Preprints of the 3rd IFAC Workshop on
Distributed Estimation and Control in Networked Systems
September 14-15, 2012. Santa Barbara, CA, USA

A comparison between upper bounds on performance of two consensus-based distributed optimization algorithms ⋆

Ion Matei [*] John S. Baras [**]

[*] *Institute for Research in Electronics and Applied Physics at University of Maryland, College Park, 20742, USA (*`imatei@umd.edu`*)*
[**] *Institute for Systems Research at University of Maryland, College Park, 20742, USA (*`baras@umd.edu`*)*

Abstract: In this paper we address the problem of multi-agent optimization for convex functions expressible as sums of convex functions. Each agent has access to only one function in the sum and can use only local information to update its current estimate of the optimal solution. We consider two consensus-based iterative algorithms, based on a combination between a consensus step and a subgradient decent update. The main difference between the two algorithms is the order in which the consensus-step and the subgradient descent update are performed. We obtain upper bounds on performance metrics of the two algorithms. We show that updating first the current estimate in the direction of a subgradient and then executing the consensus step ensures a tighter upper bound compared with the case where the steps are executed in reversed order. In support of our analytical results, we give some numerical simulations of the algorithms as well.

Keywords: distributed optimization, consensus, performance analysis.

1. INTRODUCTION

Multi-agent optimization problems appear naturally in many distributed processing problems (such as network resource allocation, collaborative control and estimation, etc.), where the optimization cost is a convex function which is not necessarily separable. A distributed subgradient method for multi-agent optimization of a sum of convex functions was proposed in Nedic and Ozdalgar (2009), where each agent has only local knowledge of the optimization cost, that is, it knows only one term of the sum. The agents exchange information subject to a communication topology, modeled as a graph; graph that defines the communication neighborhoods of the agents. The agents maintain *estimates* of the optimal decision vector, which are updated in two steps. In the first step, called so forth, consensus-step, an agent executes a convex combination between its current estimate and the estimates received from its neighbors. In the second step, the result of the consensus step is updated in the direction of a subgradient of the local knowledge of the optimization cost.

The consensus step is introduced to deal with the fact that the agents have incomplete knowledge about the optimization problem. Consensus problems received a lot of attention in recent years thanks to their usefulness in modeling many phenomena involving information exchange between agents, such as cooperative control of vehicles, formation control, flocking, synchronization, parallel computing, etc. Distributed computation over networks has a long history in control theory starting with the work of Borkar and Varaiya (1982), and Tsitsiklis (1984), Tsitsiklis et al. (1986) on asynchronous agreement problems and parallel computing. A theoretical framework for solving consensus problems was introduced in Saber and Murray (2004), while Jadbabaie et al. (2004) studied alignment problems for reaching an agreement. Relevant extensions of the consensus problem were done by Ren and Beard (2005), Moreau (2005) or, more recently, by Nedic and Ozdaglar (2010). The analysis of consensus problems was extended to the case where the communication topology is random, with relevant results being found in Salehi and Jadbabaie (2010), Hatano and Mesbahi (2005), Porfiri and Stilwell (2007), or Matei et al. (2008).

A different version of a consensus-based distributed optimization algorithm was proposed in Johansson et al. (2008). In this version, in the first step the current estimate is updated in the direction of a subgradient of the local knowledge of the optimization cost. In the second step a consensus-step is executed. Performance analysis of the aforementioned algorithms and extensions to the case where the communication topologies are random were addressed in Duchi et al. (2010), Matei and Baras (2010), Lobel and Ozdalgar (2008).

In this paper we compute upper bounds on performance metrics of the two algorithms and compare them. We use two performance metrics: the first metric looks at how close the cost function evaluated at the estimates gets to the optimal value; the second metric looks at the distance between the estimates and the minimizer. We obtain error bounds for the two metrics and rate of convergence for the second metric. The results of our analysis show that, under a constant step-size multiplying the subgradient, the second version of the algorithm ensures

⋆ This material is based upon work supported by the NIST-ARRA Measurement Science and Engineering Fellowship Program award 70NANB10H026, through the University of Maryland, by the US Air Force Office of Scientific Research MURI grant FA9550-09-1-0538, by the National Science Foundation (NSF) grant CNS-1035655, and by the Defense Advanced Research Projects Agency (DARPA) and the SRC FCRP grant SA00007007.

a tighter upper bound with respect to accuracy, compared to the first version of the algorithm. The rates of convergence of the second metric however are the same. The paper presents proofs only for the main results, while the proofs of the stated preliminary results can be found at Matei and Baras (2012).

Notations: Let X be a subset of $\mathbb{R}^n$ and let y be a point in $\mathbb{R}^n$. By slight abuse of notation, let $\|y-X\|$ denote the distance from the point y to the set X, i.e., $\|y-X\| \triangleq \inf_{x\in X}\|y-x\|$, where $\|\cdot\|$ is the standard Euclidean norm. For a twice differentiable function $f(x)$, we denote by $\nabla f(x)$ and $\nabla^2 f(x)$ the gradient and Hessian of f at x, respectively. Given a symmetric matrix A, by $(A \succeq 0)$ $A \succ 0$ we understand A is positive (semi) definite. The symbol $\otimes$ represents the Kronecker product.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function. We denote by $\partial f(x)$ the subdifferential of f at x, that is, the set of all subgradients of f at x:

$$\partial f(x) = \{d \in \mathbb{R}^n | f(y) \geq f(x) + d'(y-x),\ \forall y \in \mathbb{R}^n\}. \quad (1)$$

Let $\epsilon \geq 0$ be a nonnegative real number. We denote by $\partial_\epsilon f(x)$ the ϵ-subdifferential of f at x, that is, the set of all ϵ-subgradients of f at x:

$$\partial_\epsilon f(x) = \{d \in \mathbb{R}^n | f(y) \geq f(x) + d'(y-x) - \epsilon,\ \forall y \in \mathbb{R}^n\}. \quad (2)$$

The gradient of the differentiable function $f(x)$ on $\mathbb{R}^n$ satisfies a *Lipschitz condition with constant L* if

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x-y\|,\ \forall x, y \in \mathbb{R}^n.$$

The differentiable, convex function $f(x)$ on $\mathbb{R}^n$ is *strongly convex with constant l* if

$$f(y) \geq f(x) + \nabla f(x)'(y-x) + \frac{l}{2}\|y-x\|^2,\ \forall x, y \in \mathbb{R}^n.$$

We denote by LEM and SLEM the largest and second largest eigenvalue (in modulus) of a matrix, respectively.

2. PROBLEM FORMULATION

In this section we describe the communication model and the optimization model used throughout the paper.

2.1 Communication model

We consider a network of N agents, indexed by $i = 1, \ldots, N$. The communication topology is modeled by a graph $G = (V, \mathcal{E})$, where V is the set of N vertices (nodes) and $\mathcal{E} = (e_{ij})$ is the set of edges. The edges in the set $\mathcal{E}$ correspond to communication links between agents.

Assumption 1. The graph $G = (V, \mathcal{E})$ is undirected, connected and does not have self-loops.

Let G be a graph with N nodes and no self loops and let $A \in \mathbb{R}^{N\times N}$ be a row stochastic matrix, with positive diagonal entries. We say that the matrix A *corresponds* to the graph G, or the graph G is *induced* by A, if any non-zero entry (i,j) of A, with $i \neq j$, implies a link from j to i in G and vice-versa.

2.2 Optimization model

The goal of the N agents is to minimize a convex function $f : \mathbb{R}^n \to \mathbb{R}$. The function f is expressed as a sum of N functions, i.e.,

$$f(x) = \sum_{i=1}^{N} f_i(x), \quad (3)$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ are convex. Formally expressed, the agents' goal is to cooperatively solve the following optimization problem

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{N} f_i(x). \quad (4)$$

The fundamental assumption is that each agent i has access only to the function f_i.

Let f^* denote the optimal value of f and let X^* denote the set of optimizers of f, i.e., $X^* = \{x \in \mathbb{R}^n | f(x) = f^*\}$. Let $x_i(k) \in \mathbb{R}^n$ designate the *estimate of the optimal decision vector* of (4), maintained by agent i, at time k. The agents exchange estimates among themselves subject to the communication topology described by the graph G.

As mentioned in the introductory section, we consider two versions of a multi-agent subgradient optimization algorithm. The first version, referred henceforth as *Algorithm* 1, was introduced by Nedic and Ozdalgar (2009) and is given by

$$x_i^{(1)}(k+1) = \sum_{j=1}^{N} a_{ij} x_j^{(1)}(k) - \alpha(k) d_i^{(1)}(k), \quad (5)$$

where a_{ij} is the $(i,j)^{th}$ entry of a symmetric, row stochastic matrix A, corresponding to the undirected communication graph G. The real valued scalar $\alpha(k)$ is the stepsize, while the vector $d_i^{(1)}(k) \in \mathbb{R}^n$ is a subgradient of f_i at $x_i^{(1)}(k)$, that is, $d_i^{(1)}(k) \in \partial f_i(x_i^{(1)}(k))$. Obviously, when $f_i(x)$ are assumed differentiable, $d_i^{(1)}(k)$ becomes the gradient of f_i at $x_i^{(1)}(k)$, that is, $d_i^{(1)}(k) = \nabla f_i(x_i^{(1)}(k))$.

The second version of the algorithm, referred henceforth as *Algorithm* 2, was introduced by Johansson et al. (2008), and is expressed as

$$x_i^{(2)}(k+1) = \sum_{j=1}^{N} a_{ij}\left[x_j^{(2)}(k) - \alpha(k) d_j^{(2)}(k)\right], \quad (6)$$

where $d_j^{(2)}(k)$ is the subgradient of f_j at $x_j^{(2)}(k)$, and the rest of the parameters of the algorithm are the same as in *Algorithm* 1.

In what follows we assume that the step size is constant, that is, $\alpha(k) = \alpha$, for all $k \geq 0$. Note that we use superscripts to differentiate between the estimates of the two algorithms. In addition, we note that the main difference consists of the order the two steps of the algorithms are executed. In Algorithm 1, first the consensus-step is executed followed by an update in the direction of a subgradient. In Algorithm 2, the estimate is first updated in the direction of a subgradient of the local cost function, and the result is shared with the neighboring agents; agents that use these intermediate updates to generate new updates at the next time-step, by executing a consensus step.

The following assumptions, which will not necessarily be used simultaneously, introduce restrictions on the cost function $f(x)$ considered in this paper.

Assumption 2. (*Non-differentiable functions*)

(a) The subgradients of the functions $f_i(x)$ are uniformly bounded, that is, there exists a positive scalar φ such that

$$\|d\| \leq \varphi, \forall d \in \partial f_i(x),\ \forall x \in \mathbb{R}^n,\ \ i = 1, \ldots, N,$$

(b) The optimal solution set X^* is nonempty.

Assumption 3. (*Differentiable functions*)

(a) The functions $f_i(x)$ are twice continuously differentiable on $\mathbb{R}^n$,

(b) There exist positive scalars l_i, L_i such that
$$l_i I \leq \nabla^2 f_i(x) \leq L_i I,\ \forall x \in \mathbb{R}^n \text{ and } \forall i,$$

(c) The step-size α satisfies the inequality
$$0 < \alpha < \min\left\{\frac{1}{l}, \frac{1+\underline{\lambda}}{L}\right\},$$
where $\underline{\lambda}$ is the smallest eigenvalue of A, $L = \max_i L_i$ and $l = \min_i l_i$.

If Assumption 3 -(a) holds, Assumption 3 -(b) is satisfied if the gradient of $f_i(x)$ satisfies a Lipschitz condition with constant L_i and if $f_i(x)$ is strongly convex with constant l_i. Also, under Assumptions 3, X^* has one element which is the unique minimizer of $f(x)$, denoted henceforth by x^*.

2.3 Performance metrics

We analyze these algorithms with respect to two performance metrics. First, we look at how close the cost function evaluated at the estimates gets to the optimal value f^*. Let $\bar{f}_i^{best,(a)}(k) = \min_{s=0...k} f(x_i^{(a)}(s))$ be the smallest cost value achieved by agent i at iteration k. The first metric is given by
$$\bar{f}_i^{best,a}(k) - f^*. \tag{7}$$
The second metric looks at how close the estimates computed by the agents get to the optimal value, and we can formally expressed this as
$$\|x_i^{(a)}(k) - X^*\|, \tag{8}$$
where X^* is the set of minimizers of f. In the above, the scalar index $a \in \{1,2\}$, differentiates between the two optimization algorithms. For both algorithms, our goal is to obtain upper bounds for these performance metrics and compare them.

3. PRELIMINARY RESULTS

In this section we lay the foundation for our main results. The preliminary results introduced here revolve around the idea of providing upper-bounds on a number of quantities of interest. The first quantity is represented by the distance between the estimate of the optimal decision vector and the average of all estimates. The second quantity is described by the distance between the average of all estimates and the minimizer. We introduce the *average* vector of estimates of the optimal decision vector, denoted by $\bar{x}^a(k)$ and defined by
$$\bar{x}^{(a)}(k) \triangleq \frac{1}{N}\sum_{i=1}^{N} x_i^{(a)}(k). \tag{9}$$
The dynamic equation for the average vector can be derived from (5) and (6) and takes the form
$$\bar{x}^{(a)}(k+1) = \bar{x}^{(a)}(k) - \frac{\alpha}{N}h(k), \tag{10}$$
where $h(k) = \sum_{i=1}^{N} d_i(k)$ and $a \in \{1,2\}$. We introduce also the *deviation* of the local estimates $x_i^{(a)}(k)$ from the average estimate $\bar{x}^{(a)}(k)$, which is denoted by $z_i^{(a)}(k)$ and is defined by
$$z_i^{(a)}(k) \triangleq x_i^{(a)}(k) - \bar{x}^{(a)}(k),\ i = 1 \ldots N, \tag{11}$$
and let β be a positive scalar such that
$$\|z_i^{(a)}(0)\| \leq \beta,\ i = 1 \ldots N.$$
Let us define the *aggregate* vectors of estimates, average estimates, deviations and (sub)gradients, respectively:
$$\mathbf{x}^{(a)}(k)' \triangleq [x_1^{(a)}(k)', x_2^{(a)}(k)', \ldots, x_N^{(a)}(k)'] \in \mathbb{R}^{Nn},$$
$$\bar{\mathbf{x}}^{(a)}(k)' \triangleq [\bar{x}^{(a)}(k)', \bar{x}^{(a)}(k)', \ldots, \bar{x}^{(a)}(k)'] \in \mathbb{R}^{Nn},$$
$$\mathbf{z}^{(a)}(k)' \triangleq [z_1^{(a)}(k)', z_2^{(a)}(k)', \ldots, z_N^{(a)}(k)'] \in \mathbb{R}^{Nn}$$
and
$$\mathbf{d}^{(a)}(k)' \triangleq [d_1^{(a)}(k)', d_2^{(a)}(k)', \ldots, d_N^{(a)}(k)'] \in \mathbb{R}^{Nn}.$$
From (9) we note that the aggregate vector of average estimates can be expressed as
$$\bar{\mathbf{x}}^{(a)}(k) = \mathbf{J}\mathbf{x}^{(a)}(k),$$
where $\mathbf{J} = \frac{1}{N}\mathbb{1}\mathbb{1}' \otimes I$, with I the identity matrix in $\mathbb{R}^{n\times n}$ and $\mathbb{1}$ the vector of all ones in $\mathbb{R}^N$. Consequently, the aggregate vector of deviations can be written as
$$\mathbf{z}^{(a)}(k) = (\mathbf{I} - \mathbf{J})\mathbf{x}^{(a)}(k), \tag{12}$$
where $\mathbf{I}$ is the identity matrix in $\mathbb{R}^{nN\times nN}$.

Let us define the matrices $\mathbf{A} \triangleq A \otimes I$ and $\mathbf{W} \triangleq \mathbf{A} - \mathbf{J}$ and let λ be the SLEM of A. By Assumption 1, $\lambda < 1$. In addition, it is not difficult to observe that λ is the SLEM of $\mathbf{A}$ and the LEM of W and $\mathbf{W}$.

In the next lemma we show that, under Assumption 3, for small enough α (the step-size) the gradients $\nabla f_i(x_i^{(a)}(k))$ remain bounded for all k, for both optimization algorithms.

Lemma 4. Let Assumption 3 hold and let $\mathcal{F} : \mathbb{R}^{Nn} \to \mathbb{R}$ be a function given by $\mathcal{F}(\mathbf{x}) = \sum_{i=1}^{N} f_i(x_i)$ where $\mathbf{x}' = (x_1', \ldots, x_N')$. There exists a positive scalar φ such that
$$\|\nabla f_i(x_i^{(a)}(k))\| \leq \varphi,$$
$$\|\nabla f_i(\bar{x}^{(a)}(k))\| \leq \varphi,$$
for all k and i, where $\varphi = 3L(\|\mathbf{x}(0) - \tilde{\mathbf{x}}\| + \|\tilde{\mathbf{x}}\|)$, $L = \max_i L_i$, $\tilde{\mathbf{x}}$ is the unique minimizer of $\mathcal{F}(\mathbf{x})$, $x_i^{(1)}(k)$ and $x_i^{(2)}(k)$ satisfy (5) and (6), respectively, and $\bar{x}^{(a)}(k)$ satisfies (10).

Remark 5. Throughout the rest of the paper, we are going to use φ to denote the upper bound on the subgradients of $f(x)$ (given by Assumption 2) or on the gradients of $f(x)$ (given by Assumption 3 and Lemma 4), when these quantities are computed at values given by the two distributed optimization algorithms discussed above.

The next Proposition characterizes the dynamics of the vector $\mathbf{z}^{(a)}(k)$.

Proposition 6. Let Assumptions 1 and 2 or 1 and 3 hold. Then the dynamic evolution of the aggregate vector of deviations in the case of the two optimization algorithms is given by
$$\mathbf{z}^{(a)}(k+1) = \mathbf{W}\mathbf{z}^{(a)}(k) - \alpha\mathbf{H}^{(a)}\mathbf{d}^{(a)}(k),\ \mathbf{z}^{(a)}(0) = \mathbf{z}_0, \tag{13}$$
where
$$\mathbf{H}^{(a)} = \begin{cases} \mathbf{I} - \mathbf{J}, & \text{if } a = 1, \\ \mathbf{W}, & \text{if } a = 2, \end{cases}$$
with norm upper-bound
$$\|\mathbf{z}^{(a)}(k)\| \leq \lambda^k \beta\sqrt{N} + \alpha\varphi\sqrt{N}\psi^{(a)}(\lambda), \tag{14}$$
where λ is the SLEM of A and
$$\psi^{(a)}(\lambda) = \begin{cases} \dfrac{1}{1-\lambda}, & \text{if } a = 1, \\ \dfrac{\lambda}{1-\lambda}, & \text{if } a = 2. \end{cases} \tag{15}$$

The following lemma allows us to interpret $d_i^{(a)}(k)$ as an ϵ-subgradient of f_i at $\bar{x}^{(a)}(k)$.

Lemma 7. Let Assumptions 2 or 3 hold. Then the vector $d_i^{(a)}(k)$ is an $\epsilon^{(a)}(k)$-subdifferential of f_i at $\bar{x}^{(a)}(k)$, i.e., $d_i^{(a)}(k) \in \partial_{\epsilon^{(a)}(k)} f_i(\bar{x}^{(a)}(k))$ and $h^{(a)}(k) = \sum_{i=1}^N d_i^{(a)}(k)$ is an $N\epsilon^{(a)}(k)$ - subdifferential of f at $\bar{x}^{(a)}(k)$, i.e., $h^{(a)}(k) \in \partial_{N\epsilon^{(a)}(k)} f(\bar{x}^{(a)}(k))$, for any $k \geq 0$, where $\epsilon^{(a)}(k) = 2\varphi\|\mathbf{z}^{(a)}(k)\|$.

For twice differentiable cost functions with lower and upper bounded Hessians, the next result gives an upper bound on the distance between the average vector $\bar{x}^{(a)}(k)$ and the minimizer of f.

Lemma 8. Let Assumptions 1 and 3 hold and let $\{\bar{x}^{(a)}(k)\}_{k\geq 0}$ be a sequence of vectors defined by iteration (10). Then, the following inequality holds

$$\|\bar{x}^{(a)}(k) - x^*\|^2 \leq \|\bar{x}(0) - x^*\|^2\gamma^k + 4\alpha\varphi\beta\sqrt{N}\frac{\gamma^k - \lambda^k}{\gamma - \lambda} + \\ + \frac{\alpha^2\varphi^2}{1-\gamma}\left(4\sqrt{N}\psi^{(a)}(\lambda) + 1\right), \tag{16}$$

where $\psi^{(a)}(\lambda)$ is defined in (15), $\gamma = 1 - \alpha l$, with $l = \min_i l_i$.

4. MAIN RESULTS - ERROR BOUNDS

In this section we derive upper bounds for the two performance metrics introduced in the Problem Formulation section, for the two distributed optimization algorithms. First, we give a bound on the difference between the best recorded value of the cost function f, evaluated at the estimate $x_i^{(a)}(k)$, and the optimal value f^*. Second, we focus on the distance between the estimate $x_i^{(a)}(k)$ and the minimizer of f^*. For a particular class of twice differentiable functions, we give an upper bound on this metric and show how fast the time varying part of this bound converges to zero. The purpose of this section is to show the difference in upper bounds on performance between the two algorithms.

The following result shows how close the cost function f evaluated at the estimate $x_i^{(a)}(k)$ gets to the optimal value f^*. A similar result for the standard subgradient method can be found in Nedic and Bertsekas (2001), for example.

Theorem 9. Let Assumptions 1 and 2 or 1 and 3 hold and let $\{x_i^{(a)}(k)\}_{k\geq 0}$ be a sequence generated by the two distributed optimization algorithms, where $a \in \{1,2\}$. Let $f_i^{best,(a)}(k) = \min_{s=0...k} f(x_i^{(a)}(s))$ be the smallest cost value achieved by agent i, at iteration k. Then

$$\lim_{k\to\infty} f_i^{best,(a)}(k) \leq f^* + 3\alpha\varphi^2 N\sqrt{N}\psi^{(a)}(\lambda) + \frac{N\alpha\varphi^2}{2}, \tag{17}$$

where $\psi^{(a)}(\lambda)$ is defined in (15).

Remark 10. The previous result shows that the asymptotic error bound of the first metric decreases with both α (the algorithm step-size) and λ (the connective measure). In addition, it emphasizes the difference in upper bounds on performance from the first metric perspective, in the case of the two optimization algorithms. We note that the error bound in the case of *Algorithm* 2 is improved (diminished) by a factor of $3\alpha\varphi^2 N\sqrt{N}$, compared to *Algorithm* 1.

In the case of twice differentiable functions, the next result introduces an error bound for the second metric, in the case of the two optimization algorithms. We essentially show that the estimates produced by the two algorithms "converge to within some guaranteed distance" from the optimal point, distance which can be made arbitrarily small by decreasing the stepsize α. In addition, the time varying component of the error bounds converges to zero at least linearly.

Theorem 11. Let Assumptions 1 and 3 hold. Then, for the sequence $\{x_i^{(a)}(k)\}_{k\geq 0}$ generated by iteration (5) we have

$$\limsup_{k\to\infty} \|x_i^{(a)}(k) - x^*\|^2] \leq \alpha\varphi\Big[\sqrt{N}\psi^{(a)}(\lambda) + \\ + \sqrt{\frac{4\sqrt{N}\psi^{(a)}(\lambda) + 1}{1-\gamma}}\Big], \tag{18}$$

and

$$\|x_i^{(a)}(k) - x^*\|^2 \leq \zeta_1\eta(\lambda,\gamma)^k + \zeta_2^{(a)}, \tag{19}$$

where $\psi^{(a)}(\lambda)$ is defined in (15), ζ_1 and $\zeta_2^{(a)}$ are positive constants depending on the initial conditions, and the parameters of the algorithms, and

$$\eta(\lambda,\gamma) = \begin{cases} \lambda, & \lambda \geq \gamma, \\ \sqrt{\gamma}, & \lambda < \gamma, \end{cases} \tag{20}$$

where $\gamma = 1 - \alpha l$, with $l = \min_i l_i$.

Remark 12. The previous result shows that the algorithms ensure convergence of the estimates within some distance of the optimal solution; distance that depends on the parameters of the problem, and on the connectivity of the network, parameterized by λ. This distance decreases with α and λ. However, as in the case of the standard subgradient algorithm, decreasing α induces a slower convergence rate. We also note that as long $\alpha < \frac{1-\lambda}{l}$, according to our analysis the rate of convergence (of the error bound) is dictated by γ. As in the case of the first metric, Algorithm 2 ensures a tighter upper bound with respect to precision, since the aforementioned distance is smaller compared to Algorithm 1. However, the error bounds rate of convergence in the case of the two algorithms are the same.

Remark 13. As pointed out earlier, our results show that in a worse-case scenario, the second algorithm outperforms the first algorithm. In Tu and Sayed (2012) the authors analyzed the performance of the two distributed optimization algorithms, as well. They managed to show that indeed the second algorithm performs better compared to the first algorithm. However, they assumed that the objective function is quadratic, and as a consequence the (sub)gradient is a linear function. Therefore, the optimization algorithms become linear and an exact analysis of the performance is possible. In our case, since we assume more general objective functions, an exact performance analysis has not been obtained to-date, and may not be possible.

5. NUMERICAL EXAMPLE

Our analytical results indicate that in worse case scenarios, *Algorithm* 2 performs better than *Algorithm* 1, at least from the accuracy stand point. However, we cannot claim that indeed *Algorithm* 2 performs better than *Algorithm* 1, since no lower bounds are given. In this section, we are going to test on a specific example, that the intuition provided by the analytical results is verified in numerical simulations. To this end we consider a network of ten agents organized in a star topology, where node 1 is connected to the rest of the nine nodes (Figure

1). The collaboration matrix A is chosen as $A = I + 0.101Lp$, where Lp is the Laplacian of the undirected graph shown in Figure 1. The smallest eigenvalue of A is given by $\underline{\lambda} = -0.0101$, while the SLEM of A is $\lambda = 0.8990$.

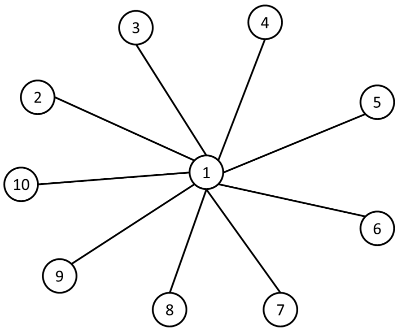

Fig. 1. Star network topology with ten nodes

The function to be collaboratively minimized is given by $f(x) = \sum_{i=1}^{10} f_i(x_1, x_2)$, where

$$f_i(x_1, x_2) = \frac{1}{i^2}x_1^2 + \frac{1}{i}x_2^2 - \frac{1}{i}x_1 - \frac{1}{i^2}x_2.$$

We note that f_i's are convex, twice continuously differentiable and

$$\frac{2}{i^2}I \leq \nabla^2 f_i(x_1, x_2) \leq \frac{2}{i}I,\ i \in \{1, \ldots, 10\}.$$

Therefore, $L = \max_i L_i = 2$, $l = \min_i l_i = 0.02$ and by choosing

$$\alpha \in \left(0, \min\left\{\frac{1}{l}, \frac{1+\underline{\lambda}}{L}\right\}\right) = (0,\ 0.495),$$

we satisfy Assumption 3. The function $f(x_1, x_2)$ admits a unique minimizer given by $x_1^* = 0.2645$ and $x_2^* = 0.9450$. Figures 2, 3 and 4 show the evolution of the second performance metric for the two algorithms, for different values of α, as the algorithms iterate. We note that the intuition given by our analysis is verified by the numerical simulations, since in all cases Algorithm 2 performs better than Algorithm 1. In addition, we observe that as expected, as we decrease α the accuracy of the two algorithms improve, but at a cost of decreased rate of convergence.

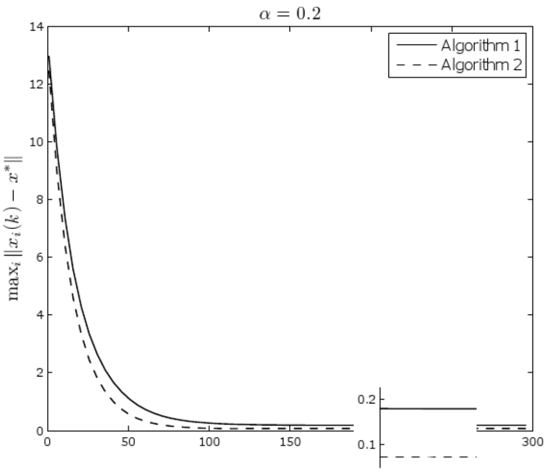

Fig. 2. Decay of $\max_i \|x_i(k) - x^*\|$ for $\alpha = 0.2$

Tables 1 and 2 summarize the asymptotic behavior of the two performance metrics, in the case of the graph presented in Figure 1. As shown analytically, for both performance metrics, Algorithm 2 fairs better than Algorithm 1.

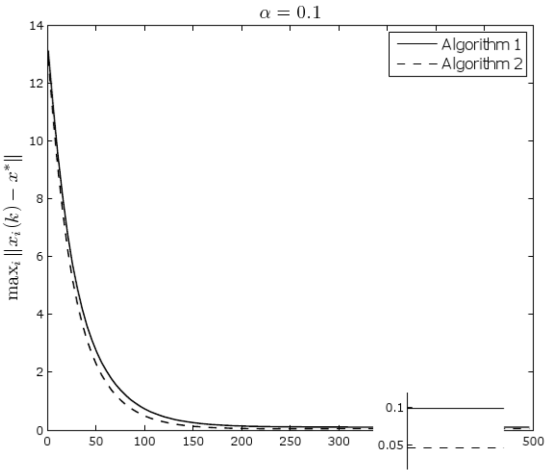

Fig. 3. Decay of $\max_i \|x_i(k) - x^*\|$ for $\alpha = 0.1$

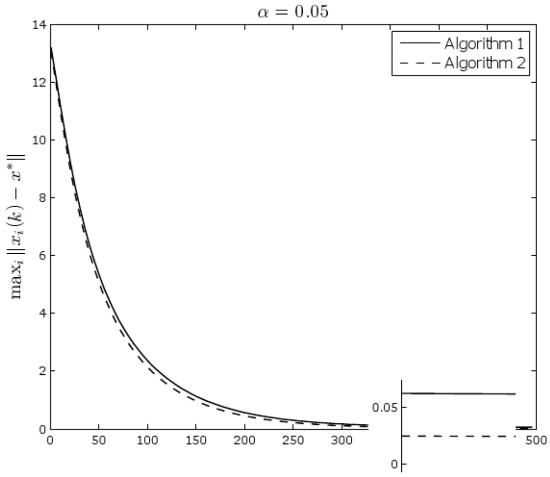

Fig. 4. Decay of $\max_i \|x_i(k) - x^*\|$ for $\alpha = 0.05$

$\limsup_{k\to\infty} \max_i \|x_i(k) - x^*\|$ for Figure 1 graph		
α	Algorithm 1	Algorithm 2
0.2	0.1785	0.0701
0.1	0.0980	0.0454
0.05	0.0517	0.0266

Table 1.

$\lim_{k\to\infty} \max_i f(x_i(k)) - f^*$ for Figure 1 graph		
α	Algorithm 1	Algorithm 2
0.2	52.1×10^{-3}	8.5×10^{-3}
0.1	15.9×10^{-3}	3.4×10^{-3}
0.05	4.48×10^{-3}	1.27×10^{-3}

Table 2.

We repeated the numerical simulations for a graph with improved connectivity; graph shown in Figure 5. As before, we choose as collaboration matrix $A = I + 0.101Lp$, where Lp is the Laplacian of the undirected graph shown in Figure 5. It can be checked that the smallest eigenvalue in this case is $\underline{\lambda} = -0.0101$ and the SLEM is given by $\lambda = 0.8868$, which shows the improved connectivity.

The asymptotic behavior of the two metrics for the new graph is shown in Tables 3 and 4. As proved by our analysis, improved

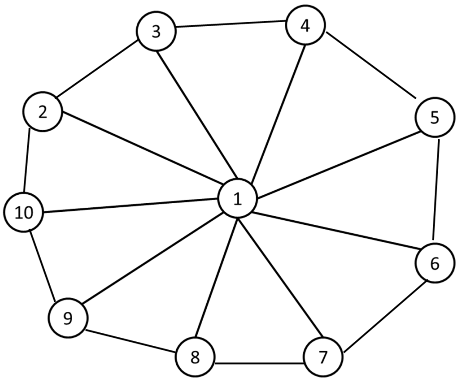

Fig. 5. Ten nodes graph with improved connectivity

connectivity improves the performance metrics for the two optimization algorithms, phenomenon observed in numerical simulations as well.

$\limsup_{k\to\infty}\max_i \|x_i(k)-x^*\|$ for Figure 5 graph		
α	Algorithm 1	Algorithm 2
0.2	0.1634	0.0369
0.1	0.0860	0.0209
0.05	0.0504	0.0138

Table 3.

$\lim_{k\to\infty}\max_i f(x_i(k))-f^*$ for Figure 5 graph		
α	Algorithm 1	Algorithm 2
0.2	43.8×10^{-3}	2.4×10^{-3}
0.1	12.3×10^{-3}	7.81×10^{-4}
0.05	4.2×10^{-3}	3.19×10^{-4}

Table 4.

6. CONCLUSIONS

In this paper we addressed two consensus-based distributed subgradients algorithms. The main difference between these algorithms is the order in which the consensus step is executed. In the first algorithm first a consensus step is executed, followed by an update in the direction of a subgradient. In the case of the second algorithm, the order is reversed. Under a set of assumptions on the objective function, we gave upper bounds on the accuracy of the algorithms and on the rate of convergence. We showed that in the case of the second algorithm, the upper bound on the accuracy is tighter. This suggests that in a worse-case scenario, the second algorithm performs better. A similar effect was not observed in the case of the rate of convergence. In addition, we presented numerical simulations of the two algorithms that confirm the superiority of the second algorithm.

REFERENCES

Borkar, V. and Varaiya, P. (1982). Asymptotic agreement in distributed estimation. *IEEE Trans. Autom. Control*, AC-27(3), 650–655.

Duchi, J., Agarwal, A., and Wainwright, M. (2010). Distributed dual averaging in netwroks. *Proceedings of the 2010 Neural Information System Foundation Conference*.

Hatano, Y. and Mesbahi, M. (2005). Agreement over random networks. *IEEE Trans. Autom. Control*, 50(11), 1867–1872.

Jadbabaie, A., Lin, J., and Morse, A. (2004). Coordination of groups of mobile autonomous agents using nearest neighbor. *IEEE Trans. Autom. Control*, 48(6), 998–1001.

Johansson, B., Keviczky, T., Johansson, M., and Johansson, K. (2008). Subgradient methods and consensus algorithms for solving convex optimization problems. *Proceedings of the 47th IEEE Conference on Decision and Control*, 4185–4190.

Lobel, I. and Ozdalgar, A. (2008). Convergence analysis of distributed subgradient methods over random networks. *Proceedings of* 46^{th} *Allerton Conference on Communication, Control, and Computing*, 353–360.

Matei, I. and Baras, J. (2010). Performance evaluation of the consensus-based distributed subgradient method under random communication topologies. *Proceedings of the 19th International Symposium of Mathematical Theory of Networks and Systems*, 1247–1254.

Matei, I. and Baras, J. (2012). A performance comparison between two consensus-based distributed optimization algorithms. *ISR Technical Report*. URL `http://hdl.handle.net/1903/12480`.

Matei, I., Martins, N., and Baras, J. (2008). Almost sure convergence to consensus in markovian random graphs. *Proceedings of the 47th IEEE Conference on Decision and Control*, 3535–3540.

Moreau, L. (2005). Stability of multi-agents systems with time-dependent communication links. *IEEE Trans. Autom. Control*, 50(2), 169–182.

Nedic, A. and Bertsekas, D. (2001). Incremental subgradient methods for nondifferential optimization. *SIAM Journal on Optimization*, 12(1), 109–138.

Nedic, A. and Ozdaglar, A. (2010). Convergence rate for consensus with delays. *Journal of Global Optimization*, 47(3), 437–456.

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

Porfiri, M. and Stilwell, D. (2007). Consensus seeking over random directed weighted graphs. *IEEE Trans. Autom. Control*, 52(9), 1767–1773.

Ren, W. and Beard, R. (2005). Consensus seeking in multi-agents systems under dynamically changing interaction topologies. *IEEE Trans. Autom. Control*, 50(5), 655–661.

Saber, R. and Murray, R. (2004). Consensus problem in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control*, 49(9), 1520–1533.

Salehi, A.T. and Jadbabaie, A. (2010). Consensus over ergodic stationary graph processes. *IEEE Trans. Autom. Control*, 55(1), 225–230.

Tsitsiklis, J. (1984). Problems in decentralized decision making and computation. *Ph.D. dissertation, Dept. Electr. Eng.*

Tsitsiklis, J., Bertsekas, D., and Athans, M. (1986). Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Trans. Autom. Control*, 31(9), 803–812.

Tu, S. and Sayed, A. (2012). Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. *arXiv archive*. URL `http://arxiv.org/abs/1205.3993v1`.