Exact Spectral-Like Gradient Method for Distributed Optimization

Dušan Jakovetić [*] Nataša Krejić [*]
Nataša Krklec Jerinkić [*]

Abstract

Since the initial proposal in the late 80s, spectral gradient methods continue to receive significant attention, especially due to their excellent numerical performance on various large scale applications. However, to date, they have not been sufficiently explored in the context of distributed optimization. In this paper, we consider unconstrained distributed optimization problems where n nodes constitute an arbitrary connected network and collaboratively minimize the sum of their local convex cost functions. In this setting, building from existing exact distributed gradient methods, we propose a novel exact distributed gradient method wherein nodes' step-sizes are designed according to the novel rules akin to those in spectral gradient methods. We refer to the proposed method as Distributed Spectral Gradient method (DSG). The method exhibits R-linear convergence under standard assumptions for the nodes' local costs and safeguarding on the algorithm step-sizes. We illustrate the method's performance through simulation examples.

Keywords: Distributed optimization, spectral gradient, R-linear convergence.

AMS subject classification. 90C25, 90C53, 65K05

[*] Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia. e-mail: {djakovet@uns.ac.rs, natasak@uns.ac.rs, natasa.krklec@dmi.uns.ac.rs}. Research supported by the Serbian Ministry of Education, Science, and Technological Development, Grant no. 174030. This work is also supported by the I-BiDaaS project, funded by the European Commission under Grant Agreement No. 780787. This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which maybe made of the information contained therein.

1 Introduction

We consider a connected network with n nodes, each of which has access to a local cost function $f_i : \mathbb{R}^d \to \mathbb{R},\ i = 1, \ldots, n$. The objective for all nodes is to minimize the aggregate cost function $f : \mathbb{R}^d \to \mathbb{R}$, defined by

$$f(y) = \sum_{i=1}^{n} f_i(y). \tag{1}$$

Problems of this form attract a lot of scientific interest as they arise in many emerging applications like distributed inference in sensor networks [29, 16, 18, 8], distributed control, [22], distributed learning, e.g., [7], etc.

For example, with distributed supervised learning, a training data set is partitioned into n blocks which correspond to distinct nodes in the network (e.g., servers, nodes in a computer cluster, etc.) The goal is then to train a machine learning model based on the data by all nodes without transferring data to a single location, due to, e.g., storage limitations or privacy concerns. In this context, function $f_i(\cdot)$ is the empirical loss with respect to the data available at node i:

$$f_i(x) = \sum_{j \in \mathcal{J}_i} \ell_i\left(x,\, \mathcal{D}_{i,j}\right) + \mathcal{R}_i(x),$$

where $\mathcal{D}_{i,j}$ is a data sample at node i, $j \in \mathcal{J}_i$, $\mathcal{J}_i$ is the indices set of node i's data samples, $\ell_i(\cdot, \cdot)$ is the loss function at node i (e.g., logistic, quadratic, hinge, etc.), and $\mathcal{R}_i(\cdot)$ is the regularization function at node i (e.g., the quadratic regularization). More concretely, with L2-regularized logistic losses, we have:

$$\ell_i\left(x,\, \mathcal{D}_{i,j}\right) = \ln\left(1 + \exp(-b_i(a_i^\top x))\right), \qquad \mathcal{R}_i(x) = \frac{c}{2}\|x\|^2.$$

Here, $\|\cdot\|$ stands for the 2-norm, $\mathcal{D}_{i,j} = (a_{i,j}, b_{i,j})$, where $a_{i,j} \in \mathbb{R}^d$ is a feature vector, $b_{i,j} \in \{-1, +1\}$ is the corresponding class label, and $c > 0$ is the regularization tuning parameter; see, e.g., [7].

To solve this and related problems several distributed first order methods, e.g., [25, 8, 13], and second order methods, e.g., [19, 20, 15], have been proposed. The methods of this type converge to an approximate solution of problem (1) if a constant (non-diminishing) step size is used; they can be interpreted through a penalty-like reformulation of (1); see [14, 19] for details.

Convergence to an exact solution can be achieved by using diminishing step-sizes, but this comes at a price of slower convergence.

More recently, *exact* distributed first order methods, e.g., [31, 12, 30, 26, 11], and second order methods, e.g., [21, 20], have been proposed, that converge to the exact solution under *constant* step sizes. The method in [30] uses two different weight matrices, differently from the standard distributed gradient method that utilizes a single weight matrix. The methods in [26, 23, 24] implement tracking of the network-wide average gradient and correct the dynamics of the standard distributed method [25] by replacing the nodes' local gradients with the tracked global average gradient estimates. A unification of a class of exact first order methods and some further improvements are presented in [11]. References [31, 23] study exact methods with uncoordinated step-sizes, while reference [12] proposes exact methods for non-convex problems. An exact distributed second order method has been developed in [21]. We refer to [11] for a detailed review of other works on exact distributed methods.

Spectral gradient methods are a popular class of methods in centralized optimization due to their simplicity and efficiency. The class originated with the proposal of the Barzilei-Borwein method [1] and its analysis therein for two-dimensional convex quadratic functions, while the method has been subsequently extended to more general optimization problems, both unconstrained and constrained, [27, 28, 6]. Spectral gradient methods can be viewed as a mean to incorporate second-order information in a computationally efficient manner into gradient descent methods. In practice, they achieve significantly faster convergence with respect to standard gradient methods while the additional computational overhead per iteration is very small. Roughly speaking, the main idea of spectral gradient methods is to approximate the Hessian at each iteration with a scalar matrix (the leading scalar of the matrix is called the spectral coefficient) that approximately fits the secant equation. Calculating the spectral gradient's scalar matrix is much cheaper than evaluation of the Newton direction while the convergence speed is usually much better than that of the gradient method. Spectral methods are characterized by a non-monotone behaviour which makes them suitable for combination with non-monotone line search methods, [28]. It was demonstrated in [28] that the spectral gradient method can be more efficient than the conjugate gradient method for certain classes of optimization problems. The R-linear convergence rate was established in [9], while extensions to constrained optimization in the form of Spectral Projected Gradient (SPG) methods are developed in

[3, 4, 5]. A vast number of applications is available in the literature, and a comprehensive overview is presented in [6].

The principal aim of this paper is to provide a generalization of spectral gradient methods to distributed optimization and give preliminary numerical tests of its efficiency. Extension of spectral gradient methods to a distributed setting is a highly nontrivial task. We develop an exact method (converging to the exact solution) that we refer to as Distributed Spectral Gradient method (DSG). The method utilizes step-sizes that are akin to those of centralized spectral methods. The spectral-like step-sizes are embedded into the exact distributed first order method in [26]; see also [23, 24]. We utilize the primal-dual interpretation of the method in [26] – as provided in [11] (see also [24]) – and the corresponding form of the error recursion equation. An analogy with the error recursion of the conventional spectral method stated in [27] is exploited to define the time-varying, node dependent, algorithm step-sizes. This analogy also allows for an intuitive interpretation of the proposed method.

We show that the proposed DSG method exhibits R-linear convergence rate under appropriate assumptions on the nodes' local objectives, static, undirected networks, and appropriate safeguarding of the step-sizes.

The proposed DSG method has several favorable features. Namely, simulations suggest that DSG converges under a significantly wider range of admissible step-sizes than existing exact first order methods like [26, 23]. Indeed, existing methods require for convergence that step-sizes be sufficiently small, both in theory and in practical implementations. We show by simulation examples that DS converges for step-size ranges which are orders of magnitude broader than the admissible step-size ranges of [23]. We further show analytically on a consensus problem-special case, under a special structure of the underlying weight matrix W, that DSG converges without any a priory upper bound on the step-sizes and with a lower bound on the step-sizes, while the method in [26] diverges on the same example for the step-size larger than two.

Another important feature of the DSG method is that it adaptively adjusts the step-sizes over iterations such that good convergence speed is achieved. This eliminates the need to hand-optimize and/or align beforehand the step-size values across nodes, as it is the case with existing methods like [26]. This beforehand tuning may be expensive, resource-consuming, and tedious process, in many scenarios. In contrast, the proposed method requires only a coarse estimate (to within a factor of 10-100, for example) of the nodes' gradients' Lipschitz constant beforehand. We compare the performance of DSG by simulation with [26] under a hand-optimized step-size. While the latter method with hand-optimized step-size may converge faster than DSG, it may also converge worse than DSG, when the step-size of [26] is chosen poorly. Therefore, when aligning and hand-tuning of step-sizes is not feasible beforehand, the proposed method represents a valuable choice.

The paper is organized as follows. Some preliminary considerations and assumptions are presented in Section 2. The proposed distributed spectral method (DSG) is introduced in Section 3, while the convergence theory is developed in Section 4. Initial numerical tests are presented in Section 5, and some conclusions are drawn in Section 6. Some auxiliary proofs are relegated to the Appendix.

2 Model and preliminaries

The network and optimization models that we assume are described in Subsection 2.1. The proposed method is based on the distributed gradient method developed in [26] and the centralized spectral gradient method [27] which are briefly reviewed in Subsection 2.2 and 2.3. The convergence analysis is based on the Small Gain Theorem which is stated in Subsection 2.4.

2.1 Optimization and network models

We impose a set of standard assumptions on the functions f_i in (1) and on the underlying network.

Assumption A1. Assume that each local function $f_i : \mathbb{R}^d \to \mathbb{R},\ i = 1, \ldots, n$ is twice continuously differentiable and for all $i = 1, \ldots, n$ and all $y \in \mathbb{R}^p$, there holds

$$\mu_i I \preceq \nabla^2 f_i(y) \preceq l_i I \tag{2}$$

where $l_i \geq \mu_i \geq 0$ and $\mu := \sum_{i=1}^n \mu_i > 0$.

Here, notation $\Gamma \preceq \Upsilon$ means that matrix $(\Upsilon - \Gamma)$ is positive semi-definite. This implies that the gradients of the f_i's are Lipschitz continuous with constants l_i and that the full gradient ∇f is Lipschitz continuous with constant

$$L := \sum_{i=1}^n l_i. \tag{3}$$

Moreover, under the Assumption A1, the objective function f is μ-strongly convex and problem (1) is solvable and has a unique solution, denoted by y^*. For future reference, let us introduce the function $F : \mathbb{R}^{nd} \to \mathbb{R}$, defined by:

$$F(x) = \sum_{i=1}^{n} f_i(x_i), \tag{4}$$

where $x \in \mathbb{R}^{nd}$ consists of n blocks $x_i \in \mathbb{R}^d$, i.e., $x = ((x_1)^T, ..., (x_n)^T)^T$. Assumption A1 clearly implies that ∇F is Lipschitz continuous, where a Lipschitz constant can be taken as $\max_{i=1,\ldots,n} l_i$. For the sake of simplicity, we retain the same Lipschitz constant as for ∇f, i.e., for any $y, z \in \mathbb{R}^{nd}$, there holds:

$$\|\nabla F(y) - \nabla F(z)\| \leq L\|y - z\|, \tag{5}$$

where L is defined by (3).

We assume that the network of nodes is an undirected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges, i.e., all pairs $\{i, j\}$ of nodes which can exchange information through a communication link. **Assumption A2.** The network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is connected, undirected and simple (no self-loops nor multiple links).

Let us denote by O_i the set of nodes that are connected with node i through a direct link (neighborhood set), and let $\bar{O}_i = O_i \bigcup \{i\}$. Associate with $\mathcal{G}$ a symmetric, doubly stochastic $n \times n$ matrix W. The elements of W are all nonnegative and both rows and columns sum up to one. More precisely, the following is assumed.

Assumption A3. The matrix $W = W^T \in \mathbb{R}^{n \times n}$ is doubly stochastic, with elements w_{ij} such that

$$w_{ij} > 0 \text{ if } \{i, j\} \in \mathcal{E}, \; w_{ij} = 0 \text{ if } \{i, j\} \notin \mathcal{E}, \; i \neq j, \; \text{ and } w_{ii} = 1 - \sum_{j \in O_i} w_{ij}$$

and there exist constants w_{min} and w_{max} such that for $i = 1, \ldots, n$

$$0 < w_{min} \leq w_{ii} \leq w_{max} < 1.$$

Denote by $\lambda_1 \geq \ldots \geq \lambda_n$ the eigenvalues of W. It can be shown that $\lambda_1 = 1$, and $|\lambda_i| < 1$, $i = 2, ..., n$.

For future reference, define the $n \times n$ matrix J that has all the entries equal $1/n$. We refer to J as the ideal consensus matrix; see, e.g., [17]. Also, introduce the $(nd)\times(nd)$ matrix $\mathcal{W}=W\otimes I$, where $\otimes$ denotes the Kronecker product and I is the identity matrix from $\mathbb{R}^{d\times d}$. It can be seen that $d\times d$ block on the (i,j)-th position of the matrix $\mathcal{W}$ equals to $w_{ij}\,I$. By properties of the Kronecker product, the eigenvalues of $\mathcal{W}$ are $\lambda_1,...,\lambda_n$, each one occurring with the multiplicity d. We also introduce the $(nd)\times(nd)$ matrix $\mathcal{J}=J\otimes I$, where, as before, J is the $n\times n$ ideal consensus matrix, and I is the $d\times d$ identity matrix. Also, we denote by $\mathcal{I}$ the $(nd)\times(nd)$ identity matrix.

2.2 Exact Distributed first order method

Let us now briefly review the distributed first order method in [26]; see also [23, 24]. These methods serve as a basis for the development of the proposed distributed spectral gradient method. The method in [26] maintains over iterations $k=0,1,...$, at each node i, the solution estimate $x_i^k\in\mathbb{R}^d$ and an auxiliary variable $z_i^k\in\mathbb{R}^d$. Specifically, the update rule is as follows

$$x_i^{k+1} = \sum_{j\in\bar{O}_i} w_{ij}\,x_j^{(k)}-\alpha\,z_i^k \tag{6}$$

$$z_i^{k+1} = \sum_{j\in\bar{O}_i} w_{ij}\,z_j^{(k)}+\left(\nabla f_i(x_i^{k+1})-\nabla f_i(x_i^k)\right),\ k=0,1,... \tag{7}$$

Here, $\alpha>0$ is a constant step-size; the initialization x_i^0, $i=1,...,n$, is arbitrary, while $z_i^0=\nabla f_i(x_i^0)$, $i=1,...,n$. Equation (6) shows that each node i, as with standard distributed gradient method [25], makes two-fold progress: 1) by weight-averaging its solution estimate with its' neighbors; and 2) by taking a step opposite to the estimated gradient direction. The standard distributed gradient method in [25] takes a negative step in the direction of $\nabla f_i(x_i^k)$, while the method in [26] makes a step in direction of z_i^k. This vector serves as a tracker of the network-wide gradient $\sum_{i=1}^n\nabla f_i(x_i^k)$. This modification in the update rule enables convergence to the exact solution under a constant step-size [26].

It is useful to represent method (6)–(7) in vector format. Let $x^k\in\mathbb{R}^{nd}$, $z^k\in\mathbb{R}^{nd}$, and recall function F in (4) and matrix $\mathcal{W}=W\otimes I$. Then, the method (6)–(7) in the vector form becomes

$$x^{k+1} = \mathcal{W}\,x^{(k)}-\alpha\,z^k \tag{8}$$

$$z^{k+1} = \mathcal{W}\,z^{(k)}+\left(\nabla F(x^{k+1})-\nabla F(x^k)\right),\ k=0,1,..., \tag{9}$$

with arbitrary x^0 and $z^0=\nabla F(x^0)$.

The method (8)–(9) allows for a primal-dual interpretation; see [11] and also [24] for a similar interpretation. The primal-dual interpretation will be important for the development of the proposed distributed spectral gradient method. Namely, it is demonstrated in [11] that (8)–(9) is equivalent to the following update rule

$$\begin{aligned} x^{k+1} &= \mathcal{W}x^k - \alpha(\nabla F(x^k) + u^k) \qquad (10)\\ u^{k+1} &= \mathcal{W}u^k + (\mathcal{W} - \mathcal{I})\nabla F(x^k), \qquad (11)\end{aligned}$$

with variable $u^0 = 0 \in \mathbb{R}^{dn}$ and arbitrary x^0. It can be shown that, under appropriately chosen step-size α, the sequence $\{x^k\}$ converges to $x^* := \mathbf{1} \otimes y^* = (\,(y^*)^T, ..., (y^*)^T\,)^T$, and u^k converges to $-\nabla F(\mathbf{1} \otimes y^*) = -(\nabla f_1(y^*)^T, ..., \nabla f_n(y^*)^T)^T$. Here, $\mathbf{1} \in \mathbb{R}^n$ is the vector with all components equal to one.

2.3 Centralized spectral gradient method

Let us briefly review the spectral gradient (SG) method in centralized optimization. Consider the unconstrained minimization problem with a generic objective function $\phi : \mathbb{R}^d \to \mathbb{R}$ which is continuously differentiable. Let the initial solution estimate be arbitrary $x^0 \in \mathbb{R}^d$. The SG method generates the sequence of iterates $\{x^k\}$ as follows

$$x^{k+1} = x^k - \sigma_k^{-1}\, \nabla \phi(x^k),\ k = 0, 1, \ldots, \qquad (12)$$

where the initial spectral coefficient $\sigma_0 > 0$ is arbitrary and σ_k, $k = 1, 2, ...$, is given by

$$\sigma_k = \mathcal{P}_{[\sigma_{\min}, \sigma_{\max}]}(\sigma_k'),\ \ \sigma_k' = \frac{(s^{k-1})^T y^{k-1}}{(s^{k-1})^T s^{k-1}}. \qquad (13)$$

Here, $0 < \sigma_{\min} < \sigma_{\max} < +\infty$ are given constants, $s^{k-1} = x^k - x^{k-1}$, $y^{k-1} = \nabla \phi(x^k) - \nabla \phi(x^{k-1})$, and $\mathcal{P}_{[a,b]}$ stands for the projection of a scalar onto the interval $[a, b]$. The projection onto the interval $[\sigma_{\min}, \sigma_{\max}]$ is the safeguarding that is necessary for convergence. The spectral coefficient σ_k' is derived as follows. Assume that the Hessian approximation in the form $B_k = \sigma_k I$. Then the approximate secant equation

$$B_k s^{k-1} \approx y^{k-1} \qquad (14)$$

can be solved in the least square sense. It is easy to show that least squares solution of (14) yields exactly (13). For future reference, we briefly review the result on the evolution of error for the SG method stated in [27]. Consider the special case of a strongly convex quadratic function $\phi(x) = \frac{1}{2}x^T Ax + b^T x$ for a symmetric positive definite matrix A, and denote by $e^k := x^* - x^k$ the error at iteration k, where $x^\star$ is the minimizer of ϕ. Then, it can be shown that the error evolution can be expressed as [27]:

$$e^{k+1} = (I - \sigma_k^{-1} A)e^k. \tag{15}$$

The above relation will play a key role in the intuitive explanation of the distributed spectral gradient method proposed in this paper.

2.4 Small gain theorem

Convergence analysis of the proposed method will be based upon the Small Gain Theorem, e.g. [10]. This technique has been previously used and proved successful for the analysis of exact distributed gradient methods in, e.g., [23, 24]. We briefly introduce the concept here, while more details are available in [10, 23].

Denote by $\mathbf{a} := a^1, a^2, \ldots$ an infinite sequence of vectors, $a^k \in \mathbb{R}^d,\ k = 0, 1, \ldots$. For a fixed $\delta \in (0, 1)$, define

$$\|\mathbf{a}\|^{\delta,K} = \max_{k=0,1,\ldots,K} \{\frac{1}{\delta^k}\|a^k\|\}$$

$$\|\mathbf{a}\|^{\delta} = \sup_{k\geq 0}\{\frac{1}{\delta^k}\|a^k\|\}.$$

Obviously, for any $K' \geq K \geq 0$ we have $\|\mathbf{a}\|^{\delta,K} \leq \|\mathbf{a}\|^{\delta,K'} \leq \|\mathbf{a}\|^{\delta}$. Also, if $\|\mathbf{a}\|^{\delta}$ is finite for some $\delta \in (0, 1)$ than the sequence $\mathbf{a}$ converges to zero R-linearly. We present the Small Gain Theorem in a simplified form that involves only two sequences, as this will suffice for our considerations; for more general forms of the result see [10, 23].

Theorem 2.1. *[10, 23]. Consider two infinite sequences $\mathbf{a} = a^0, a^1, \ldots,\ \mathbf{b} = b^0, b^1, \ldots,$ with $a^k, b^k \in \mathbb{R}^d,\ k = 0, 1, \ldots$. Suppose that for some $\delta \in (0, 1)$ and for all $K = 0, 1, \ldots,$ there holds*

$$\|\mathbf{a}\|^{\delta,K} \leq \gamma_1\|\mathbf{b}\|^{\delta,K} + w_1$$

$$\|\mathbf{b}\|^{\delta,K} \leq \gamma_2\|\mathbf{a}\|^{\delta,K} + w_2,$$

where $\gamma_1 \cdot \gamma_2 \in [0,1)$. *Then*

$$\|\mathbf{a}\|^{\delta} \leq \frac{1}{1-\gamma_1\gamma_2}(w_1\gamma_2 + w_2).$$

Furthermore, $\lim_{k\to\infty} a^k = 0$ *R-linearly.*

Following, for example, the proof of Lemma 6 in [11] (see also [23]), it is easy to derive the result below.

Lemma 2.1. *Consider three infinite sequence* $\mathbf{a} = a^0, a^1, \ldots,\ \mathbf{b} = b^0, b^1, \ldots,\ \mathbf{c} = c^0, c^1, \ldots$ *with* $a^k, b^k, c^k \in \mathbb{R}^d,\ k = 0,1,\ldots$. *Suppose that there holds*

$$\|a^{k+1}\| \leq c_1\|a^k\| + c_2\|b^k\| + c_3\|c^k\|,\ k = 0,1,\ldots$$

where $c_1, c_2, c_3 \geq 0$. *Then, for all* $K = 0,1,\ldots$ *and* $0 \leq c_1 < \delta < 1$,

$$\|\mathbf{a}\|^{\delta,K} \leq \frac{c_2}{\delta - c_1}\|\mathbf{b}\|^{\delta,K} + \frac{c_3}{\delta - c_1}\|\mathbf{c}\|^{\delta,K} + \frac{\delta}{\delta - c_1}\|a^0\|.$$

3 Spectral gradient method for distributed optimization

3.1 The algorithm

Let us now present the proposed Distributed Spectral Gradient, DSG, method. The method incorporates spectral-like step size policy into (8)–(9). The step-sizes are locally computed and vary both across nodes and across iterations. As (8)–(9), the DSG method maintains the sequence of solution estimates $x^k \in \mathbb{R}^{nd}$ and an auxiliary sequence $z^k \in \mathbb{R}^{nd}$. Specifically, the update rule is as follows

$$x^{k+1} = \mathcal{W}\,x^k - \Sigma_k^{-1}\,z^k \qquad (16)$$

$$z^{k+1} = \mathcal{W}\,z^k + \left(\nabla F(x^{k+1}) - \nabla F(x^k)\right),\ k = 0,1,... \qquad (17)$$

The initial solution estimate x^0 is arbitrary, while $z^0 = \nabla F(x^0)$. Here,

$$\Sigma_k = diag\left(\sigma_1^k\, I, \ldots, \sigma_n^k\, I\right),$$

is the $nd \times nd$ diagonal matrix that collects inverse step-sizes σ_i^k at all nodes $i = 1, ..., n$. The inverse step-sizes σ_i^k are given by:

$$\sigma_i^k = \mathcal{P}_{[\sigma_{\min}, \sigma_{\max}]} \left\{ \frac{(s_i^{k-1})^T y_i^{k-1}}{(s_i^{k-1})^T s_i^{k-1}} + \sigma_i^{k-1} \sum_{j \in \bar{O}_i} w_{ij} \left(1 - \frac{(s_j^{k-1})^T s_i^{k-1}}{(s_i^{k-1})^T s_i^{k-1}} \right) \right\} \quad (18)$$

$$\begin{aligned} s_i^{k-1} &= x_i^k - x_i^{k-1} \\ y_i^{k-1} &= \nabla f_i(x_i^k) - \nabla f_i(x_i^{k-1}), \end{aligned}$$

where $0 < \sigma_{\min} < \sigma_{\max} < +\infty$ are, as before, the safeguarding parameters.

Notice that the proposed step-size choice does not incur an additional communication overhead; each node i only needs to additionally store in its memory u_j^k for all its neighbors $j \in O_i$.

In view of (10)–(11), the method (16)–(17) can be equivalently represented as follows

$$\begin{aligned} x^{k+1} &= \mathcal{W} x^k - \Sigma_k^{-1} (\nabla F(x^k) + u^k) & (19) \\ u^{k+1} &= \mathcal{W} u^k + (\mathcal{W} - \mathcal{I}) \nabla F(x^k), \ k = 0, 1, ..., & (20) \end{aligned}$$

with variable $u^0 = 0 \in \mathbb{R}^{nd}$.

At the beginning of each iteration $k + 1$, a node i holds the current $x_i^k, \nabla f_i(x_i^k), u_i^k$, computes $s_i^k = x_i^k - x_i^{k-1}$, $y_i^{k-1} = \nabla f_i(x_i^k) - \nabla f_i(x_i^{k-1})$ and computes σ_i^k by (18). After that, it updates its' estimation of x_i through communication with all neighbouoring nodes $j \in O_i$ as

$$x_i^{k+1} = \sum_{j \in \bar{O}_i} w_{ij} x_j^k - (\sigma_i^k)^{-1} \left(\nabla f_i(x_i^k) + u_i^k \right)$$

$$u_i^k = \sum_{j \in \bar{O}_i} w_{ij} u_j^k + \sum_{j \in \bar{O}_i} w_{ij} \nabla f_j(x_j^k) - \nabla f_i(x_i^k).$$

Therefore, the iteration is fully distributed and each node interchanges messages only locally, with immediate neighbors.

We next comment on the safeguarding parameters in (18). In practice, the safeguarding upper bound $\sigma_{\max}$ can be set to a large number, e.g., $\sigma_{\max} = 10^8$; the safeguarding lower bound can be set to $\sigma_{\min} = \frac{L}{c}$, with $c \in [10, 100]$. This in particular means that the proposed algorithm (19)–(20) can take step-sizes $\frac{1}{\sigma_i^k}$ that are much larger than the maximal allowed step-sizes with [26]. In other words, as shown in Section 5 by simulations, $\sigma_{\min}$ can be chosen such that the method in [26] with step-size $\alpha = 1/\sigma_{\min}$ diverges, while the novel method (19)–(20) with time-varying step sizes and the safeguard lower bound $\sigma_{\min}$ (hence potentially taking step-size values close or equal to $1/\sigma_{\min}$) still converges.

3.2 Step-size derivation

We now provide a derivation and a justification of the step-size choice (18). For notational simplicity, assume for the rest of this Subsection that $d = 1$ and thus $\mathcal{W} = W$ and $\mathcal{J} = J$. Let each f_i be a strongly convex quadratic function, i.e.,

$$f_i(x_i) = \frac{1}{2} h_i (x_i - b_i)^2,$$

and $H = diag(h_1, \dots, h_n)$, $h_i > 0$, for all i. Then, for the primal error $e^k := x^k - x^*$ and the dual error $\tilde{u}^k := u^k + \nabla F(x^*)$, one can show that the following recursion holds:

$$\begin{bmatrix} e^{k+1} \\ \tilde{u}^{k+1} \end{bmatrix} = \begin{bmatrix} W - \Sigma_k^{-1} H & -\Sigma_k^{-1} \\ (W - I) H & W - J \end{bmatrix} \cdot \begin{bmatrix} e^k \\ \tilde{u}^k \end{bmatrix} \tag{21}$$

We now make a parallel identification between the error dynamics of the centralized SG method for a strongly convex quadratic cost with leading matrix A given in (15) and the error dynamics of the proposed distributed method in (21). Consider first the centralized SG method. The error dynamics matrix is given by $I - \sigma_k^{-1} A$, while the (new) spectral coefficient is sought to fit the secant equation with least mean square deviation: $\sigma_k (x^k - x^{k-1}) = A(x^k - x^{k-1})$. That is, the error dynamics matrix $I - \sigma_k^{-1} A$ is made small by letting $\sigma_k I$ be a scalar matrix approximation for matrix A, i.e., solving

$$\min_{\sigma > 0} \| \sigma s^k - y^k \|^2 = \min_{\sigma > 0} \| \sigma s^k - A s^k \|^2.$$

Now, consider the error dynamics of the proposed distributed method in (21), and specifically focus on the update for the primal error:

$$\begin{aligned} e^{k+1} &= \left(W - \Sigma_k^{-1} H \right) e^k + \Sigma_k^{-1} \widetilde{u}^k \\ &= \left(I - \Sigma_k^{-1} \left[\Sigma_k (I - W) + H \right] \right) e^k + \Sigma_k^{-1} \widetilde{u}^k. \end{aligned} \tag{22}$$

Notice that the second error equation in (21) does not depend on Σ_k. Comparing (15) with (22), we first see that both the primal and the dual error play a role in (22). The effect of the dual error $\widetilde{u}^k$ can be controlled by making Σ_k^{-1} small enough. This motivates the safeguarding of Σ_k from below by $\sigma_{\min}$. Regarding the effect of the primal error e^k, one can see that it influences the error through the matrix $I - \Sigma_k^{-1}\, [\Sigma_k\,(I-W)+H]$. Analogously to the centralized SG case, this matrix can be made small by the following identification

$$A \equiv \Sigma_k\,(I-W)+H, \text{ and } \sigma_k \equiv \Sigma_k.$$

Therefore, we seek Σ_{k+1} as the least mean squares error fit to the following equation

$$\Sigma_{k+1}\left(x^{k+1}-x^k\right) = \left(\Sigma_k(I-W)+H\right)\left(x^{k+1}-x^k\right). \tag{23}$$

For generic (non-quadratic) cost functions, this translates into the following:

$$\Sigma_{k+1}\left(x^{k+1}-x^k\right) = \left(\,\Sigma_k(I-W)\,\right)\left(x^{k+1}-x^k\right) + \left(\,\nabla F(x^{k+1}) - \nabla F(x^k)\,\right). \tag{24}$$

The rationale behind the generalization from (23) to (24) is as follows. For quadratic functions, the term $\left(\,\nabla F(x^{k+1}) - \nabla F(x^k)\,\right)$ equals precisely $H\left(x^{k+1}-x^k\right)$. Therefore, for quadratic functions, equations (23) and (24) are identical. This motivates the argument that (24) can be viewed as a generalization of (23). It is worth noting that a similar argument is used in [27] for motivating the spectral step-size choice for centralized gradient methods.

The (intermediate) inverse step-size matrix Σ'_{k+1} is now obtained by minimizing

$$\left\|\Sigma_{k+1}\left(x^{k+1}-x^k\right) - \left(\,\Sigma_k(I-W)\,\right)\left(x^{k+1}-x^k\right) - \left(\,\nabla F(x^{k+1}) - \nabla F(x^k)\,\right)\right\|^2. \tag{25}$$

This leads to the step-size choice in (18). Finally, to ensure strictly positive step-sizes on the one hand, and a bounded effect of the dual error on the other hand, Σ'_{k+1} is projected entry-wise onto the interval $[\,\sigma_{\min},\, \sigma_{\max}\,]$.

4 Convergence analysis

In Subsection 4.1, we prove that the proposed DSG method, (19)-(20), converges to the solution of problem (1) provided that the spectral coefficients σ_i^k are uniformly bounded with properly chosen constants. In Subsection 4.2, we then prove that the method converges without any a priori upper bound on the step-sizes and with a lower bound on the step-sizes, for a special case of the assumed setting.

4.1 Analysis for the generic case in the presence of safeguarding

For the sake of simplicity, we will restrict our attention to one dimensional case, i.e., $d = 1$, while the general case is proved analogously. Hence, we have $\mathcal{W} = W$ and $\mathcal{J} = J$ in this Section. The following notation and relations are used. Recall that $x^* = \mathbf{1} \otimes y^*$ where y^* is the solution of (1). Define $\tilde{x}^k = x^k - Jx^k$ and $\bar{x}^k = \mathbf{1}^T x^k / n$. Then

$$\tilde{x}^k = x^k - \frac{1}{n}\mathbf{1}\mathbf{1}^T x^k = x^k - \mathbf{1} \otimes \bar{x}^k.$$

Also, for $e^k = x^k - x^*$,

$$(I - J)e^k = (I - J)x^k - (I - J)x^* = \tilde{x}^k - x^* + \mathbf{1} \otimes y^* = \tilde{x}^k.$$

Moreover, notice that $J^2 = J$ and therefore $J(I - J) = 0$, which further implies $J\tilde{x}^k = 0$. Now, for $\tilde{W} = W - J$ we obtain

$$\tilde{W}\tilde{x}^k = (W - J)(I - J)x^k = W\tilde{x}^k - J\tilde{x}^k = W\tilde{x}^k$$

and

$$(I - J)We^k = W(I - J)e^k = W\tilde{x}^k = \tilde{W}\tilde{x}^k. \tag{26}$$

Define $\bar{e}^k = \bar{x}^k - y^*$. So, the following equalities hold

$$e^k = x^k - \mathbf{1} \otimes \bar{x}^k + \mathbf{1} \otimes \bar{x}^k - \mathbf{1} \otimes y^* = \tilde{x}^k + \mathbf{1} \otimes \bar{e}^k. \tag{27}$$

Given that W is doubly stochastic, there follows $Wx^* = x^*$, $\mathbf{1}^T W = \mathbf{1}^T$ and $\mathbf{1}^T(W - I) = 0$. So, multiplying (20) from the left with $\mathbf{1}^T$, we obtain $\bar{u}^{k+1} = \bar{u}^k$, where $\bar{u}^k = \mathbf{1}^T u^k / n$. Since $u^k = 0$, we conclude that

$$\bar{u}^k = 0, \quad k = 0, 1, ... \tag{28}$$

See Lemma 8 in [11] that applies here as well, since the update (20) is a special case of update (16) in [11], with $\mathcal{B} = 0$ defined therein. Moreover, define $\tilde{u}^k = \nabla F(x^*) + u^k$. Using the fact that $\mathbf{1}^T \nabla F(x^*) = 0$ we obtain

$$J\tilde{u}^k = \frac{1}{n}\mathbf{1} \otimes (\mathbf{1}^T u^k + \mathbf{1}^T \nabla F(x^*)) = Ju^k = \mathbf{1} \otimes \bar{u}^k = 0 \tag{29}$$

Now, Assumption A1 together with the Mean value theorem implies that for all $i = 1, 2, ..., n$ and $k = 1, 2, ...,$ there exists θ_i^k such that

$$\nabla f_i(x_i^k) - \nabla f_i(y^*) = \nabla^2 f_i(\theta_i^k)(x_i^k - y^*).$$

Therefore, there exists a diagonal matrix H_k such that

$$\nabla F(x^k) - \nabla F(x^*) = H_k(x^k - x^*) = H_k e^k, \quad 0 \preceq H_k \preceq LI. \tag{30}$$

The following standard lemma in the convex optimization theory, [2] will be used in the proof.

Lemma 4.1. *Let f be μ-strongly convex and ∇f be L- Lipschitz. For $0 < \alpha < \frac{2}{L}$, we have*

$$\|x - \alpha \nabla f(x) - y^*\| \leq \tau \|x - y^*\|, \tag{31}$$

where $\tau = \max\{|1 - \mu\alpha|, |1 - L\alpha|\}$ and y^* is the unique minimizer of f.

The R-linear convergence result for the DSG method is stated in the following theorem. The Theorem corresponds to a worst case analysis that does not take into account the specific form of σ_i^k in (18) but only utilizes information on the safeguarding parameters $\sigma_{\min}$ and $\sigma_{\max}$. Hence, the Theorem may be seen as an extension of Theorem 2 in [23] that assumes node-varying but time-invariant step-sizes (here step-sizes are both node- and time-varying), though we follow here a somewhat different proof path.

Theorem 4.1. *Suppose that the assumptions A1-A3 hold. There exist $0 < \sigma_{\min} < \sigma_{\max}$ such that the sequence $\{x^k\}_{k \in \mathbb{N}}$ generated by DSG method converges R-linearly to the solution of problem (1).*

Proof. Let us first introduce the notation $\sigma_{\min}^{-1} = d_{\max}$ and $\sigma_{\max}^{-1} = d_{\min}$, $\Delta = d_{\max} - d_{\min}$. Choose $d_{\min}, d_{\max}$ such that

$$\frac{d_{\max}}{d_{\min}} < 1 + \frac{\mu}{L} \tag{32}$$

and

$$0 < d_{\min} < d_{\max} < \frac{1 - \lambda_2}{\mu + L}. \tag{33}$$

Define $\delta = \delta(d_{\min}, d_{\max})$ such that $\delta > 0$ and

$$1 > \delta > 1 - d_{\min}\mu + \Delta L. \tag{34}$$

As $1 - d_{\min}\mu + \Delta L < 1$ due to (32), $\delta(d_{\min}, d_{\max})$ is well defined. To simplify notation from now we will write δ to denote $\delta(d_{\min}, d_{\max})$. Due to (33) we have

$$1 - d_{\min}\mu + \Delta L > \lambda_2 + d_{\max}L > \lambda_2$$

and hence

$$0 < \delta - (1 - d_{\min}\mu + \Delta L) < \delta - (\lambda_2 + d_{\max}L) < \delta - \lambda_2. \tag{35}$$

Notice further that $\delta - (1 - d_{\min}\mu + \Delta L)$ is a decreasing function of $d_{\max}$ and Δ and therefore decreasing $d_{\max}, \Delta$ if needed does not violate 35 if (32-33) are satisfied. In fact one can take $d_{\max}, \Delta$ arbitrary small with the corresponding $d_{\min}$ without violating (32)-(35).

Denote $D_k = \Sigma_k^{-1}$ and $d_i^k = (\sigma_i^k)^{-1}$ and notice that $d_i^k \geq d_{\min}$. Subtracting x^* from both sides of (19) and using the fact that $Wx^* = x^*$ we obtain

$$e^{k+1} = We^k - D_k(\nabla F(x^k) + u^k \pm \nabla F(x^*)) = We^k - D_k(\nabla F(x^k) - \nabla F(x^*)) - D_k\tilde{u}^k.$$

From (30) we obtain

$$e^{k+1} = (W - D_k H_k)e^k - D_k\tilde{u}^k. \tag{36}$$

Now, adding $\nabla F(x^*)$ on both sides of (20) we obtain

$$\begin{aligned} \tilde{u}^{k+1} &= Wu^k + (W - I)\nabla F(x^k) + \nabla F(x^*) \pm W\nabla F(x^*) \\ &= W\tilde{u}^k + (W - I)(\nabla F(x^k) - \nabla F(x^*)). \end{aligned} \tag{37}$$

Using (29) and (30) we get

$$\tilde{u}^{k+1} = (W - J)\tilde{u}^k + (W - I)H_k e^k. \tag{38}$$

Taking the norm and using (27), we obtain

$$\|\tilde{u}^{k+1}\| \leq \lambda_2\|\tilde{u}^k\| + (1 - \lambda_n)L(\|\tilde{x}^k\| + \sqrt{n}|\bar{e}^k|). \tag{39}$$

Lemma 2.1 with $c_1 = \lambda_2,\ c_2 = c_3 = (1 - \lambda_n)L$ yields

$$\|\tilde{u}\|^{\delta,K} \leq \frac{c_2}{\delta - c_1}(\|\tilde{x}\|^{\delta,K} + |\sqrt{n}\bar{e}|^{\delta,K}) + \frac{\delta}{\delta - c_1}\|\tilde{u}^0\|, \tag{40}$$

with $\delta - c_1 > 0$ due to (35). Define $\gamma_1 := c_2/(\delta - c_1)$.

Multiplying both sides of (19) from the left with $\frac{1}{n}\mathbf{1}^T$ and using $\mathbf{1}^T W = \mathbf{1}^T$, (28) and $\mathbf{1}^T \nabla F(x^*) = 0$ we obtain

$$\begin{aligned} \bar{x}^{k+1} &= \bar{x}^k - \frac{1}{n}\sum_{i=1}^n d_i^k \nabla f_i(x_i^k) - \frac{1}{n}\sum_{i=1}^n d_i^k u_i^k \pm \frac{1}{n}\sum_{i=1}^n d_{\min} \nabla f_i(\bar{x}^k) + \frac{1}{n}\sum_{i=1}^n d_{\min} u_i^k \\ &= \bar{x}^k - \frac{d_{\min}}{n}\sum_{i=1}^n \nabla f_i(\bar{x}^k) + \frac{d_{\min}}{n}\sum_{i=1}^n \nabla f_i(\bar{x}^k) - \frac{1}{n}\sum_{i=1}^n d_i^k \nabla f_i(x_i^k) \\ &+ \frac{1}{n}\sum_{i=1}^n (d_{\min} - d_i^k) u_i^k \end{aligned} \tag{41}$$

So, after subtracting y^* from both sides, we obtain

$$\begin{aligned} \bar{e}^{k+1} &= \bar{e}^k - \frac{d_{\min}}{n}\sum_{i=1}^n \nabla f_i(\bar{x}^k) + \frac{1}{n}\sum_{i=1}^n d_{\min} \nabla f_i(\bar{x}^k) - \frac{1}{n}\sum_{i=1}^n d_i^k \nabla f_i(x_i^k) \\ &+ \frac{1}{n}\sum_{i=1}^n (d_{\min} - d_i^k) u_i^k \\ &= \bar{e}^k - \frac{d_{\min}}{n}\sum_{i=1}^n \nabla f_i(\bar{x}^k) + \frac{1}{n}\sum_{i=1}^n d_{\min} (\nabla f_i(\bar{x}^k) - \nabla f_i(x_i^k)) \\ &- \frac{1}{n}\sum_{i=1}^n (d_i^k - d_{\min}) \nabla f_i(x_i^k) + \frac{1}{n}\sum_{i=1}^n (d_{\min} - d_i^k) u_i^k \pm \frac{1}{n}\sum_{i=1}^n (d_i^k - d_{\min}) \nabla f_i(y^*) \\ &= \bar{e}^k - \frac{d_{\min}}{n}\sum_{i=1}^n \nabla f_i(\bar{x}^k) + \frac{1}{n} d_{\min} \sum_{i=1}^n (\nabla f_i(\bar{x}^k) - \nabla f_i(x_i^k)) \\ &- \frac{1}{n}\sum_{i=1}^n (d_i^k - d_{\min}) (\nabla f_i(x_i^k) - \nabla f_i(y^*)) \\ &+ \frac{1}{n}\sum_{i=1}^n (d_{\min} - d_i^k) u_i^k + \frac{1}{n}\sum_{i=1}^n (d_{\min} - d_i^k) \nabla f_i(y^*). \end{aligned} \tag{42}$$

Given that $\tilde{u}^k = \nabla F(x^*) + u^k$, we have $\tilde{u}_i^k = u_i^k + \nabla f_i(y^*)$ and the above inequalities imply

$$\begin{aligned}\bar{e}^{k+1} &= \bar{e}^k - \frac{d_{\min}}{n}\sum_{i=1}^n \nabla f_i(\bar{x}^k) + \frac{d_{\min}}{n}\sum_{i=1}^n (\nabla f_i(\bar{x}^k) - \nabla f_i(x_i^k)) \\ &- \frac{1}{n}\sum_{i=1}^n (d_i^k - d_{\min})(\nabla f_i(x_i^k) - \nabla f_i(y^*)) \\ &+ \frac{1}{n}\sum_{i=1}^n (d_{\min} - d_i^k)\tilde{u}_i^k. \end{aligned} \tag{43}$$

Assumption A1 implies that

$$\|\nabla f_i(x_i^k) - \nabla f_i(\bar{x}^k)\| \leq l_i|\tilde{x}_i^k|. \tag{44}$$

which further implies

$$|\frac{d_{\min}}{n}\sum_{i=1}^n (\nabla f_i(\bar{x}^k) - \nabla f_i(x_i^k))| \leq \|\tilde{x}^k\|_1 \frac{d_{\min}}{n}\sum_{i=1}^n l_i = \|\tilde{x}^k\|_1 \frac{d_{\min}}{n} L. \tag{45}$$

Similarly we obtain

$$|\frac{1}{n}\sum_{i=1}^n (d_i^k - d_{\min})(\nabla f_i(x_i^k) - \nabla f_i(y^*))| \leq \|e^k\|_1 \frac{\Delta}{n} L \tag{46}$$

and

$$|\frac{1}{n}\sum_{i=1}^n (d_{\min} - d_i^k)\tilde{u}_i^k| \leq \|\tilde{u}^k\|_1 \frac{\Delta}{n} \tag{47}$$

Furthermore, Lemma 4.1 implies

$$|\bar{e}^k - \frac{d_{\min}}{n}\sum_{i=1}^n \nabla f_i(\bar{x}^k)| \leq \tau|\bar{e}^k|$$

with $\tau = \max\{|1 - \mu d_{\min}|, |1 - L d_{\min}|\}$. Since (32) implies that $d_{\min} < 1/L$, we obtain that $\tau = 1 - \mu d_{\min}$. Putting all together we obtain

$$|\bar{e}^{k+1}| \leq (1 - d_{\min}\mu)|\bar{e}^k| + \frac{d_{\min}}{n} L\|\tilde{x}^k\|_1 + \frac{\Delta}{n}(L\|e^k\|_1 + \|\tilde{u}^k\|_1).$$

Using the norm equivalence $\|\cdot\|_1 \leq \sqrt{n}\|\cdot\|_2$, and multiplying both sides of the previous inequality with $\sqrt{n}$, we get

$$\sqrt{n}|\bar{e}^{k+1}| \leq (1-d_{\min}\mu)\sqrt{n}|\bar{e}^k| + d_{\min}L\|\tilde{x}^k\| + \Delta(L\|e^k\| + \|\tilde{u}^k\|).$$

Furthermore, taking (27) into account, the previous inequality becomes

$$\sqrt{n}|\bar{e}^{k+1}| \leq (1-d_{\min}\mu + \Delta L)\sqrt{n}|\bar{e}^k| + (d_{\min}+\Delta)L\|\tilde{x}^k\| + \Delta\|\tilde{u}^k\|. \quad (48)$$

Lemma 2.1 with $\tilde{c}_1 = 1 - d_{\min}\mu + \Delta L,\ \tilde{c}_2 = d_{\max}L,\ \tilde{c}_3 = \Delta$ implies

$$|\sqrt{n}\bar{e}|^{\delta,K} \leq \frac{1}{\delta - \tilde{c}_1}(\tilde{c}_2\|\tilde{x}\|^{\delta,K} + \tilde{c}_3\|\tilde{u}\|^{\delta,K} + \delta|\sqrt{n}\bar{e}^0|), \quad (49)$$

for $\delta \in (\tilde{c}_1, 1)$. Notice that (35) implies that $\delta - \tilde{c}_1 > 0$. Define

$$\theta_2 = \frac{\tilde{c}_3}{\delta - \tilde{c}_1},\ \gamma_2 = \frac{\tilde{c}_2}{\delta - \tilde{c}_1}.$$

Incorporating (40) into (49) and rearranging, we obtain

$$|\sqrt{n}\bar{e}|^{\delta,K} \leq \frac{\gamma_2 + \theta_2\gamma_1}{1-\theta_2\gamma_1}\|\tilde{x}\|^{\delta,K} + \frac{\theta_2\delta\|\tilde{u}^0\|}{(\delta - c_1)(1-\theta_2\gamma_1)} + \frac{\delta|\sqrt{n}\bar{e}^0|}{(\delta-\tilde{c}_1)(1-\theta_2\gamma_1)}, \quad (50)$$

provided that $\theta_2\gamma_1 < 1$. This condition reads

$$\frac{\Delta}{\delta - (1 - d_{\min}\mu + \Delta L)}\frac{(1-\lambda_n)L}{\delta - \lambda_2} < 1. \quad (51)$$

Clearly, there exists $\delta, d_{\min}, d_{\max}$ such that for $d_{\max}, \Delta$ small enough (51) holds as the left-hand side expression in (51) is increasing function of $d_{\max}, \Delta$ and the corresponding $d_{\min}$ satisfies (33).

Now, multiplying (36) from the left with $I - J$ and using (4.1) and (26), we have

$$\tilde{x}^{k+1} = \tilde{W}\tilde{x}^k - (I-J)D_kH_ke^k - (I-J)D_k\tilde{u}^k.$$

Furthermore, (27) implies

$$\tilde{x}^{k+1} = (\tilde{W} - (I-J)D_kH_k)\tilde{x}^k - (I-J)D_kH_k(\mathbf{1}\otimes\bar{e}^k) - (I-J)D_k\tilde{u}^k.$$

The inequality $\|\tilde{W}\| \leq \lambda_2$ yields

$$\|\tilde{x}^{k+1}\| \leq (\lambda_2 + d_{\max}L)\|\tilde{x}^k\| + d_{\max}L\sqrt{n}|\bar{e}^k| + d_{\max}\|\tilde{u}^k\|.$$

Again, Lemma 2.1 with $\hat{c}_1 = \lambda_2 + d_{\max} L,\ \hat{c}_2 = d_{\max} L,\ \hat{c}_3 = d_{\max}$, implies

$$\|\tilde{x}\|^{\delta,K} \leq \frac{\hat{c}_2}{\delta - \hat{c}_1} |\sqrt{n}\bar{e}|^{\delta,K} + \frac{\hat{c}_3}{\delta - \hat{c}_1} \|\tilde{u}\|^{\delta,K} + \frac{\delta}{\delta - \hat{c}_1} \|\tilde{x}^0\|,$$

with $\delta - \hat{c}_1 > 0$ due to (35). Define $\gamma_3 = \hat{c}_2/(\delta - \hat{c}_1)$ and $\theta_3 = \hat{c}_3/(\delta - \hat{c}_1)$. Using (40) and rearranging, we obtain

$$\|\tilde{x}\|^{\delta,K} \leq \frac{\gamma_3 + \theta_3}{1 - \theta_3\gamma_1} |\sqrt{n}\bar{e}|^{\delta,K} + \frac{\theta_3 \delta}{(\delta - c_1)(1 - \theta_3\gamma_1)} \|\tilde{u}^0\| + \frac{\delta}{(\delta - \hat{c}_1)(1 - \theta_3\gamma_1)} \|\tilde{x}^0\|, \tag{52}$$

with $\theta_3\gamma_1 < 1$ for $d_{\max}$ small enough, due to the fact that

$$\theta_3\gamma_1 = \frac{d_{\max}}{\delta - (\lambda_2 + d_{\max} L)} \frac{(1 - \lambda_n)L}{\delta - \lambda_2}$$

is an increasing function of $d_{\max}$.

Finally, considering (50), (52) and Theorem 2.1, we conclude that $\tilde{x}^k$ and $\bar{e}^k$ tend to zero R-linearly if

$$\frac{\gamma_2 + \theta_2\gamma_1}{1 - \theta_2\gamma_1} \frac{\gamma_3 + \theta_3}{1 - \theta_3\gamma_1} < 1. \tag{53}$$

The definition of γ_2 implies that it can be arbitrary small if $d_{\max}$ is small enough. As already stated, $\theta_2\gamma_1/(1-\theta_2\gamma_1)$ is increasing function of Δ Therefore, taking Δ small enough, with the proper choice of $d_{\min}$, one can make the first term in (53) arbitrary small. On the other hand,

$$\theta_3 + \gamma_3 = \frac{d_{\max}(L+1)}{\delta - (\lambda_2 + d_{\max} L)}$$

is again increasing function of $d_{\max}$ as is the function $(1 - \theta_3\gamma_1)^{-1}$. So, for $d_{\max}, \Delta$ small enough and $d_{\min}$ such that (32-33) hold, the inequality (52) holds and the statement is proved. □

4.2 Analysis for a special case without step-size upper bounds

Establishing convergence for generic costs in the absence of safeguarding or under a less restrictive safeguarding is very challenging. We show here that

DSG achieves convergence without any a priori safeguarding upper bound on the step-sizes and under an assumed safeguarding lower bound on the step-sizes, for a special case of the consensus problem and for a special structure of weight matrix W.

Specifically, we consider $f_i : \mathbb{R} \to \mathbb{R}$, with:

$$f_i(y) = \frac{1}{2}(y - a_i)^2, \tag{54}$$

for some $a = (a_1, \ldots, a_n) \in \mathbb{R}^n$. Note that here the solution to (1) equals $y^\star = \frac{1}{n}\mathbf{1}^T a$. Denote as before, for future reference, $x^\star = y^\star \mathbf{1}$, the $n \times 1$ vector whose entries equal the solution to (1).

Let us further assume that the network is fully connected and that the matrix W is given by

$$W = (1 - \theta)I + \theta J, \tag{55}$$

for some $\theta \in (0, 1)$, where we recall the $n \times n$ ideal consensus matrix $J = (1/n)\mathbf{1}\mathbf{1}^T$. Note that, while the network is fully connected, the weight matrix does not equal the ideal consensus matrix J. This example hence corresponds to a non-trivial distributed optimization scenario where algorithms of type (7)-(8) or (15)-(17) require an iterative process to correctly diffuse information for convergence.

We first need the following Lemma on the method in (7)-(8), proved in the Appendix. The Lemma shows that, for the special case of consensus, the admissible size of the step-size α with algorithm (7)-(8), under which the algorithm is convergent, can be made larger than what standard analysis for generic costs says [26]. On the other hand, for a sufficiently large α, algorithm (7)-(8) is divergent.

Lemma 4.2. *Consider optimization problem (1) with the f_i's as in* (54)*. Let the underlying network and weight matrix W satisfy assumptions A2 and A3, and moreover assume that W is positive definite. Consider algorithm (7)-(8) with step-size $\alpha > 0$, and let the initial iterates satisfy: $\mathbf{1}^T x^0 = \mathbf{1}^T a, \mathbf{1}^T z^0 = 0$. Then, the sequence x^k generated by* (8)–(9) *converges R-linearly to the solution $x^\star = \frac{1}{n}(\mathbf{1}^T a)\mathbf{1}$ if $\alpha \leq 1/2$, and it diverges, in the sense that $\|x^k\| \to \infty$, when $\alpha > 2$.*

We now state our result on the DSG method.

Proposition 4.3. *Consider optimization problem (1) with the f_i's as in* (54) *and the weight matrix W as in (55), with $\theta \in [3/4, 1)$. Further, let the initial iterates of the DSG method in (15)-(17) be such that: $\mathbf{1}^T x^0 =$*

$\mathbf{1}^T a, \mathbf{1}^T z^0 = 0$. *Assume next that* $\sigma_{\max} \in [2,3]$, *and* $\sigma_{\min} = 0$. *Assume further that the initial step sizes* $1/\sigma_i^0$ *are equal across all nodes* $i = 1, ..., n$, *with* $\sigma_i^0 = \sigma \in [\sigma_{\min}, \sigma_{\max}]$. *Then, the sequence* x^k *generated by the DSG method (15)-(17) converges R-linearly to the solution* $x^\star = \frac{1}{n}(\mathbf{1}^\top a)\mathbf{1}$.

We now prove Proposition 4.3.

Proof. Consider the DSG algorithm in (16)–(18) under the setting of Proposition 4.3. Note that $\nabla F(x) = x - a$.

Also, the inverse-step size at node i and iteration k becomes:

$$\sigma_i^{k+1} = P_{[0,\sigma_{\max}]}\left(1 + \sigma_i^{k-1} \sum_{j \in \bar{O}_i} w_{ij}(1 - \frac{s_j^{k-1}}{s_i^{k-1}})\right). \tag{56}$$

The update rule (16)–(17) simplifies to the following:

$$x^{k+1} = Wx^k - \Sigma_k^{-1} z^k,$$
$$z^{k+1} = Wz^k + x^{k+1} - x^k.$$

In view of the assumed initialization, we have

$$x^1 = Wx^0 - (1/\sigma)z^0,\ \mathbf{1}^T(x^1 - x^0) = \mathbf{1}^T(Wx^0 - x^0) = 0$$

and therefore $\mathbf{1}^T s^0 = 0$, i.e., $\sum_{i=1}^n s_i^0 = 0$. This also means that:

$$\mathbf{1}^T z^1 = \mathbf{1}^T W z^0 + \mathbf{1}^T(x^1 - x^0) = \mathbf{1}^T z^0 + \mathbf{1}^T s^0 = 0.$$

We next analyze the step-sizes of the nodes at iteration $k = 1$. Denote by $\sigma \in [0, \sigma_{\max}]$ the initial step-size assumed equal at all nodes, and consider the step-size at node 1 at the next iteration:

$$\begin{aligned}
\sigma_1^1 &= \mathcal{P}_{[0,\sigma_{\max}]}\left(1 + \sigma_1^0 \sum_{j \in \bar{O}_1} w_{1j}(1 - \frac{s_j^0}{s_1^0})\right) \\
&= \mathcal{P}_{[0,\sigma_{\max}]}\left(1 + \theta\sigma(\frac{n-1}{n} - \frac{1}{n}\sum_{j \neq 1} \frac{s_j^0}{s_1^0})\right) \\
&= \mathcal{P}_{[0,\sigma_{\max}]}(1 + \sigma\theta) \\
&= \min\{1 + \sigma\theta, \sigma_{\max}\}.
\end{aligned}$$

Here, we used the fact that $\sum_{i=1}^{n} s_i^0 = 0$, and so $s_1^0 = -\sum_{j \neq 1} s_j^0$. It is easy to see now, due to symmetry, that we also have $\sigma_j^1 = \sigma_1^1, j = 2, \ldots, n$, and so

$$\sigma_1^1 = \sigma_2^1 = \ldots = \sigma_n^1 = \sigma^1 := \min\{1 + \sigma\theta, \sigma_{\max}\}.$$

Consider now the second algorithm iteration. Because $\sigma_i^1 = \sigma^1$, for all $i = 1, ..., n$, and $\mathbf{1}^T z^1 = 0$, we have: $x^2 = Wx^1 - 1/(\sigma^1)s^1$, and $\mathbf{1}^T(x^2 - x^1) = \mathbf{1}^T(Wx^1 - \sigma_1^{-1}s^1 - x^1) = \mathbf{1}^T(x^1 - x^1) = 0$, i.e., $\mathbf{1}^T s^1 = 0$. This further implies that $\mathbf{1}^T z^2 = 0$, and

$$\sigma_1^2 = \sigma_2^2 = \ldots = \sigma_n^2 = \min\{1 + \sigma_1\theta, \sigma_{\max}\} = \min\{1 + \theta + \sigma\theta^2, \sigma_{\max}\}.$$

Now, by induction, it follows that, across all iterations k, all nodes employ the same step-size $1/\sigma^k$, where

$$\sigma^k = \sigma_1^k = \ldots \sigma_n^k = \min\{1 + \theta + \ldots + \theta^{k-1} + \sigma\theta^k, \sigma_{\max}\}.$$

Next, because $\sigma_{\max} \leq 3$, and $\theta \geq 3/4$ (as it is assumed in Proposition 4.3), we can see that, at a certain iteration $k = k'$, we have that:

$$1 + \theta + \ldots + \theta^{k-1} + \sigma\theta^{k'} \geq \sigma_{\max},$$

and so at all nodes $i = 1, ..., n$, we have:

$$\sigma^{k'} := \sigma_1^{k'} = \ldots \sigma_n^{k'} = \sigma_{\max}.$$

Furthermore, in view of (56) and the fact that $\sigma_{\max} \leq 3$, and $\theta \geq 3/4$, we also have that, for all $k \geq k'$, there holds:

$$\sigma^k := \sigma_1^k = \ldots \sigma_n^k = \sigma_{\max}.$$

However, this means that, starting from a finite iteration k' onwards, the algorithm utilizes a constant step-size $\alpha = 1/\sigma_{\max}$ equal across all nodes and hence reduces to (8)–(9). Furthermore, because $\sigma_{\max} \geq 2$, we have that $\alpha \leq 1/2$, and hence, applying Lemma 4.2, we conclude that the DSG algorithm converges R-linearly to the solution $x^\star$. The proof is complete.

Proposition 4.3 sets the safeguarding lower bound on the step-size $1/\sigma_{\max} \in [1/3, 1/2]$, and the safeguarding step-size upper bound on $1/\sigma_{\min} = +\infty$. The proposition hence shows that, under the considered setting, DSG converges without any a priori upper bound on the step-sizes and with a lower bound on the step-sizes. Proposition 4.3 hence provides an example where DSG is significantly more robust in terms of the step-sizes admissible range than (8)–(9). The proposition also helps in providing insights as to why DSG converges under a wider admissible step-sizes range in simulations for more generic and more practical scenarios (see Section 5).

5 Numerical experiments

This section provides a numerical example to illustrate the performance of the proposed distributed spectral method.

We consider the problem with strongly convex local quadratic costs; that is, for each $i = 1, ..., n$, let $f_i : \mathbb{R}^d \to \mathbb{R}$, $f_i(x) = \frac{1}{2}(x - b_i)^T A_i (x - b_i)$, $d = 10$, where $b_i \in \mathbb{R}^d$ and $A_i \in \mathbb{R}^{d \times d}$ is a symmetric positive definite matrix. The data pairs A_i, b_i are generated at random, independently across nodes, as follows. Each b_i's entry is generated mutually independently from the uniform distribution on $[1, 31]$. Each B_i is generated as $B_i = Q_i\, D_i\, Q_i^T$; here, Q_i is the matrix of orthonormal eigenvectors of $\frac{1}{2}(\widehat{B}_i + \widehat{B}_i^T)$, and $\widehat{B}_i$ is a matrix with independent, identically distributed (i.i.d.) standard Gaussian entries; D_i is a diagonal matrix with the diagonal entries drawn in an i.i.d. fashion from the uniform distribution on $[1, 101]$.

The network is a $n = 30$-node instance of the random geometric graph model with the communication radius $r = \sqrt{\frac{\ln(n)}{n}}$, and it is connected. The weight matrix W is set as follows: for $\{i, j\} \in E$, $i \neq j$, $w_{ij} = \frac{1}{2(1+\max\{d_i, d_j\})}$, where d_i is the node i's degree; for $\{i, j\} \notin E$, $i \neq j$, $w_{ij} = 0$; and $w_{ii} = 1 - \sum_{j \neq i} w_{ij}$, for all $i = 1, ..., n$.

The proposed DSG method is compared with the method in [26]. This is a meaningful comparison as the method in [26] is a state-of-the-art distributed first order method, and the proposed method is based upon it. The comparison thus allows to assess the benefits of incorporating spectral-like step-sizes into distributed first order methods. As an error metric, the relative error averaged across nodes

$$\frac{1}{n} \sum_{i=1}^{n} \frac{\|x_i - y^*\|}{\|y^*\|}, \ y^* \neq 0.$$

is used.

All parameters for both algorithms are set in the same way, except for step-sizes. With the method in [26], the step-size is $\alpha = 1/(3L)$, where

$L = \max_{i=1,\ldots,n} \mu_i$, and μ_i is the maximal eigenvalue of A_i. This step-size corresponds to the maximal possible step-size for the method in [26] as empirically evaluated in [26]. It is worth noting that, with [26], the maximal possible step-size may not necessarily correspond to the best possible choice. However, the optimal step-size is dependent on the cost functions' and network parameters, and it may be very resource-consuming in many applications. (See ahead Figure 2 for the hand-optimized step-size case.) With the DSG method, at all nodes the initial step-size value is set to $1/(3L)$. The safeguard parameters on the step-sizes are set to 10^{-8} (lower threshold for safeguarding), and $10 \times \frac{1}{3L}$ (upper threshold for safeguarding). Hence, the step-sizes in DSG are allowed to reach up to 10 times larger values than the maximal possible value with the method from [26].

Figure 1 (top) plots the relative error versus number of iterations with the two methods. One can see that the DSG method significantly improves the convergence speed. For example, to reach the relative error 0.01, the DSG method requires about 340 iterations, while the method in [26] takes about 560 iterations for the same target accuracy; this corresponds to savings of about 40%.

Figure 1 (bottom) repeats the experiment for a $n = 100$-node connected random geometric graph, with the remaining data and network parameters as before. We can see that the DSG method achieves similar gains. For example, for the 0.01 accuracy, the DSG method needs about 650 iterations, while the method in [26] needs about 1150, corresponding to decrease of about 43% in computational costs. We also report that the method in [26] and step-size equal to $1/\sigma_{\min} = 10/(3L)$ diverges. This demonstrates that, on the considered example, DSG exhibits convergence under a significantly wider set of step-sizes than [26].

Figure 2 plots the error versus iteration number for the DSG method and the method in [26] with various values of the step-size α. Specifically, $\alpha = 1/(2L)$ was the maximal possible choice for which the method in [26] is convergent on the considered example. On the other hand, decreasing step-size below $\alpha = 1/(100L)$ yields poorer convergence than for the case $\alpha = 1/(100L)$. We can see that there exist choices of α for which [26] converges faster than DSG; an optimal value of α is close to $1/(20L)$ for the considered example. However, for other choices of α, DSG is faster; this happens, e.g., for $\alpha = 1/(3L)$ or $\alpha = 1/(100L)$. We can see that DSG achieves good performance without the need for aligning or hand-tuning of step-sizes.

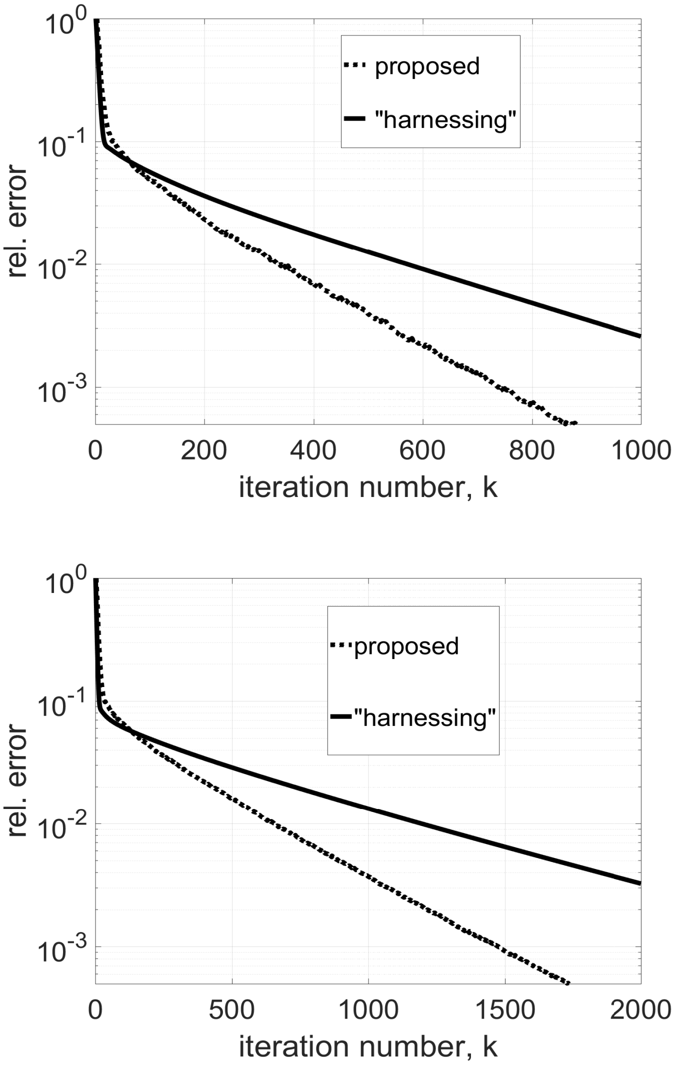

Figure 1: Relative error versus iteration number for the method in [26] ("harnessing", solid line) and the proposed method (dotted line).

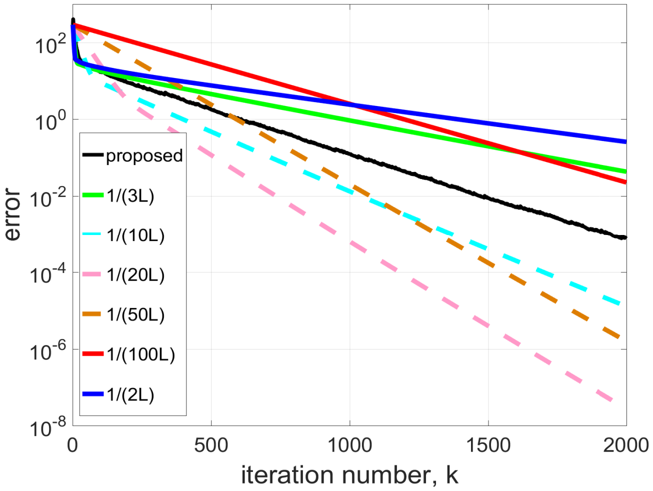

Figure 2: Error versus iteration number for the method in [26] with various step-size values, and for the proposed DSG method.

6 Conclusion

The method proposed in this paper, DSG, is a distributed version of the Spectral Gradient method for unconstrained optimization problems. Following the approach of exact distributed gradient methods in [23] and [26], at each iteration the nodes update two quantities – the local approximation of the solution and the local approximation of the average gradient. The key novelty developed here is the step size selection which is defined in a spectral-like manner. Each node approximates the local Hessian by a scalar matrix thereby incorporating a degree of second order information in the gradient method. The spectral-like step-size coefficients are derived by exploiting an analogy with the error dynamics of the classical spectral method for quadratic functions and embedding this dynamics into a primal-dual framework. This step-size calculation is computationally cheap and does not incur additional communication overhead. Under a set of standard assumptions regarding the objective functions and assuming a connected communication network, the DSG method generates a sequence of iterates which converges R-linearly to the exact solution of the aggregate objective function. A distinctive property of DSG is that it works under a broader range (and hence possibly larger) step-sizes than existing exact first order methods like [26]. Moreover, DSG provides a good automatic tuning of the nodes' local step sizes, despite the absence of global coordination and beforehand tuning. The spectral gradient method is well known for its efficiency in classical, centralized optimization. Preliminary numerical tests demonstrate similar gains of incorporating spectral step-sizes in the distributed setting as well.

Appendix

Proof of Lemma 4.2. Consider algorithm (8)–(9). For the special case considered here, the update rule (7)-(8) becomes:

$$x^{k+1} = W x^k - \alpha\, z^k, \tag{57}$$

$$z^{k+1} = W z^k + x^{k+1} - x^k. \tag{58}$$

Denote by ξ^k the $(2n)\times 1$ vector defined by $\xi^k = \left(e^k\,, z^k\right)$, where $e^k = x^k - x^\star$. Then, it is easy to show that ξ^k obeys the following recursion:

$$\xi^{k+1} = E\,\xi^k,$$

where E is the $(2n)\times(2n)$ matrix with the following $n \times n$ blocks:[1]

$$E_{11} = W - J,\;\; E_{12} = -\alpha\, I,\;\; E_{21} = W - I,\;\; E_{22} = W - \alpha\, I.$$

Consider the eigenvalue decomposition of matrix $W = Q \Lambda Q^T$, where Q is the matrix of orthonormal eigenvectors, and Λ is the matrix of eigenvalues ordered in a descending order. We have that $\lambda_1 = 1$, and $\lambda_i \in (0,1)$, $i \neq 1$. Note that the matrix E can now be decomposed as follows:

$$E = \widehat{Q}\, \widehat{P}\, \widehat{\Lambda}\, \widehat{P}^T\, \widehat{Q}^T.$$

Here, $\widehat{Q}$ is the $(2n)\times(2n)$ orthonormal matrix with the $n \times n$ blocks at positions (1,1) and (2,2) equal to Q, and zero-off diagonal $n \times n$ blocks; and

[1] Lemma 4.3 can be proved similarly, if we work with representation (9)-(10) instead of (7)-(8). The corresponding error recursion matrix then becomes as in (20), with $\Sigma_k^{-1} = \alpha\, I$ and $H = I$. The matrix has the same blocks as E up to a permutation, and the results through the alternative analysis will be equivalent.

$\widehat{P}$ is an appropriate permutation matrix. Furthermore, $\widehat{\Lambda}$ is the $(2n) \times (2n)$ block-diagonal matrix with the 2×2 diagonal blocks $D_1, ..., D_n$, as follows:

$$D_1 = \begin{bmatrix} 0 & -\alpha \\ 0 & 1-\alpha \end{bmatrix}, \;\; D_i = \begin{bmatrix} \lambda_i & -\alpha \\ \lambda_i - 1 & \lambda_i - \alpha \end{bmatrix}, \;\; i \neq 1.$$

It is then clear that the matrix E has the same eigenvalues as the matrix $\widehat{\Lambda}$, and hence the two matrices have the same spectral radius. Next, by evaluating the eigenvalues of the 2×2 matrices D_i, $i = 1, ..., n$, it is straightforward to verify sufficient conditions on α such that the spectral radius $\rho(\widehat{\Lambda})$ is strictly less than one, and such that $\rho(\widehat{\Lambda})$ is strictly greater than one. Namely, for $i \neq 1$, it is easy to show that $\rho(D_i) < 1$ if and only if $\alpha \in (0, \overline{\alpha}_i)$, where $\overline{\alpha}_i = \frac{(1+\lambda_i)^2}{2}$. On the other hand, for $i = 1$, we have that $\rho(D_1) = 1 - \alpha < 1$. In view of the fact that $\lambda_i > 0$, $i = 2, ..., n$, the latter implies that, when $\alpha \leq 1/2$, we have that $\rho(\widehat{\Lambda}) < 1$; also, $\rho(\widehat{\Lambda}) > 1$ whenever $\alpha > 2$. This in particular implies that x^k converges R-linearly to $x^\star$ if $\alpha \leq 1/2$, and that x^k diverges, when $\alpha > 2$. The proof is complete.

References

[1] Barzilai J, Borwein JM, Two Point Step Size Gradient Methods, IMA Journal of Numerical Analysis, 8 (1988), 141 - 148.

[2] Bertsekas, D.P., Nonlinear Programming, Athena Scientific, Belmont, 1997.

[3] Birgin, E.G, Martínez, J.M, Raydan M., Nonmonotone Spectral Projected Gradient Methods on Convex Sets, SIAM Journal on Optimization, 10, (2000), 1196-1211.

[4] Birgin, E.G., Martínez, J.M., Raydan M., Algorithm 813: SPG - Software for Convex- Constrained Optimization, ACM Transactions on Mathematical Software, 27 (2001), 340-349.

[5] Birgin, E.G., Martínez, J.M., Raydan M Inexact Spectral Projected Gradient Methods on Convex Sets, IMA Journal of Numerical Analysis, 23, (2003), 539-559.

[6] Birgin, E.G., Martínez, J.M., Raydan M Spectral Projected Gradient Methods: Review and Perspectives, Journal of Statistical Software 60(3), (2014), 1-21.

[7] Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning, Volume 3, Issue 1, (2011) pp. 1-122.

[8] Cattivelli, F., Sayed, A. H., Diffusion LMS strategies for distributed estimation, *IEEE Transactions on Signal Processing*, vol. 58, no. 3, (2010) pp. 1035–1048.

[9] Dai, Y.H., Liao, L.Z., R-Linear Convergence of the Barzilai and Borwein Gradient Method, IMA Journal on Numerical Analysis, 22 (2002), 1-10.

[10] Desoer, C., Vidyasagar, M., Feedback Systems: Input-Output Properties, SIAM 2009.

[11] Jakovetić, D., A Unification and Generalization of Exact Distributed First Order Methods, arxiv preprint, arXiv:1709.01317, 2017.

[12] P. Di Lorenzo and G. Scutari, Distributed nonconvex optimization over networks, in IEEE International Conference on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2015, pp. 229-232.

[13] Jakovetić, D., Xavier, J., Moura, J. M. F., Fast distributed gradient methods, *IEEE Transactions on Automatic Control*, vol. 59, no. 5, (2014) pp. 1131–1146.

[14] Jakovetić, D., Moura, J. M. F., Xavier, J., Distributed Nesterov-like gradient algorithms, in *CDC'12, 51st IEEE Conference on Decision and Control*, Maui, Hawaii, December 2012, pp. 5459–5464.

[15] Jakovetić, D., Bajović, D., Krejić, N., Krklec Jerinkić, N., Newton-like Method with Diagonal Correction for Distributed Optimization, SIAM J. Optimization, 27 (2), (2017), 1171-1203.

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

[17] Kar, S., Moura, J. M. F., Distributed Consensus Algorithms in Sensor Networks With Imperfect Communication: Link Failures and Channel Noise, *IEEE Transactions on Signal Processing*, vol. 57, no. 1, (2009) pp. 355–369.

[18] Lopes, C., Sayed, A. H., Adaptive estimation algorithms over distributed networks, in *21st IEICE Signal Processing Symposium*, Kyoto, Japan, Nov. 2006.

[19] Mokhtari, A., Ling, Q., Ribeiro, A., Network Newton–Part I: Algorithm and Convergence, 2015, available at: http://arxiv.org/abs/1504.06017

[20] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, DQM: Decentralized Quadratically Approximated Alternating Direction Method of Multipliers, to appear in IEEE Trans. Sig. Process., 2016, DOI: 10.1109/TSP.2016.2548989

[21] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, A Decentralized Second Order Method with Exact Linear Convergence Rate for Consensus Optimization, 2016, available at: http://arxiv.org/abs/1602.00596

[22] Mota, J., Xavier, J., Aguiar, P., Püschel, M., Distributed optimization with local domains: Applications in MPC and network flows, *to appear in IEEE Transactions on Automatic Control*, 2015.

[23] Nedic, A., Olshevsky, A., Shi, W., Uribe, C.A., Geometrically convergent distributed optimization with uncoordinated step-sizes, arXiv preprint, arXiv:1609.05877, 2016.

[24] Nedic, A., Olshevsky, A., Shi, W., Achieving Geometric Convergence for Distributed Optimization over Time-Varying Graphs, arxiv preprint, arXiv:1607.03218, 2016.

[25] Nedić, A., Ozdaglar, A., Distributed subgradient methods for multiagent optimization, *IEEE Transactions on Automatic Control*, vol. 54, no. 1, (2009) pp. 48–61.

[26] Qu, G., Li, N., Harnessing smoothness to accelerate distributed optimization, IEEE Transactions on Control of Network Systems (to appear)

[27] Raydan, M., On the Barzilai and Borwein Choice of Steplength for the Gradient Method, IMA Journal of Numerical Analysis, 13 (1993), 321-326.

[28] Raydan M, Barzilai and Borwein Gradient Method for the Large Scale Unconstrained Minimization Problem, SIAM Journal on Optimization 7 (1997), 26 - 33.

[29] Schizas, I. D. , Ribeiro, A., Giannakis, G. B., Consensus in ad hoc WSNs with noisy links – Part I: Distributed estimation of deterministic signals, *IEEE Transactions on Signal Processing*, vol. 56, no. 1, (2009) pp. 350–364.

[30] Shi, W., Ling, Q., Wu, G., Yin, W., EXTRA: an Exact First-Order Algorithm for Decentralized Consensus Optimization, *SIAM Journal on Optimization*, No. 25 vol. 2, (2015) pp. 944-966.

[31] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant step- sizes, in IEEE Conference on Deci- sion and Control (CDC), 2015, pp. 2055-2060.