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

Optimal scaling of the ADMM algorithm for distributed quadratic programming

André Teixeira, Euhanna Ghadimi, Iman Shames, Henrik Sandberg, and Mikael Johansson

Abstract— **This paper addresses the optimal scaling of the ADMM method for distributed quadratic programming. Scaled ADMM iterations are first derived for generic equality-constrained quadratic problems and then applied to a class of distributed quadratic problems. In this setting, the scaling corresponds to the step-size and the edge-weights of the underlying communication graph. We optimize the convergence factor of the algorithm with respect to the step-size and graph edge-weights. Explicit analytical expressions for the optimal convergence factor and the optimal step-size are derived. Numerical simulations illustrate our results.**

I. Introduction

Recently, a number of applications have triggered a strong interest in distributed algorithms for large-scale quadratic programming. These applications include multi-agent systems [1], [2], distributed model predictive control [3], [4], and state estimation in networks [5], to name a few. As these systems become larger and their complexity increases, more efficient algorithms are required. It has been argued that the alternating direction method of multipliers (ADMM) is a particularly powerful and efficient approach [6]. One attractive feature of ADMM is that it is guaranteed to converge for all (positive) values of its step-size parameter [6]. This contrasts many alternative techniques, such as dual decomposition, where mistuning of the step-size for the gradient updates can render the iterations unstable.

The ADMM method has been observed to converge fast in many applications [6]–[9] and for certain classes of problems it has a guaranteed linear rate of convergence [10]–[12]. However, the solution times are sensitive to the choice of the step-size parameter, and the ADMM iterations can converge (much) slower than the standard gradient algorithm if the parameter is poorly tuned. Recently, [11] provides optimal tuning parameters for the general (centralized) class of quadratic programming with linear inequality constraints. For distributed programming, however, the ADMM algorithm is tuned empirically for each specific application. In particular, [7]–[9] report various rules of thumb for picking the step-size. A thorough analysis and design of optimal step-size and scaling rules for the ADMM algorithm is largely unaddressed in the literature. The aim of this paper is to close this gap for a class of distributed quadratic programming problems.

We first consider a particular class of equality-constrained quadratic programming problems and derive the corresponding iterations for the ADMM method. The iterations are shown to be linear and the corresponding eigenvalues are characterized as roots of quadratic polynomials. These results are then used to develop optimally scaled ADMM iterations for a class of distributed quadratic programming problems that appear in power network state-estimation applications [13]. In this class of problems, a number of agents collaborate with neighbors in a graph to minimize a convex objective function with a specific sparsity structure over a mix of shared and private variables. We show that quadratic programming problems with this structure can be reduced to an equality constrained convex quadratic programming problem in terms of private variables only. The ADMM iterations for this quadratic problem are then formulated taking into account the communication network constraints. The network-constrained scaling of the ADMM method includes the step-size and edge weights of the communication graph. Methods to minimize the convergence factor by optimal scaling of the ADMM iterations are proposed for generic connected graphs. In particular, analytical expressions for the optimal step-size and convergence factor are derived in terms of the spectral properties of the communication graph. A tight lower-bound for the convergence factor is also obtained. Finally, given that the optimal step-size is chosen, we propose methods to further minimize the convergence factor by optimizing the edge weights.

The outline of this paper is as follows. Section II gives an elementary background to the ADMM method. The ADMM iterations for a class of equality-constrained quadratic programming problems are formulated and analyzed in Section III. Distributed quadratic programming and optimal networked-constrained scaling of the ADMM algorithm are addressed in Section IV. Numerical examples illustrating our results and comparing them to state-of-the art techniques are presented in Section V. Finally, a discussion and outlook on future research concludes the paper.

A. Notation

We denote the set of real and complex numbers with $\mathbf{R}$, and $\mathbf{C}$, respectively. For a given matrix $A \in \mathbf{R}^{n\times m}$, denote $\mathcal{R}(A) \triangleq \{y \in \mathbf{R}^n |\ y = Ax,\, x \in \mathbf{R}^m\}$ as its range-space and let $\mathcal{N}(A) \triangleq \{x \in \mathbf{R}^m |\ Ax = 0\}$ be the null-space of A. For A with full-column rank, define $A^\dagger \triangleq (A^\top A)^{-1}A^\top$ as the pseudo-inverse of A and $\Pi_{\mathcal{R}(A)} \triangleq AA^\dagger$ as the

A. Teixeira, E. Ghadimi, H. Sandberg, and M. Johansson are with the ACCESS Linnaeus Center, Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden. I. Shames is with the Department of Electrical and Electronic Engineering, University of Melbourne, Australia. This work was sponsored in part by the Swedish Foundation for Strategic Research (SSF), the Swedish Research Council (VR), a UoM Early Career Researcher Grant, and a McKenzie Fellowship. {andretei,euhanna,hsan,mikaelj}@kth.se, iman.shames@unimleb.edu.au.

978-1-4673-5716-6/13/$31.00 ©2013 IEEE

orthogonal projector onto $\mathcal{R}(A)$. Since $\mathcal{R}(A)$ and $\mathcal{N}(A^\top)$ are orthogonal complements, we have $\Pi_{\mathcal{N}(A^\top)} = I - \Pi_{\mathcal{R}(A)}$ and $\Pi_{\mathcal{R}(A)}\Pi_{\mathcal{N}(A^\top)} = 0$. Now consider $A, D \in \mathbf{R}^{n\times n}$, with D invertible. The generalized eigenvalues of (A, D) are defined as the values $\lambda \in \mathbf{C}$ such that $(A - \lambda D)v = 0$ holds for some nonzero vector $v \in \mathbf{C}^n$. Additionally, $A \succ 0$ ($A \succeq 0$) denotes that A is positive definite (semi-definite).

Consider the sequence $\{x^k\}$ converging to the fixed-point $x^\star$. The convergence factor of the converging sequence is defined as [14]

$$\phi^\star \triangleq \sup_{x^k \neq x^\star} \frac{\|x^{k+1} - x^\star\|}{\|x^k - x^\star\|}. \tag{1}$$

Definitions from graph theory are now presented [15]. Let $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$ be a connected weighted undirected graph with vertex set $\mathcal{V}$ with n vertices, edge set $\mathcal{E}$ with m edges, and edge-weights $\mathcal{W}$. Each vertex $i \in \mathcal{V}$ represents an agent, and an undirected edge $e_k = (i, j) \in \mathcal{E}$ means that agents i and j can exchange information. Letting $w_{e_k} \geq 0$ be the weight of e_k, the edge-weight matrix is defined as $W \triangleq \mathrm{diag}([w_{e_1} \dots w_{e_m}])$. The weighted graph is connected if the unweighted subgraph obtained from $\mathcal{G}$ by removing the edges with zero weights, $\tilde{\mathcal{G}}(\mathcal{V}, \tilde{\mathcal{E}})$ with $\tilde{\mathcal{E}} \triangleq \{e_k \in \mathcal{E} : w_{e_k} > 0\}$, is connected. Denote $\mathcal{N}_i \triangleq \{j \neq i | (i, j) \in \mathcal{E}\}$ as the neighbor set of node i. Define $\mathcal{A}$ as the span of real symmetric matrices, $\mathcal{S}^n$, with sparsity pattern induced by $\mathcal{G}$, $\mathcal{A} \triangleq \{S \in \mathcal{S}^n | S_{ij} = 0 \, \text{if} \, i \neq j \, \text{and} \, (i, j) \notin \mathcal{E}\}$. The adjacency matrix $A \in \mathcal{A}$ is defined as $A_{ij} = A_{ji} = w_{e_k}$ for $e_k = (i, j) \in \mathcal{E}$ and $A_{ii} = 0$. The diagonal degree matrix D is given by $D_{ii} = \sum_{j \in \mathcal{N}_i} A_{ij}$. The incidence matrix $B \in \mathbf{R}^{m\times n}$ is defined as $B_{ij} = 1$ if the edge e_i is incident to node $j \in \mathcal{V}$ and $B_{ij} = 0$ otherwise

II. The ADMM method

The ADMM algorithm solves problems of the form

$$\begin{array}{ll} \underset{x,\, z}{\text{minimize}} & f(x) + g(z) \\ \text{subject to} & Ex + Fz - h = 0 \end{array} \tag{2}$$

where f and g are convex functions, $x \in \mathbf{R}^n$, $z \in \mathbf{R}^m$, $h \in \mathbf{R}^p$. Moreover, $E \in \mathbf{R}^{p\times n}$ and $F \in \mathbf{R}^{p\times m}$ have full-column rank; see [6] for a detailed review. The method is based on the *augmented Lagrangian*

$$\begin{aligned} L_\rho(x, z, \mu) = f(x) + g(z) + (\rho/2)\|Ex + Fz - h\|_2^2 \quad (3) \\ + \mu^\top(Ex + Fz - h) \end{aligned}$$

and performs sequential minimization of the x and z variables, followed by a dual variable update:

$$\begin{aligned} x^{k+1} &= \underset{x}{\operatorname{argmin}}\, L_\rho(x, z^k, \mu^k) \\ z^{k+1} &= \underset{z}{\operatorname{argmin}}\, L_\rho(x^{k+1}, z, \mu^k) \\ \mu^{k+1} &= \mu^k + \rho(Ex^{k+1} + Fz^{k+1} - h). \end{aligned} \tag{4}$$

These iterations indicate that the method is particularly useful when the x- and z-minimizations can be carried out efficiently (e.g. admit closed-form expressions). One advantage of the method is that there is only one single algorithm parameter, ρ, and under rather mild conditions, the method can be shown to converge for all values of the parameter; see, e.g., [6]. However, ρ has a direct impact on the convergence speed of the algorithm, and inadequate tuning of this parameter may render the method very slow. In the remaining parts of this paper, we will derive explicit expressions for the step-size ρ that minimizes the convergence factor (1) for some particular classes of problems.

III. ADMM for a class of equality-constrained quadratic programming problems

In this section, we develop scaled ADMM iterations for a particular class of equality-constrained convex quadratic programming problems. In terms of the standard formulation (2), these problems have $f(x) = \frac{1}{2}x^\top Qx + q^T x$ with $Q \succ 0$ and $q \in \mathbf{R}^n$, $g(z) = 0$, and $h = 0$. While this formulation may appear restrictive, it covers a vast area of distributed applications from multi-agent and estimation to networked-control systems (c.f. [2], [5], [16]).

An important difference compared to the standard ADMM iterations described in the previous section is the introduction of a matrix $R \in \mathbf{R}^{r\times p}$ scaling the equality constraints

$$R(Ex + Fz) = 0. \tag{5}$$

The underlying assumption on the choice of R is that all non-zero vectors $v = Ex + Fz, \forall x \in \mathbf{R}^n, z \in \mathbf{R}^m$ do not belong to the null-space of R. In other words, after the transformation (5) the feasible set in (2) remains unchanged. Taking into account the transformation in (5), the penalty term in the *augmented Lagrangian* becomes

$$\frac{1}{2}(Ex + Fz)^\top \rho R^\top R(Ex + Fz). \tag{6}$$

Definition 1: $\rho R^\top R$ is called the *scaling* of the augmented Lagrangian (3).

Our aim is to find the optimal scaling that minimizes the convergence factor of the corresponding ADMM iterations. Specifically, introducing $\bar{E} = RE$ and $\bar{F} = RF$, the scaled ADMM iterations read

$$\begin{aligned} x^{k+1} &= (Q + \rho\bar{E}^\top\bar{E})^{-1}\left(-q - \rho\bar{E}^\top(\bar{F}z^k + u^k)\right) \\ z^{k+1} &= -(\bar{F}^\top\bar{F})^{-1}\bar{F}^\top\left(\bar{E}x^{k+1} + u^k\right) \\ u^{k+1} &= u^k + \bar{E}x^{k+1} + \bar{F}z^{k+1}, \end{aligned} \tag{7}$$

where $u^k = \mu^k/\rho$. From the z- and u-iterations we observe

$$\begin{aligned} u^{k+1} &= (u^k + \bar{E}x^{k+1}) - \bar{F}(\bar{F}^\top\bar{F})^{-1}\bar{F}^\top\left(\bar{E}x^{k+1} + u^k\right) \\ &= \Pi_{\mathcal{N}(\bar{F}^\top)}\left(\bar{E}x^{k+1} + u^k\right). \end{aligned}$$

Since $\mathcal{N}(\bar{F}^\top)$ and $\mathcal{R}(\bar{F})$ are orthogonal complements, then we have $\Pi_{\mathcal{R}(\bar{F})}u^k = 0$ for all k, which results in

$$\bar{F}z^k = -\Pi_{\mathcal{R}(\bar{F})}\bar{E}x^k. \tag{8}$$

By induction the u-iterations can be rewritten as

$$u^{k+1} = \Pi_{\mathcal{N}(\bar{F}^\top)}\left(\sum_{i=1}^{k+1}(\bar{E}x^i) + u^0\right). \tag{9}$$

Supposing $u^0 = 0$, without loss of generality, and given (8) and (9), the x-iterations can be rewritten as

$$\begin{aligned} x^{k+1} &= (Q+\rho\bar{E}^\top\bar{E})^{-1}\left(-q+\rho\bar{E}^\top\Pi_{\mathcal{R}(\bar{F})}\bar{E}x^k\right) \\ &\quad - (Q+\rho\bar{E}^\top\bar{E})^{-1}\rho\bar{E}^\top\Pi_{\mathcal{N}(\bar{F}^\top)}\sum_{i=1}^{k}(\bar{E}x^i). \end{aligned}$$

Noting that

$$\begin{aligned} x^{k+1}-x^k &= (Q+\rho\bar{E}^\top\bar{E})^{-1}\rho\bar{E}^\top\Pi_{\mathcal{R}(\bar{F})}\bar{E}x^k \\ &\quad - (Q+\rho\bar{E}^\top\bar{E})^{-1}\rho\bar{E}^\top\Pi_{\mathcal{N}(\bar{F}^\top)}\bar{E}x^k \\ &\quad - (Q+\rho\bar{E}^\top\bar{E})^{-1}\rho\bar{E}^\top\Pi_{\mathcal{R}(\bar{F})}\bar{E}x^{k-1}, \end{aligned}$$

the iterations can be written in matrix form as

$$\begin{bmatrix} x^{k+1} \\ x^k \end{bmatrix} = \underbrace{\begin{bmatrix} M_{11} & M_{12} \\ I & 0 \end{bmatrix}}_{M}\begin{bmatrix} x^k \\ x^{k-1} \end{bmatrix}, \tag{10}$$

with

$$\begin{aligned} M_{11} &= \rho(Q+\rho\bar{E}^\top\bar{E})^{-1}\bar{E}^\top\left(\Pi_{\mathcal{R}(\bar{F})}-\Pi_{\mathcal{N}(\bar{F}^\top)}\right)\bar{E}+I, \\ M_{12} &= -\rho(Q+\rho\bar{E}^\top\bar{E})^{-1}\bar{E}^\top\Pi_{\mathcal{R}(\bar{F})}\bar{E}. \end{aligned} \tag{11}$$

The convergence properties of the ADMM iterations are characterized by the spectral properties of the matrix M. In particular, let $\{\phi_i\}$ denote the ordered eigenvalues of M so that $|\phi_1| \leq \cdots \leq |\phi_{2n-s}| < |\phi_{2n-s+1}| = \cdots = |\phi_{2n}|$ for $s \geq 1$. The ADMM iterations converge to the optimal solution if $\phi_{2n} = \cdots = \phi_{2n-s+1} = 1$ and the corresponding convergence factor (1) equals $\phi^\star = |\phi_{2n-s}|$. In the remainder of this paper, we address the following problem:

Problem 1: Given an optimization problem on the form (2), determine the jointly optimal scalar ρ and matrix R that minimize the convergence factor of the ADMM iterations.

As the initial step to tackle Problem 1, we first characterize the eigenvalues ϕ_i of M. Let $[u^\top\ v^\top]$ be an eigenvector of M associated with the eigenvalue ϕ, from which we conclude $\phi v = u$. Thus the following holds for the eigenvalues and corresponding eigenvectors of M

$$\phi^2 v = \phi M_{11} v + M_{12} v. \tag{12}$$

Our analysis will be simplified by picking R such that $\bar{E}^\top\bar{E} = \kappa Q$ for some $\kappa > 0$. The following lemma indicates that such an R can always be found.

Lemma 1: For $E \in \mathbf{R}^{p\times n}$ with full-column rank and $\kappa > 0$, there exists an R that does not change the constraint set in (2) and ensures that $\bar{E}^\top\bar{E} = \kappa Q$.

Proof: The proof may be found in [17]. ■

Now, replacing $\bar{E}^\top\bar{E} = \kappa Q$ in (11) we have

$$\begin{aligned} M_{11} &= \frac{\rho}{1+\rho\kappa}(\bar{E}^\top\bar{E})^{-1}\bar{E}^\top\left(\Pi_{\mathcal{R}(\bar{F})}-\Pi_{\mathcal{N}(\bar{F}^\top)}\right)\bar{E}+I, \\ M_{12} &= -\frac{\rho}{1+\rho\kappa}(\bar{E}^\top\bar{E})^{-1}\bar{E}^\top\Pi_{\mathcal{R}(\bar{F})}\bar{E}, \end{aligned}$$

The next result presents the explicit form of the eigenvalues of M in (10).

Theorem 1: Consider the ADMM iterations (10). If $\bar{E}^\top\bar{E} = \kappa Q$, the eigenvalues of M are described by

$$2\phi = \left(f(\rho)\bar{\lambda}+1\right) \pm \sqrt{\left(f(\rho)\bar{\lambda}+1\right)^2 - 2f(\rho)(\bar{\lambda}+1)}, \tag{13}$$

with

$$\begin{aligned} f(\rho) &= \frac{\rho\kappa}{1+\rho\kappa}, \\ \bar{\lambda} &= \frac{v^\top\left(\bar{E}^\top\left(\Pi_{\mathcal{R}(\bar{F})}-\Pi_{\mathcal{N}(\bar{F}^\top)}\right)\bar{E}\right)v}{v^\top(\bar{E}^\top\bar{E})v}, \\ \kappa &= v^\top(\bar{E}^\top\bar{E})v. \end{aligned}$$

Proof: The result follows from (12) and $\bar{E}^\top\bar{E} = \kappa Q$. ■

From (13) one directly sees how ρ and R affect the eigenvalues of M. Specifically, $f(\rho)$ is a function of ρ, while $\bar{\lambda}$ only depends on R. In the next section we address and solve Problem 1 for a particular class of problems. The analysis follows by applying Theorem 1 and studying the properties of (13) with respect to ρ and $\bar{\lambda}$.

IV. ADMM for Distributed Quadratic Programming

We are now ready to develop optimal scalings for the ADMM iterations for distributed quadratic programming. Specifically, we will consider a scenario where n agents collaborate to minimize an objective function on the form

$$\underset{\eta}{\text{minimize}} \quad \tfrac{1}{2}\eta^\top\bar{Q}\eta + \bar{q}^\top\eta, \tag{14}$$

where $\eta = [\eta_1^\top \ldots \eta_n^\top\ \eta_s]^\top$ and $\eta_i \in \mathbf{R}^{n_i}$ represents the private decisions of the agent i, $\eta_s \in \mathbf{R}$ is a shared decision among all agents, and $\bar{Q}$ has the structure

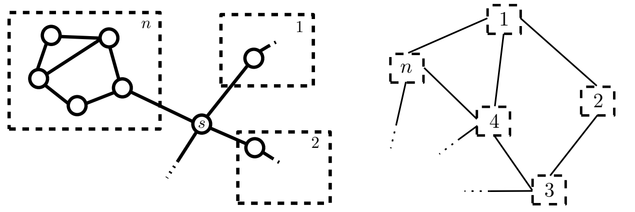
$$\bar{Q} = \begin{bmatrix} Q_{11} & 0 & \cdots & 0 & Q_{s1} \\ 0 & Q_{22} & & & Q_{s2} \\ \vdots & & \ddots & & \vdots \\ 0 & & & Q_{nn} & Q_{sn} \\ Q_{1s} & Q_{2s} & \cdots & Q_{ns} & Q_{ss} \end{bmatrix} \tag{15}$$

$$\bar{q}^\top = \begin{bmatrix} q_1^\top & \ldots & q_s^\top \end{bmatrix} \tag{16}$$

Here, $Q_{ss} \in \mathbf{R}$ for simplicity, $Q_{ii} \succ 0$, and $Q_{si} = Q_{is}^\top \in \mathbf{R}^{n_i}$. Such structured cost functions are common in optimization for interconnected systems. For instance, the problem of state estimation in electric power networks [13] gives rise to such sparsity structure. Given that η_s is scalar, state estimation for an electric power network with the physical structure depicted in Fig. 1(a) results in such structured $\bar{Q}$.

The optimization problem is almost decoupled, except for the shared variable η_s, and can be solved in a distributed fashion by introducing copies of the shared variable $x_{(i,s)} = \eta_s$ to each agent and solving the optimization problem

$$\begin{aligned} \underset{\{\eta_i\},\{x_{(i,s)}\}}{\text{minimize}} \quad & \textstyle\sum_{i=1}^n f_i(\eta_i, x_{(i,s)}) \\ \text{subject to} \quad & x_{(i,s)} = x_{(j,s)}, \quad \forall i,j,\ i \neq j \end{aligned}$$



(a) Coupling graph. (b) Communication graph.

Fig. 1. The cost coupling resulting in $\bar{Q}$ as outlined in (15). In (a) each agent $i \neq s$ represents a large area of the power network, while node s corresponds to the connection point between all the areas. In (b) the agents from different areas need to jointly minimize (14) constrained by the communication network.

with

$$f_i(\eta_i, x_{(i,s)}) \triangleq \frac{1}{2}\begin{bmatrix}\eta_i\\ x_{(i,s)}\end{bmatrix}^\top \begin{bmatrix}Q_{ii} & Q_{is}\\ Q_{si} & \alpha_i Q_{ss}\end{bmatrix}\begin{bmatrix}\eta_i\\ x_{(i,s)}\end{bmatrix} + \begin{bmatrix}q_i\\ \alpha_i q_s\end{bmatrix}^\top \begin{bmatrix}\eta_i\\ x_{(i,s)}\end{bmatrix},$$

where $\alpha_i > 0$ indicates how the cost associated with η_s is distributed among the the copies $x_{(i,s)}$ with $\sum_{i=1}^n \alpha_i = 1$.

Since the private variables η_i are unconstrained, one can solve for them analytically with respect to the shared variables, yielding

$$\begin{aligned} f_i(x_{(i,s)}) &\triangleq \frac{1}{2}x_{(i,s)}^\top \hat{Q}_i x_{(i,s)} + \hat{q}_i^\top x_{(i,s)},\\ \hat{Q}_i &= \left(Q_{ss}\alpha_i - Q_{is}Q_{ii}^{-1}Q_{si}\right),\\ \hat{q}_i &= (q_s\alpha_i - Q_{si}Q_{ii}^{-1}q_i). \end{aligned}$$

When $\bar{Q}$ is positive definite, there exist a set $\{\alpha_i\}$ such that each $f_i(x_{(i,s)})$ is convex, as stated in the following result.

Lemma 2: For $\bar{Q} \succ 0$, there exist $\{\alpha_i\}$ such that $\sum_{i=1}^n \alpha_i = 1$ and $\hat{Q}_i > 0$ for all $i = 1, \dots, n$.

Proof: The proof is given in [17]. ■

Hence the optimization problem can be rewritten as

$$\begin{array}{ll} \underset{\{x_{(i,s)}\}}{\text{minimize}} & \sum_{i=1}^n f_i(x_{(i,s)}) \\ \text{subject to} & x_{(i,s)} = x_{(j,s)}, \quad \forall\, i, j,\ i \neq j \end{array} \tag{17}$$

which reduces to an agreement, or consensus, problem on the shared variable x_s between all the nodes $i \neq s$ depicted in Fig. 1(b). Each agent i holds a local copy of the shared variable $x_i \triangleq x_{(i,s)}$ and it only coordinates with its neighbors $\mathcal{N}_i$ to compute the network-wide optimal solution to the agreement problem (17). The constraints imposed by the graph can be formulated in different ways, for instance by assigning auxiliary variables to each edge or node [2]. The former is illustrated next.

A. Enforcing agreement with edge variables

Constraints must be imposed on the distributed problem so that consensus is achieved. One such way is to enforce all pairs of nodes connected by an edge to have the same value, i.e. $x_i = x_j$ for all $(i, j) \in \mathcal{E}$. To include this constraint in the ADMM formulation, the auxiliary variable $z_{(i,j)}$ is created for each edge (i, j), with $z_{(i,j)} = z_{(j,i)}$, and the problem is formulated as

$$\begin{array}{ll} \underset{\{x_i\},\{z_{(i,j)}\}}{\text{minimize}} & \frac{1}{2}\sum_{i\in\mathcal{V}} f_i(x_i) \\ \text{subject to} & x_i = z_{(i,j)}, \quad \forall i \in \mathcal{V},\ \forall (i,j) \in \mathcal{E} \\ & z_{(i,j)} = z_{(j,i)}, \quad \forall (i,j) \in \mathcal{E}. \end{array}$$

Consider an arbitrary direction for each edge $e_i \in \mathcal{E}$. Now decompose the incidence matrix as $B = B_I + B_O$, where $[B_I]_{ij} = 1$ ($[B_O]_{ij} = 1$) if, and only if, node j is the head (tail) of the edge $e_i = (j, k)$. The optimization problem can be rewritten as

$$\begin{array}{ll} \underset{x,z}{\text{minimize}} & \frac{1}{2}x^\top Q x + q^\top x \\ \text{subject to} & \begin{bmatrix}RB_O\\ RB_I\end{bmatrix} x - \begin{bmatrix}R\\ R\end{bmatrix} z = 0, \end{array} \tag{18}$$

where $Q = \mathrm{diag}([\hat{Q}_1 \dots \hat{Q}_n])$, $q^\top = [\hat{q}_1 \dots \hat{q}_n]$, and $W = R^\top R$ is the non-negative diagonal matrix corresponding to the edge-weights.

Assumption 1: The graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$ is connected.

As derived in the previous section, the ADMM iterations can be written in matrix form as (10). Since $\Pi_{\mathcal{R}(\bar{F})} = 2W$, $\bar{E}^\top \Pi_{\mathcal{R}(\bar{F})} \bar{E} = \frac{1}{2}(B_I + B_O)^\top W (B_I + B_O) = \frac{1}{2}(D + A)$, and $\bar{E}^\top \bar{E} = B_O^\top W B_O + B_I^\top W B_I = D$, we have

$$\begin{aligned} M_{11} &= \rho(Q + \rho D)^{-1}A + I,\\ M_{12} &= -\tfrac{\rho}{2}(Q + \rho D)^{-1}(D + A). \end{aligned} \tag{19}$$

The main result in this paper is stated below and, for given $W = R^\top R$, it explicitly characterizes the optimal ρ solving Problem 1 and the corresponding convergence factor of (10) with M_{11} and M_{12} derived in (19).

Theorem 2: Suppose $W \succeq 0$ is chosen so that $\mathcal{G}$ is connected and $D = \kappa Q$ for $\kappa > 0$. Let $\{\lambda_i\}$ be the set of ordered generalized eigenvalues of $(A,\ D)$ for which $\lambda_1 \leq \dots \leq \lambda_n = 1$. The optimal step-size $\rho^\star$ that minimizes the convergence factor $\phi^\star$ is

$$\rho^\star = \begin{cases} \frac{1}{\kappa\sqrt{1-\lambda_{n-1}^2}} & ,\ \lambda_{n-1} \geq 0 \\ \frac{1}{\kappa} & ,\ \lambda_{n-1} < 0. \end{cases}$$

Furthermore, the corresponding convergence factor is

$$\phi^\star = |\phi_{2n-1}| = \begin{cases} \frac{1}{2}\left(1 + \frac{\lambda_{n-1}}{1+\sqrt{1-\lambda_{n-1}^2}}\right) & ,\ \lambda_{n-1} \geq 0 \\ \frac{1}{2} & ,\ \lambda_{n-1} < 0. \end{cases}$$

Proof: The proof may be found in [17]. ■

Note that, for a given W, the optimal $\rho^\star$ and convergence factor $|\phi_{2n-1}|$ are parameterized by κ and λ_{n-1}. Moreover, it is easy to see that $|\phi_{2n-1}| \geq \frac{1}{2}$ and minimizing λ_{n-1} leads to the minimum convergence factor. Hence, by finding $W^\star$ as the edge-weights minimizing λ_{n-1}, the optimal scaling is then given by $\rho^\star(\lambda_{n-1}^\star)W^\star$. The optimal choice of $W^\star$ is described in the following section.

B. Optimal network-constrained scaling

Here we address the second part of Problem 1 by computing the optimal scaling matrix $R^\star$ that, together with $\rho^\star$, provides the optimal scaling minimizing the ADMM convergence factor. But first we introduce a transformation to relax the assumption that $D = \kappa Q$. The constraints in the agreement problem (18) enforce $x = \mathbf{1}_n y$ for some $y \in \mathbf{R}$, where $\mathbf{1}_n \in \mathbf{R}^n$ is a vector with all entries equal to 1. Therefore the optimization problem is equivalent to

$$\underset{y}{\text{minimize}} \quad \tfrac{1}{2} y \mathbf{1}_n^\top Q \mathbf{1}_n y + q^\top \mathbf{1}_n y. \tag{20}$$

The next result readily follows.

Lemma 3: Consider the optimization problem (18). For given diagonal $D \succ 0$, the optimal solution to (18) remains unchanged when Q is replaced by $\frac{1}{\kappa} D$ if $\kappa = \frac{\mathbf{1}_n^\top D \mathbf{1}_n}{\mathbf{1}_n^\top Q \mathbf{1}_n}$.

Proof: The proof follows directly from converting (18) to (20) and having $\mathbf{1}_n^\top Q \mathbf{1}_n = \frac{1}{\kappa} \mathbf{1}_n^\top D \mathbf{1}_n$. ■

Thus, the constraint $D = \kappa Q$ can be achieved for any $D \succ 0$ by modifying the original problem (18), replacing Q with $\frac{1}{\kappa} D$ and letting $\kappa = \frac{\mathbf{1}_n^\top D \mathbf{1}_n}{\mathbf{1}_n^\top Q \mathbf{1}_n}$. Below we show how the minimization of λ_{n-1} with respect to the edge-weight matrix W can be formulated as a quasi-convex optimization problem.

Theorem 3: Consider the weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ and assume there exist non-negative edge-weights $\mathcal{W} = \{w_{e_k}\}$ such that $\mathcal{G}$ is connected. The non-negative edge-weights $\{w_{e_k}\}$ minimizing the second largest generalized eigenvalue of (A, D), λ_{n-1}, while having $\mathcal{G}$ connected are obtained from the optimal solution to the quasi-convex problem

$$\begin{array}{ll} \underset{\{w_{e_k}\},\lambda}{\text{minimize}} & \lambda \\ \text{subject to} & w_{e_k} \geq 0, \qquad \forall\, e_k = (i,j) \in \mathcal{E}, \\ & A_{ij} = w_{e_k}, \quad\ \forall\, e_k = (i,j) \in \mathcal{E}, \\ & A_{ji} = w_{e_k}, \quad\ \forall\, e_k = (i,j) \in \mathcal{E}, \\ & A_{ij} = 0, \qquad\ \forall\, (i,j) \notin \mathcal{E}, \\ & D = \operatorname{diag}(A \mathbf{1}_n), \\ & D \succ \epsilon I, \\ & A - D - \mathbf{1}_n \mathbf{1}_n^\top \prec 0, \\ & P^\top (A - \lambda D) P \prec 0, \end{array} \tag{21}$$

where the columns of $P \in \mathbf{R}^{n \times n-1}$ form an orthonormal basis of $\mathcal{N}(\mathbf{1}_n^\top)$ and $\epsilon > 0$. Denoting m as the number of edges, the edge-weight matrix is given by $W = \operatorname{diag}([w_{e_1} \ldots w_{e_m}])$.

Proof: The proof may be found in [17]. ■

Given the results derived in this section, the optimal scaling $\rho^\star W^\star$ solving Problem 1 can be computed as summarized in Algorithm 1.

Algorithm 1 Optimal Network-Constrained Scaling

1) Compute $W^\star$ and the corresponding $D^\star$ and $\lambda_{n-1}^\star$ according to Theorem 3;
2) Given $D^\star$ and Q, compute $\kappa^\star$ from Lemma 3;
3) Given $\kappa^\star$ and $\lambda_{n-1}^\star$, compute the optimal step-size $\rho^\star$ as described in Theorem 2;
4) The optimal scaling for the ADMM algorithm with Q replaced by $\frac{1}{\kappa^\star} D^\star$ is $\rho^\star W^\star$.

V. Numerical examples

Next we illustrate our results in numerical examples.

A. Distributed quadratic programming

Consider a distributed quadratic programming problem with $n = 3$ agents and an objective function defined by

$$\bar{Q} = \left[\begin{array}{cc|cc|cc|c} 4 & 1 & & & & & 1 \\ 1 & 6 & & & & & 2 \\ \hline & & 5 & 4 & & & 3 \\ & & 4 & 8 & & & 4 \\ \hline & & & & 8 & 7 & 5 \\ & & & & 7 & 9 & 6 \\ \hline 1 & 2 & 3 & 4 & 5 & 6 & 8 \end{array}\right]$$

$$\bar{q}^\top = \left[\begin{array}{cc|cc|cc|c} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right].$$

As shown previously, the optimization problem can be reformulated on the form

$$\begin{array}{ll} \underset{\{x_{(i,s)}\}}{\text{minimize}} & \sum_{i=1}^{n} \frac{1}{2} x_{(i,s)}^\top \hat{Q}_i x_{(i,s)} + \hat{q}_i^\top x_{(i,s)} \\ \text{subject to} & x_{(i,s)} = x_{(j,s)}, \quad \forall i \neq j \end{array}$$

with $n = 3$, $\alpha = \frac{1}{n}[0.5\, 0.9\, 1.6]$, $\hat{Q}_1 = 0.5507$, $\hat{Q}_2 = 0.0667$, $\hat{Q}_3 = 0.2232$, $\hat{q}_1 = -0.3116$, $\hat{q}_2 = -0.3667$, and $\hat{q}_3 = -0.1623$. As for the communication graph, we consider a line graph with node 2 connected to nodes 1 and 3. Algorithm 1 is applied, resulting in $\lambda_{n-1}^\star = 0$ with the edge weights $w_{e_1} = w_{e_2} = 0.1566$ and degree matrix $D = \operatorname{diag}([0.1566 \;\; 0.3132 \;\; 0.1566])$. From Theorem 2 we then have $\rho^\star = \frac{1}{\kappa} = \frac{\sum_{i=1}^{3} \hat{Q}_i}{\mathbf{1}_n^\top D \mathbf{1}_n}$ and $\phi^\star = |\phi_{2n-1}| = 0.5$, which is the best achievable convergence factor. The performance of the ADMM algorithm with optimal network-constrained scaling is presented in Fig. 2. The performance of the unscaled ADMM algorithm with unitary edge weights and manually optimized step-size ρ is also depicted for comparison. The convergence factor of the manually tuned ADMM algorithm is $|\phi_{2n-1}| = 0.557$, thus exhibiting worse performance than the optimally scaled algorithm as depicted in Fig. 2.

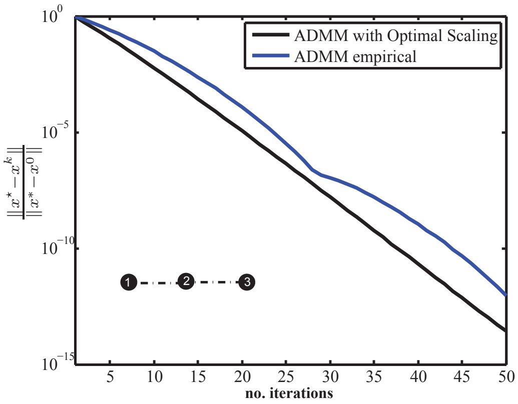
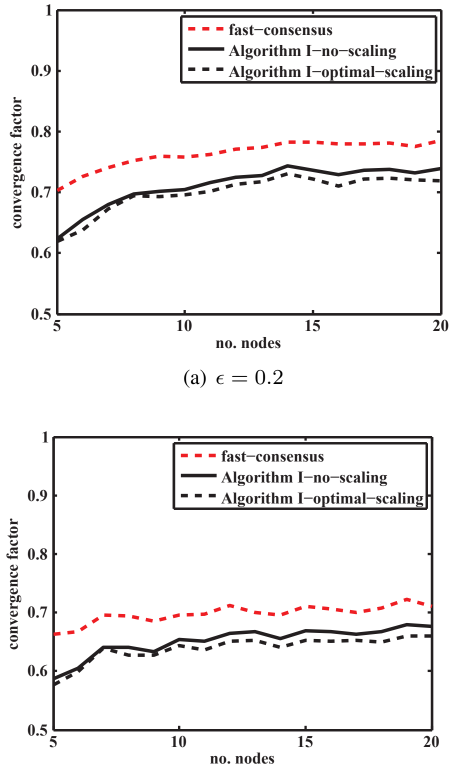


Fig. 2. Normalized error for the scaled ADMM algorithm with $W^\star$, and $\rho^\star$ obtained from Algorithm 1 and the unscaled ADMM algorithm with unitary edge weights and manually selected best step-size $\rho = 0.55$ via exhaustive search.



(a) $\epsilon = 0.2$

(b) $\epsilon = 0.8$

Fig. 3. Performance comparison of the proposed optimal scaling for the ADMM algorithm with state-of-the-art fast-consensus [2]. The network of size $n = [5, 20]$ is randomly generated by Erdős-Rényi graphs with low and high densities $\epsilon = \{0.2, 0.8\}$.

B. Distributed consensus

In this section we apply our methodology to derive optimally scaled ADMM iterations for the average consensus problems and compare our convergence factors with the state-of-the art fast consensus algorithm presented in [2]. The average consensus problem is a particular case of (18) where $x \in \mathbf{R}$, $Q = \alpha I$ for some $\alpha \in \mathbf{R}$, and $q = \mathbf{0}$. As an indicator of the performance, we compute the convergence factors for the two methods on a large number of randomly generated Erdős-Rényi graphs. Fig. 3 presents Monte Carlo simulations of the convergence factors versus the number of nodes $n \in [5, 20]$. Each component (i, j) in the adjacency matrix A is non-zero with probability $p = (1 + \epsilon)\frac{\log(n)}{n}$, where $\epsilon \in (0, 1)$ and n is the number of vertices. In our simulations, we consider two scenarios: sparse graphs with $\epsilon = 0.2$ and dense topologies $\epsilon = 0.8$. For every network size, 50 network instances are generated, the convergence factors are computed and averaged to generate the depicted results. The figure shows two versions of Algorithm 1 with and without weight optimization in Theorem 3. We observe a significant improvement compared to the state-of-the-art fast consensus [2] in both sparse and dense topologies.

VI. Conclusions and future work

Optimal scaling of the ADMM method for distributed quadratic programming was addressed. In particular, a class of distributed quadratic problems were cast as equality-constrained quadratic problems, to which the scaled ADMM method is applied. For this class of problems, the network-constrained scaling corresponds to the usual step-size constant and the edge weights of the communication graph. Under mild assumptions on the communication graph, analytical expressions for the optimal convergence factor and the optimal step-size were derived in terms of the spectral properties of the graph. Supposing the optimal step-size is chosen, the convergence factor is further minimized by optimally choosing the edge weights. Our results were illustrated in numerical examples and significant performance improvements over state-of-the art techniques were demonstrated.

References

[1] A. Nedic, A. Ozdaglar, and P. Parrilo, “Constrained consensus and optimization in multi-agent networks,” *Automatic Control, IEEE Transactions on*, vol. 55, no. 4, pp. 922–938, Apr. 2010.

[2] T. Erseghe, D. Zennaro, E. Dall’Anese, and L. Vangelista, “Fast consensus by the alternating direction multipliers method,” *Signal Processing, IEEE Transactions on*, vol. 59, pp. 5523 –5537, 2011.

[3] P. Giselsson, M. D. Doan, T. Keviczky, B. D. Schutter, and A. Rantzer, “Accelerated gradient methods and dual decomposition in distributed model predictive control,” *Automatica*, vol. 49, no. 3, pp. 829 – 833, 2013.

[4] F. Farokhi, I. Shames, and K. H. Johansson, “Distributed MPC via dual decomposition and alternative direction method of multipliers,” *Distributed MPC Made Easy, Springer*, 2013.

[5] D. Falcao, F. Wu, and L. Murphy, “Parallel and distributed state estimation,” *Power Systems, IEEE Transactions on*, vol. 10, no. 2, pp. 724–730, May 1995.

[6] 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*, vol. 3 Issue: 1, pp. 1–122, 2011.

[7] C. Conte, T. Summers, M. Zeilinger, M. Morari, and C. Jones, “Computational aspects of distributed optimization in model predictive control,” in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, 2012.

[8] M. Annergren, A. Hansson, and B. Wahlberg, “An ADMM algorithm for solving ℓ_1 regularized MPC,” 2012.

[9] J. Mota, J. Xavier, P. Aguiar, and M. Puschel, “Distributed admm for model predictive control and congestion control,” in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, 2012.

[10] Z. Luo, “On the linear convergence of the alternating direction method of multipliers,” *ArXiv e-prints*, 2012.

[11] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, “Optimal parameter selection for the alternating direction method of multipliers (admm): quadratic problems,” *ArXiv e-prints*, 2013.

[12] W. Deng and W. Yin, “On the global and linear convergence of the generalized alternating direction method of multipliers,” Rice university Technical Report, Tech. Rep., 2012.

[13] A. Gómez-Expósito, A. de la Villa Jaén, C. Gómez-Quiles, P. Rousseaux, and T. V. Cutsem, “A taxonomy of multi-area state estimation methods,” *Electric Power Systems Research*, vol. 81, no. 4, pp. 1060–1069, 2011.

[14] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation:Numerical methods*. New York: Athena Scientific, 1997.

[15] C. Godsil and G. Royle, *Algebraic Graph Theory*, ser. Graduate Texts in Mathematics. Springer, 2001, vol. 207.

[16] A. Teixeira, J. Araujo, H. Sandberg, and K. H. Johansson, “Distributed actuator reconfiguration in networked control systems,” in *Necsys: IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 2013.

[17] A. Teixeira, E. Ghadimi, I. Shames, H. Sandberg, and M. Johansson, “Optimal scaling of the ADMM algorithm for distributed quadratic programming,” Tech. Rep., 2013. [Online]. Available: http://arxiv.org/abs/1303.6680