \leq j-1$. Conditioned on the latter, each node i updates equally likely, with conditional probability $1/N$, and therefore:

$$
\begin{aligned}
& \mathbb{E}\left[\, \|x(k,s+1) - x'(k+1)\|^2 \; | \, x(k), \mu(k), \tau(k) \right. \\
& \left. = j, x(k,1), ..., x(k,s) \, \right] \\
& \leq \|x(k,s) - x'(k+1)\|^2 + \\
& \frac{1}{N} \, \delta^2 \sum_{i=1}^{N} \sum_{j=1}^{N} W_{ij} \, (x_j(k,s) - x_j'(k+1))^2 \\
& - \frac{1}{N} \sum_{i=1}^{N} (x_i(k,s) - x_i'(k+1))^2 \\
& = \|x(k,s) - x'(k+1)\|^2 \\
& + \frac{1}{N} \, \delta^2 \sum_{j=1}^{N} (x_j(k,s) - x_j'(k+1))^2 \, \sum_{i=1}^{N} W_{ij} \\
& - \frac{1}{N} \|x(k,s) - x'(k+1)\|^2 \quad (62) \\
& = \|x(k,s) - x'(k+1)\|^2 + \frac{1}{N} \delta^2 \|x(k,s) - x'(k+1)\|^2 \\
& - \frac{1}{N} \|x(k,s) - x'(k+1)\|^2, \; \forall \omega \in \mathcal{A}_{k,j}. \quad (63)
\end{aligned}
$$

Here, inequality (63) uses the fact that $\sum_{i=1}^{N} W_{ij} = 1$, $\forall j$. Rewriting (63), we get:

$$
\begin{aligned}
& \mathbb{E}\left[\, \|x(k,s+1) - x'(k+1)\|^2 \right| \\
& \qquad \left| x(k), \mu(k), \tau(k) = j, x(k,1), ..., x(k,s) \right] \\
& \leq \left(1 - \frac{1}{N}(1 - \delta^2) \right) \|x(k,s) - x'(k+1)\|^2, \forall \omega \in \mathcal{A}_{k,j}.
\end{aligned}
$$

Denote by $\delta' := \left(1 - \frac{1}{N}(1 - \delta^2)\right)^{1/2}$. Using the following inequality for quadratic convex functions and conditional expectation: $\mathbb{E}[U^2 \,|\, V] \geq \mathbb{E}^2[|U| \;|\, V]$, we obtain:

$$
\begin{aligned}
& \mathbb{E}\left[\|x(k,s+1) - x'(k+1)\| | x(k), \mu(k), \tau(k) = j, \right. \\
& \qquad \qquad \left. x(k,1), ..., x(k,s) \right] \\
& \leq \delta' \, \|x(k,s) - x'(k+1)\| \, , \forall \omega \in \mathcal{A}_{k,j}.
\end{aligned}
$$

Integrating with respect to $x(k,1), ..., x(k,s)$:

$$
\begin{aligned}
& \mathbb{E}\left[\|x(k,s+1) - x'(k+1)\| | x(k), \mu(k), \tau(k) = j \right] \\
& \leq \delta' \mathbb{E}\left[\|x(k,s) - x'(k+1)\| \, | x(k), \mu(k), \tau(k) = j \right], \\
& \qquad \qquad \forall \omega \in \mathcal{A}_{k,j}.
\end{aligned}
$$

Applying the above inequality for $s = 0, 1, ..., j-1$, and using $x(k, s = \tau(k) = j) = x(k+1)$:

$$
\begin{aligned}
& \mathbb{E}\left[\, \|x(k+1) - x'(k+1)\| \, | \, x(k), \mu(k), \tau(k) = j \, \right] \\
& \leq (\delta')^j \mathbb{E}\left[\|x(k) - x'(k+1)\| \, | x(k), \mu(k), \tau(k) = j \right], \\
& \qquad \qquad \forall \omega \in \mathcal{A}_{k,j}, \forall j = 0, 1, ...,
\end{aligned}
$$

and so:

$$
\begin{aligned}
& \mathbb{E}\left[\, \|x(k+1) - x'(k+1)\| \, | \, x(k), \mu(k), \tau(k) \, \right] \\
& \leq (\delta')^{\tau(k)} \mathbb{E}\left[\|x(k) - x'(k+1)\| \, | x(k), \mu(k), \tau(k) \right], \\
& \text{almost surely (a.s.)}
\end{aligned}
$$

Integrating with respect to $x(k), \mu(k)$:

$$
\begin{aligned}
& \mathbb{E}\left[\|x(k+1) - x'(k+1)\| | \tau(k) \right] \\
& \leq (\delta')^{\tau(k)} \mathbb{E}\left[\|x(k) - x'(k+1)\| \, | \tau(k) \right] \\
& = (\delta')^{\tau(k)} \mathbb{E}\left[\|x(k) - x'(k+1)\| \right], \text{a.s.},
\end{aligned}
$$

where we used independence of $\tau(k)$ and $x(k), \mu(k)$. Taking expectation, we obtain:

$$
\begin{aligned}
& \mathbb{E}\left[\, \|x(k+1) - x'(k+1)\| \right] \\
& \leq \mathbb{E}\left[\, (\delta')^{\tau(k)} \, \right] \, \mathbb{E}\left[\|x(k) - x'(k+1)\| \right].
\end{aligned}
$$

Because $\tau(k)$ is distributed according to the Poisson distribution with parameter $N\tau$, we have: $\mathbb{E}\left[(\delta')^{\tau(k)}\right] = \sum_{l=0}^{\infty} (\delta')^l \frac{e^{-N\tau}(N\tau)^l}{l!} = e^{-(1-\delta')N\tau}$. We get:

$$
\begin{aligned}
& \mathbb{E}\left[\, \|x(k+1) - x'(k+1)\| \, \right] \leq \quad (64) \\
& e^{-(1-\delta')N\tau} \, \mathbb{E}\left[\|x(k) - x'(k+1)\| \right].
\end{aligned}
$$

Substituting the expression for η, we obtain the desired result. ■

Consider Theorem 1. Note that it does not apply directly to the randomized algorithm variants. However, it can be easily adapted to the randomized variants as well. Namely, consider the following random inexact AL method. Use the same initialization as for (14)–(15). Given $x(k)$, $\mu(k)$, define (as before) $x'(k+1) := x'(\mu(k)) := \arg\min_x L_a(x;\mu(k))$. The primal update is as follows: let $x(k+1)$ be a random variable that obeys $\mathbb{E}[\|x(k+1)-x'(k+1)\|] \le \xi\, \mathbb{E}[\|x(k)-x'(k+1)\|]$. (This replaces (14) in Theorem 1.) The dual update is the same as in (15). Then, it is straightforward to show that, under condition (16), the following holds: $\mathbb{E}[\, \|x_i(k) - x^\star\| \,] \le r^k\, \sqrt{N} \max\left\{ D_x, \frac{2D_\mu}{\sqrt{\lambda_2(\mathcal{L})}h_{\min}} \right\}$, where r is in (17). Now, applying Lemma 7, the last result holds for the randomized Gauss-Seidel-type variant, with $\xi = e^{-\eta\,\tau}$. It turns out that an analogous conclusion also holds for the randomized gradient variant, with η replaced by η', defined in the following Lemma.

Lemma 8 (Randomized gradient) Consider the distributed AL algorithm with randomized gradient primal updates, let the expected number of inner iterations equal τ, and ler the primal step-size $\beta \le 1/(h_{\max}+\rho)$. Further, let Assumptions 1 and 2 hold. Then, for all $k = 0, 1, \cdots$:

$$\mathbb{E}\left[\|x(k+1) - x'(k+1)\|\right] \le e^{-\eta'\tau}\mathbb{E}\left[\|x(k) - x'(k+1)\|\right],$$

where

$$\eta' := N\left\{1 - \left[1 - \frac{1}{N}\beta h_{\min}(2-\beta h_{\min})\right]^{1/2}\right\}. \qquad (65)$$

The proof of Lemma 8 is similar to that of Lemma 7. For the randomized algorithm and gradient updates, (59)–(60) hold with $\frac{\rho^2}{(\rho+h_{\min})^2}$ replaced by $(1-\beta\, h_{\min})^2$.

V. Simulation example

We provide a simulation example with l_2-regularized logistic losses. The simulations corroborate a globally linear convergence for both the deterministic and randomized distributed AL methods, and show that it is usually advantageous to take a small number of inner iterations τ.

Optimization problem. We detail the simulation. We consider distributed learning via the l_2-regularized logistic loss; see, e.g., [56] for further details. Nodes minimize the logistic loss:

$$\sum_{i=1}^N f_i(x) = \sum_{i=1}^N \left(\log\left(1 + e^{-b_i(a_i^\top x_1 + x_0)}\right) + \frac{\mathcal{P}\,\|x\|^2}{2\,N} \right),$$

where $\mathcal{P} > 0$ is the regularization parameter, $x = (x_1^\top, x_0)^\top \in \mathbb{R}^{15}$, $a_i \in \mathbb{R}^{14}$ is the node i's feature vector, and $b_i \in \{-1,+1\}$ is its class label. The Hessian $\nabla^2 f_i(x) = \frac{\mathcal{P}}{N}I + \frac{e^{-c_i^\top x}}{(1+e^{-c_i^\top x})^2}c_i c_i^\top$, where $c_i = (b_i a_i^\top, b_i)^\top \in \mathbb{R}^{15}$. We take node i's constants $h_{\min,i}$ and $h_{\max,i}$ as: $h_{\min,i} = \frac{\mathcal{P}}{N}$ and $h_{\max,i} = \frac{\mathcal{P}}{N} + \frac{1}{4}\,\|c_i\, c_i^\top\|$. (Note that $\frac{e^{-c_i^\top y}}{(1+e^{-c_i^\top y})^2} \le 1/4$ for all y.) Further, we let $h_{\min} = \min_{i=1,\cdots,N} h_{\min,i}$ and $h_{\max} = \max_{i=1,\cdots,N} h_{\max,i}$. For the specific problem instance here, the condition number $\gamma = h_{\max}/h_{\min} = 49.55$.

Data. The a_i's are independent over i. Their entries and the entries of the "true" vector $x^\star = ({x_1^\star}^\top, x_0^\star)^\top$ are independent standard normal. The class labels are $b_i = \text{sign}\left({x^\star}_1{}^\top a_i + {x^\star}_0 + \epsilon_i\right)$, where the ϵ_i's are independent zero mean, standard deviation 0.001, Gauss.

Network. The network is geometric, connected, with 10 nodes placed uniformly randomly on a unit square, connected by an edge (28 links) if their distance less than a radius.

Algorithm parameters, metrics, and implementation. We set the weight matrix $W = \frac{1.1}{2}I + \frac{0.9}{2}W_m$, where W_m is the Metropolis weight matrix. (Note that $W \succ 0$.) Further, $\alpha = \rho = h_{\min}$ with all algorithm variants, and $\beta = \frac{1}{\rho+h_{\max}} = \frac{1}{(\gamma+1)h_{\min}}$ with the methods that use the gradient primal updates. For the deterministic variant and Jacobi-type updates, we set the number of inner iterations $\tau = \left\lceil \frac{\log\left(\frac{3(1+\gamma)}{\lambda_2(\mathcal{L})}\right)}{\log(2)} \right\rceil$; with the deterministic gradient variant $\tau = \left\lceil \frac{\log\left(\frac{3(1+\gamma)}{\lambda_2(\mathcal{L})}\right)}{\log\left(\frac{\gamma+1}{\gamma}\right)} \right\rceil$; with the randomized Gauss-Seidel-type variant $\tau = \left\lceil \frac{\left|\log\left(\frac{3(1+\gamma)}{\lambda_2(\mathcal{L})}\right)\right|}{N\left(1-(1-3/(4\,N))^{1/2}\right)} \right\rceil$; and with the randomized gradient variant $\tau = \left\lceil \frac{\left|\log\left(\frac{3(1+\gamma)}{\lambda_2(\mathcal{L})}\right)\right|}{N\left(1-\left(1-\frac{1+2\gamma}{N(1+\gamma)^2}\right)^{1/2}\right)} \right\rceil$. The above values of the algorithm parameters α, β, ρ, and τ satisfy conditions of Theorem 1 and Lemmas 5–8, and hence they guarantee linear convergence rates. We also simulate the methods with $\tau = 1$ (although our theory does not guarantee linear convergence in such case.) We initialize from zero the primal and dual variables with all methods. We consider $\frac{1}{N}\sum_{i=1}^N \frac{f(x_i)-f^\star}{f(0)-f^\star}$. We compare the methods in terms of: 1) total number of transmissions (across all nodes), and 2) total computational time. We implement the methods via a serial implementation – one processor works the jobs of all nodes. We count the CPU time for the overall jobs across all nodes. With the methods that use the Gauss-Seidel and Jacobi-type updates in (6), we solve the local problems via the fast Nesterov gradient method for strongly convex functions. At the inner iteration s and outer iteration k, to solve (6), we initialize the Nesterov gradient method by $x_i(k,s)$. We stop the algorithm after: $\left\lceil \left| \frac{\log\left(\frac{2\epsilon}{(R')^2 L'}\right)}{\log(1-\sqrt{\gamma'})} \right| \right\rceil$ iterations, with[7] $\epsilon = 10^{-5}$. This guarantees that the optimality gap upon termination is below $\epsilon = 10^{-5}$. Here, L' is a Lipschitz constant for the cost function in (6) that (at node i) we take as $h_{\max,i} + \rho$. Further, $\gamma' = L'/\nu'$ is the cost condition number, where $\nu' = h_{\min,i}+\rho$ is the Hessian lower bound. The estimate of the distance to the solution is $R' = \frac{1}{\rho+\mathcal{P}/N}\|\nabla \widehat{f}_i(x_i(k,s)) + (\mathcal{P}/N+\rho)x_i(k,s) + (\mu_i(k) - \rho\overline{x}_i(k,s))\,\|$, $\widehat{f}_i(x) = \log(\,1+\exp(-b_i(a_i^\top x_1 + x_0))\,)$. All Figures are in semi-log scale.

In Figure 1 (top left), we plot the relative error in the cost function for the deterministic variants versus the number of communications, while in Figure 1 (top right), we depict the same quantity versus the CPU time (This is the cumulative CPU time across all nodes.) We simulate the Jacobi-type method with both theoretical value of τ and $\tau = 1$, and the gradient method with both theoretical value of τ and $\tau = 1$. The Figures illustrate the linear convergence of the proposed methods. We report that the gradient method with the theoretical value of τ also shows a linear convergence in

[7] We implicitly assume that the physical time allocated for each inner iteration s suffices to perform optimization (6).

the number of communications, but it converges slowly due to the large value of τ. The Jacobi-type variant is better in terms of communication cost but is worse in terms of computational cost. Figures 1 (bottom left and right) present the same plots for the randomized Gauss-Seidel-type and gradient-type methods. The behavior is similar to the deterministic variants. The theoretical value for τ of the randomized gradient method is very large, and, consequently, the algorithm shows slow convergence for the latter choice of τ.

VI. Conclusion

We consider distributed optimization where N nodes minimize the sum of their convex costs f_i's by four distributed augmented Lagrangian (AL) methods that differ in the primal variable updates: 1) deterministic AL with Jacobi-type updates; 2) deterministic AL with gradient descent; 3) randomized AL with nonlinear Gauss-Seidel-type; and 4) randomized AL with gradient descent-type updates. With twice continuously differentiable costs with bounded Hessian, we establish globally linear (geometric) convergence rates for all methods and give explicit dependence of the rates on the underlying network parameters. Simulation examples demonstrate linear convergence of our methods.

References

[1] S. Kar, J. M. F. Moura, and K. Ramanan, "Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3575–3605, June 2012.

[2] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *IPSN 2004, 3rd International Symposium on Information Processing in Sensor Networks*, Berkeley, California, USA, April 2004, pp. 20 – 27.

[3] J. Mota, J. Xavier, P. Aguiar, and M. Pueschel, "Distributed basis pursuit," *IEEE Trans. Sig. Process.*, vol. 60, no. 4, pp. 1942–1956, July 2012.

[4] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 11, pp. 5262–5276, November 2010.

[5] J. A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1847–1862, March 2010.

[6] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links – Part I: Distributed estimation of deterministic signals," *IEEE Trans. Sig. Process.*, vol. 56, no. 1, pp. 350–364, Jan. 2009.

[7] ——, "Consensus in ad hoc WSNs with noisy links – Part I: Distributed estimation and smoothing of random signals," *IEEE Trans. Sig. Process.*, vol. 56, no. 4, pp. 1650–1666, April 2009.

[8] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Cooperative convex optimization in networked systems: Augmented Lagrangian algorithms with directed gossip communication," *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pp. 3889–3902, August 2011.

[9] 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)*, Milano, Italy, August 2011, identifier: 10.3182/20110828-6-IT-1002.01959.

[10] W. Shi, Q. Ling, G. Wu, and W. Yin, "On the linear convergence of ADMM in decentralized consensus optimization," *to appear IEEE Trans. Sig. Process.*, DOI: 10.1109/TSP.2014.2304432.

[11] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "Linearly convergent decentralized consensus optimization with the alternating direction method of multipliers," in *ICASSP 2013, IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vancouver, Canada, May 2013, pp. 4613–4617.

[12] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, January 2009.

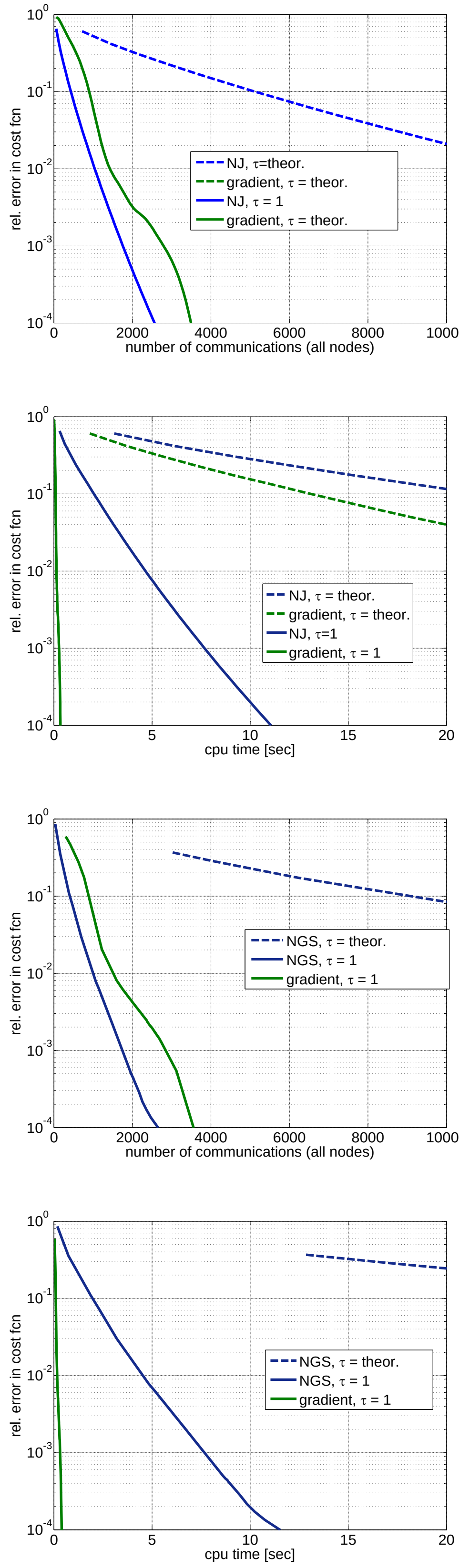

Fig. 1. Deterministic (two top most) and randomized (two bottom right) AL methods: Average relative error in the cost function $\frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)-f^\star}{f(0)-f^\star}$. First and third plots: communication cost (total number of communications across all nodes). Second and fourth plots: computational cost (total CPU time across all nodes.) NJ–Jacobi; NGS–Gauss-Seidel.

[13] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," 2013, available at: http://arxiv.org/abs/1310.7063.

[14] I. Matei and J. S. Baras, "Performance evaluation of the consensus-based distributed subgradient method under random communication topologies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 754–771, 2011.

[15] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *CDC 2012, IEEE International Conference on Decision and Control*, Maui, Hawaii, Dec. 2012, pp. 5445–5450.

[16] E. Ghadimi, M. Johansson, and I. Shames, "Accelerated gradient methods for networked optimization," November 2012, arxiv post: arxiv.org/abs/1211.2132.

[17] T. Erseghe, D. Zennaro, E. Dall'Anese, and L. Vangelista, "Fast consensus by the alternating direction multipliers method," *IEEE Trans. Sig. Process.*, vol. 59, no. 11, pp. 5523–5537, Nov. 2011.

[18] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, Jan. 2012.

[19] B. Gharesifard and J. Cortes, "Distributed continuous-time convex optimization on weighted-balanced digraphs," 2012, available at: arxiv.org/abs/1204.0304.

[20] J. Wang and N. Elia, "Control approach to distributed optimization," in *48th Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Oct. 2010, pp. 557–561.

[21] ——, "A control perspective to centralized and distributed convex optimization," in *50th CDC Conference on Decision and Control*, Orlando, Florida, Dec. 2011, pp. 3800–3805.

[22] G. Lan and R. D. C. Monteiro, "Iteration-complexity of first-order augmented Lagrangian methods for convex programming," 2008, technical Report, School of Industrial and Systems Engineering, Georgia Institute of Technology.

[23] V. Nedelcu, I. Necoara, and Q. T. Dinh, "Computational complexity of inexact gradient augmented Lagrangian methods: Application to constrained MPC," 2013, available at: arxiv.org/abs/1302.4355.

[24] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293–318, 1992.

[25] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximations," *Computers and Mathematics with Applications*, vol. 2, pp. 17–40, 1976.

[26] R. Glowinski and A. Marrocco, "Sur l´approximation, par éléments finis d´ordre 1, et la résolution, par pénalisation-dualité, d´une classe de problémes de Dirichlet non-linéaires," *Revue Francaise d´Automatique, Informatique, et Recherche Opérationelle*, vol. 9, pp. 41–76, 1975.

[27] Z. Q. Luo and P. Tseng, "On the linear convergence of descent methods for convex essentially smooth optimization," *SIAM J. Control and Optimization*, vol. 30, no. 2, pp. 408–425, 1992.

[28] B. W. Kort and D. P. Bertsekas, "Combined primal-dual and penalty methods for convex programming," *Siam J. Control and Optimization*, vol. 14, no. 2, pp. 268–294, Feb. 1976.

[29] R. T. Rockafellar, "Augmented Lagrangian and applications of the proximal point algorithm in convex programming," *Math. Oper. Res.*, vol. 1, pp. 97–116, 1976.

[30] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," 2012, arxiv post: arxiv.org/abs/1208.3922.

[31] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," 2012, Rice University CAAM Technical Report TR12-14.

[32] J.-B. H. Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms I: Fundamentals*. Springer Verlag, 1996.

[33] C. Lemaréchal, "Lagrangian relaxation," *Lecture Notes in Computer Science, Springer*, vol. 2241, pp. 112–156, 2001.

[34] M. R. Hestenes, "Multiplier and gradient methods," *Jour. Opt. Theory Appl.*, vol. 4, pp. 302–320, 1969.

[35] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*. Optimization (R. Fletcher, ed.), Academic Press, 1969.

[36] K. Arrow, L. Hurwicz, and H. Uzawa, *Studies in Linear and Nonlinear Programming*. Stanford University Press, Stanford, CA, 1958.

[37] E. Gol'shtein, "A generalized gradient method for finding saddle points," *Matekon*, vol. 10, pp. 36–52, 1974.

[38] D. Maistroskii, "Gradient methods for finding saddle points," *Matekon*, vol. 13, pp. 3–22, 1977.

[39] M. Kallio and A. Ruszczynski, "Perturbation methods for saddle point computation," *Tech. Report WP-94-38, International Institute for Applied Systems Analysis*, 1994.

[40] A. Nedic and A. Ozdaglar, "Subgradient methods for saddle point problems," *Journal of Optimization Theory and Applications*, vol. 142, no. 1, pp. 205–208, 2009.

[41] X. Chen, "Global and superlinear convergence of inexact Uzawa methods for saddle point problems with nondifferentiable mappings," 1995, dOI:10.1137/S0036142995295789.

[42] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, "Analysis of the inexact Uzawa algorithm for saddle point problems," *SIAM J. Numer. Anal.*, vol. 34, pp. 1072–1092, 1997.

[43] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," 2010.

[44] Q. Hu and J. Zou, "Nonlinear inexact Uzawa algorithms for linear and nonlinear saddle point problems," *SIAM J. Optim.*, vol. 16, no. 3, pp. 798–825, 2001.

[45] J. Lu, "Convergence analysis of the modified nonlinear inexact Uzawa algorithm for saddle point problem," *Int. J. Contemp. Math. Sciences*, vol. 7, no. 22, pp. 1067–1075, 2012.

[46] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. New York/London: (Computer Science and Applied Mathematics), Academic Press, 1970.

[47] E. Kaszkurewicz and A. Bhaya, *Matrix Diagonal Stability in Systems and Computation*. New York: Springer Science+Business Media, 2000.

[48] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation*. New Jersey: Prentice-Hall, Englewood Cliffs, 1989.

[49] M. N. E. Tarazi, "Some convergence results for asynchronous algorithms," *Numerische Mathematik*, vol. 39, pp. 325 –340, 1982.

[50] D. Chazan and W. L. Miranker, "Chaotic relaxation," *Linear Algebra and its Applications*, vol. 2, pp. 190–222, 1969.

[51] A. F. Kleptsyn, V. S. Kozyakin, M. A. Krasnoselśkii, , and N. A. Kuznetsov, "Effect of small synchronization errors on stability of complex systems. I," *Automation and Remote Control*, vol. 44, no. 7.

[52] ——, "Effect of small synchronization errors on stability of complex systems. II," *Automation and Remote Control*, vol. 45, no. 3.

[53] ——, "Effect of small synchronization errors on stability of complex systems. III," *Automation and Remote Control*, vol. 45, no. 8.

[54] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," 2011, available at: http://arxiv.org/abs/1112.2972.

[55] J. E. Marsden and A. J. Tromba, *Vector Calculus*. Freeman and Company, New York, 1996.

[56] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning, Michael Jordan, Editor in Chief*, vol. 3, no. 1, pp. 1–122, 2011.

Dušan Jakovetić (S'10) obtained a dipl. ing. diploma from the School of Electrical Engineering, University of Belgrade, in August 2007, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, and Instituto de Sistemas e Robótica (ISR), Instituto Superior Técnico (IST), Lisbon, Portugal, in May 2013. Since October 2013, he has been a research fellow at the BioSense Center, University of Novi Sad, Serbia. From June to September 2013, he was a postdoctoral researcher at IST. His research interests include distributed inference and distributed optimization.

José M. F. Moura (S'71–M'75–SM'90–F'94) received the engenheiro electrotécnico degree from Instituto Superior Técnico (IST), Lisbon, Portugal, and the M.Sc., E.E., and D.Sc. degrees in EECS from MIT, Cambridge, MA.

In 2013-14, he is a visiting Professor at New York University (NYU) and at CUSP-NYU on sabbatical leave from Carnegie Mellon University (CMU) where he is the Philip and Marsha Dowd University Professor. Previously, he was on the faculty at IST and was visiting Professor at MIT. He is founding director of ICTI@CMU, a large education and research program between CMU and Portugal, www.cmuportugal.org. His research interests include statistical, algebraic, and distributed signal and image processing and signal processing on graphs. He has published over 470 papers, has ten patents issued by the US Patent Office, and cofounded SpiralGen.

Dr. Moura is IEEE Division IX Director and member of the IEEE Board of Directors (2012-13) and has served on several IEEE Boards. He was *President* (2008-09) of the *IEEE Signal Processing Society*(SPS), served as *Editor in Chief* for the *IEEE Transactions in SP*, interim *Editor in Chief* for the *IEEE SP Letters*, and member of several Editorial Boards, including *IEEE Proceedings*, *IEEE SP Magazine*, and the ACM *Transactions on Sensor Networks*.

Dr. Moura is member of the *US National Academy of Engineering*, corresponding member of the *Academy of Sciences of Portugal*, *Fellow* of the *IEEE*, and *Fellow* of the *AAAS*. He received the IEEE Signal Processing Society *Technical Achievement Award* and the IEEE Signal Processing Society *Society Award*.

João Xavier (S'97–M'03) received the Ph.D. degree in Electrical and Computer Engineering from Instituto Superior Técnico (IST), Lisbon, Portugal, in 2002. Currently, he is an Assistant Professor in the Department of Electrical and Computer Engineering, IST. He is also a Researcher at the Institute of Systems and Robotics (ISR), Lisbon, Portugal. His current research interests are in the area of optimization and statistical inference for distributed systems.

Preprint submitted to IEEE Transactions on Automatic Control. Received: October 8, 2014 17:09:00 PST