Presented as late breaking results contribution
21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Continuous-Time Accelerated Algorithm for Distributed Optimization over Undirected Graph

Ismi Rosyiana Fitri, Jung-Su Kim*

Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul 01811, Korea e-mail: jungsu@seoultech.ac.kr.

Abstract: This paper investigates a distributed optimization problem for a group of agents where the objective function is the sum of convex local functions associated with the individual agents on connected and undirected graph topologies. Inspired by the discrete-time accelerated gradient algorithm for solving a centralized optimization problem, this paper designs a continuous-time algorithm for solving the distributed optimization problem. By using a standard Lyapunov analysis, it is proved that the proposed method converges exponentially to the optimal solution when the local costs are strongly convex with a local Lipschitz gradient. Furthermore, it is also shown that faster convergence could be easily achieved by increasing one design parameter. Numerical simulations illustrate the result.

Keywords: consensus, continuous-time, convex, distributed optimization

1. INTRODUCTION

Distributed optimization is gaining more popularity due to its variety of applications such as the economic dispatch problem in power systems, utilization maximization in communication networks, and robust estimation in wireless sensor networks, e.g., Rabbat and Nowak (2004); Ram et al. (2010); Cherukuri and Cortes (2015). The majority of existing algorithms are written in the discrete-time setting, e.g., Johansson et al. (2010); Nedic and Ozdaglar (2009); Boyd et al. (2011). Meanwhile, the continuous-time strategies are also studied over the past few years, see the study by Gharesifard and Cortes (2014), Kia et al. (2015), and Li et al. (2018). In this paper, we are interested in investigating more about a continuous-time coordination algorithm for distributed optimization.

Most of the algorithms on distributed optimization with the continuous-time setting fall under the class of first-order optimizers. For instance, in Gharesifard and Cortes (2014), the original unconstrained distributed optimization problem is viewed as a constrained one due to the consensus requirement. With the constrained problem in mind, a saddle point algorithm is used to find the optimal solution. Furthermore, it is extended by Kia et al. (2015) to reduce the number of exchanged variables between the two agents. The other algorithms, such as in Li et al. (2018), can also be seen as the corresponding gradient descent in a distributed optimization setting. Although the performance of gradient descent optimizers is generally effective, it is highly dependent on the curvature of the optimization objective. It is well known within the optimization community that the convergence rate of the gradient descent algorithm can become slow if the objective function exhibits pathological curvature, Song et al. (2018). In order to improve the convergence property, Newton's method, such as in Varagnolo et al. (2016), can be adopted since not only the first-order derivative but also the second-order one is taken into account. However, the algorithm is computationally demanding since it is required to know the Hessian of the objective function. In the discrete-time setting, a method called momentum gradient algorithm proposed by Polyak (1964) is popularly used, especially in neural network literature, for solving a centralized optimization problem with faster convergence. This algorithm works by considering the derivative of the current and previous steps in every iteration. Another method closely related to the momentum algorithm is the accelerated gradient by Nesterov (1983). For a long time, this algorithm is written in discrete-time. It is only recently that Wilson et al. (2016) and Wibisono et al. (2016) investigate the algorithm in a continuous-time setting such that the Nesterov gradient method can be recovered by a careful discretization of an ordinary differential equation. Inspired by this, a continuous-time algorithm for solving a distributed optimization problem is proposed in this paper.

The contribution of this paper is as follows. We first propose a novel continuous-time coordination algorithm for a distributed optimization. Then, we study the convergence property under a connected and undirected graph. By using the standard Lyapunov theory, it is proved that the algorithm converges exponentially fast to the optimal point. Moreover, compared to the existing algorithms, it is shown that the proposed method provides a simple way to improve, i.e. to accelerate, the convergence property of the algorithm. Finally, numerical simulations are used to validate the proposed method.

⋆ Sponsor and financial support acknowledgment goes here. Paper titles should be written in uppercase and lowercase letters, not all uppercase.

Presented as late breaking results contribution
21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

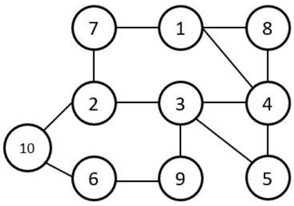

Fig. 1. Communication graph among agents

2. NOTATION AND PRELIMINARIES

Let $\mathbb{R}$, $\mathbb{R}^n$, and $\mathbb{R}_{>0}$ be the sets of real numbers, real vectors of dimension n, and positive real number, respectively. For vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$, $\mathbf{a} = (\mathbf{a}_1^\top, \ldots, \mathbf{a}_m^\top)^\top$ is the aggregate vector. The transpose of a matrix $\mathbf{A}$ is $\mathbf{A}^\top$. The identity matrix with dimension n is denoted by $\mathbf{I}_n$. We use $\mathbf{1}_n$ (resp $\mathbf{0}_n$) to represent a vector of n ones (resp. n zeros). For $\mathbf{A} \in \mathbb{R}^{n \times m}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$, $(\mathbf{A} \otimes \mathbf{B})$ denote their Kronecker product. For a vector $\mathbf{a} \in \mathbb{R}^n$, $\|\mathbf{a}\| = \sqrt{\mathbf{a}^\top \mathbf{a}}$ is the Euclidean norm of $\mathbf{a}$. For a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, ∇f is the gradient of f. A differentiable function f is μ-strongly convex over a convex set $C \subseteq \mathbb{R}^n$ iff there exists $\mu \in \mathbb{R}_{>0}$ such that $(\mathbf{x} - \mathbf{y})^\top (\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})) \geq \mu \|\mathbf{x} - \mathbf{y}\|^2$ for all $\mathbf{x}, \mathbf{y} \in C$ and $\mathbf{x} \neq \mathbf{y}$. A function f is locally Lipschitz at $\mathbf{x} \in \mathbb{R}^n$ iff there exist $M \in \mathbb{R}_{\geq 0}$ such that $\|f(\mathbf{x}) - f(\mathbf{y})\| \leq M \|\mathbf{x} - \mathbf{y}\|$, for $\mathbf{x}, \mathbf{y} \in \mathcal{W}$, where $\mathcal{W}$ denotes the neighboring set of $\mathbf{x}$. For a convex function f with M-Lipschitz gradient, $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|^2 \leq M(\mathbf{x} - \mathbf{y})^\top (\nabla f(\mathbf{x}) - \nabla f(\mathbf{y}))$.

An undirected graph $\mathcal{G}$ consists of a node set $\mathcal{V} = \{1, 2, \ldots, n\}$ and unordered edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ where self loop (i, i) is excluded. The set of neighbors of node i is denoted by $\mathcal{N}_i = \{j\}, j \in \mathcal{V}, (j, i) \in \mathcal{E}$. The graph $\mathcal{G}$ is connected if there exists a path between any pair of distinct nodes. A weighted adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ with $a_{ij} = a_{ji}$ and $a_{ij} > 0$ if only if $(i, j) \in \mathcal{E}$. The Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $D = diag(d_i)$, and $d_i = \sum_{j \in \mathcal{N}_i} a_{ij}$. For connected graph, the eigenvalues of $\mathbf{L}$ satisfy $\lambda_1 \leq \ldots \leq \lambda_n$, where $\lambda_1 = 0$.

3. PROBLEM STATEMENT AND PROPOSED ALGORITHM

3.1 Problem Formulation

Consider the following problem

$$\min_{\mathbf{x} \in \mathbb{R}^d} \mathsf{f}(\mathbf{x}) = \sum_{i=1}^{N} f_i(\mathbf{x}), \tag{1}$$

where i represents the index of an agent. Here, it is assumed that each agent has a convex local cost function $f_i : \mathbb{R}^d \to \mathbb{R}$. The objective of this paper is to design an algorithm in a continuous-time to solve this problem in a distributed manner. Furthermore, suppose that the following assumption is true.

Assumption 1. Each $f_i, i = 1, 2, \ldots, N$ is continuously differentiable, μ_i-strongly convex, and its gradient is M_i-Lipschitz on $\mathbb{R}^d$. The communication graph $\mathcal{G}$ is undirected and connected.

Owing to Assumption 1, the problem (1) can be rewritten as

$$\min_{\mathbf{x}_i, i=1,\ldots,N} f(\mathbf{x}) = \sum_{i=1}^{N} f_i(\mathbf{x}_i) \text{ subject to } (\mathbf{L} \otimes \mathbf{I}_d)\mathbf{x} = 0, \tag{2}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is the variable of the ith agent and $\mathbf{x} = [\mathbf{x}_1^\top, \ldots, \mathbf{x}_N^\top]^\top \in \mathbb{R}^{dN}$ is the network variable.

3.2 Proposed Distributed Algorithm via Continuous-Time

In this section, the proposed continuous-time distributed coordination algorithm to solve the problem (2) is stated. For $i \in \mathcal{V}$, the ith agent is given by

$$\begin{aligned}
\dot{\mathbf{x}}_i &= \eta(\mathbf{z}_i - \mathbf{x}_i), \\
\dot{\mathbf{z}}_i &= -\eta \nabla f_i(\mathbf{x}_i) - \eta\kappa \sum_{j=1}^{N} a_{ij}(\mathbf{z}_i - \mathbf{z}_j) - \eta \mathbf{v}_i, \\
\dot{\mathbf{v}}_i &= \eta\kappa \sum_{j=1}^{N} a_{ij}(\mathbf{z}_i - \mathbf{z}_j),
\end{aligned}$$

where $\mathbf{x}_i(t), \mathbf{z}_i(t), \mathbf{v}_i \in \mathbb{R}^d$, $\kappa \in \mathbb{R}$, and $\eta > 0$ is an arbitrary positive constant. In network variables, $\mathbf{x}, \mathbf{z}, \mathbf{v} \in \mathbb{R}^{dN}$, the algorithm reads as

$$\dot{\mathbf{x}} = \eta(\mathbf{z} - \mathbf{x}), \tag{3a}$$

$$\dot{\mathbf{z}} = -\eta \mathbf{h} - \eta\kappa(\mathbf{L} \otimes \mathbf{I}_d)\mathbf{z} - \eta \mathbf{v}, \tag{3b}$$

$$\dot{\mathbf{v}} = \eta\kappa(\mathbf{L} \otimes \mathbf{I}_d)\mathbf{z}, \tag{3c}$$

where $\mathbf{h} := \nabla f(\mathbf{x}) = [\nabla f_1^\top(\mathbf{x}_1), \ldots, \nabla f_N^\top(\mathbf{x}_N)]^\top$. Note that the algorithm is distributed since each agent only needs the local variables and the information of $\mathbf{z}_j$ from its neighbors. Furthermore, in contrast to the existing algorithms, the gradient $\nabla f(\mathbf{x})$ is considered in the dynamics of $\mathbf{z}$ instead of $\mathbf{x}$.

4. MAIN RESULT

Here, we study the convergence of the distributed optimization algorithm (3) over a connected and undirected gradient topology. In the following, it is shown that the equilibrium points are the optimal solution and consensus between all agent are achieved.

Lemma 1. Suppose that $\sum_{i=1}^{N} \mathbf{v}_i(0) = 0$ and Assumption 1 hold true, then the equilibrium of (3) is an optimal solution of the optimization problem (2).

Proof. The equilibrium point of (3), $(\mathbf{x}^*, \mathbf{z}^*, \mathbf{v}^*)$, satisfy the following equalities

$$\mathbf{0} = \eta(\mathbf{z}^* - \mathbf{x}^*), \tag{4a}$$

$$\mathbf{0} = -\eta \nabla f(\mathbf{x}^*) - \eta\kappa(\mathbf{L} \otimes \mathbf{I}_d)\mathbf{z}^* - \eta \mathbf{v}^*, \tag{4b}$$

$$\mathbf{0} = \eta\kappa(\mathbf{L} \otimes \mathbf{I}_d)\mathbf{z}^*. \tag{4c}$$

Since the graph is undirected and connected, we have $\mathbf{1}_N^\top \mathbf{L} = \mathbf{0}$. Thus, left multiplying (3c) by $\mathbf{1}_N^\top \otimes \mathbf{I}_d$ results in $\sum_{i=1}^{N} \dot{\mathbf{v}}_i = \mathbf{0}$. With the assumption that $\sum_{i=1}^{N} \mathbf{v}_i(0) = 0$, $\sum_{i=1}^{N} \mathbf{v}_i(t) = \mathbf{0}$ holds true. By using this and (4c), left multiplying (4b) by $\mathbf{1}_N^\top \otimes \mathbf{I}_d$ yields $\mathbf{0} = \sum_{i=1}^{N} \nabla f^i(\mathbf{x}^i)$. It implies that the optimality condition is achieved. Furthermore, according to (4c), $\mathbf{z}^*$ belongs to the null space of $\mathbf{L}$. For a connected graph, the null space of $\mathbf{L}$ is spanned by $\mathbf{1}_N$. Thus, $\mathbf{z}^* = \mathbf{1}_N \otimes \boldsymbol{\theta}$ where $\boldsymbol{\theta} \in \mathbb{R}^d$. Given (4a), the consensus on $\mathbf{x}^*$ is also attained, i.e. $\mathbf{x}^* = \mathbf{1}_N \otimes \boldsymbol{\theta}$.

With this Lemma 1 in mind, the equilibrium point $\mathbf{x}^*$ satisfies $(\mathbf{L} \otimes \mathbf{I}_d)\mathbf{x}^* = \mathbf{0}_{dN}$. The following result provides a condition on parameter κ to ensure the exponential convergence of (3) to the optimal solution $\mathbf{x}^* = \mathbf{1}_N \otimes \boldsymbol{\theta}$.

Theorem 2. Under Assumption 1, let λ_2 be the second smallest eigenvalue of $(\mathbf{L}+\mathbf{L}^\top)/2$, $\mu := \min\{\mu_1, \ldots, \mu_N\}$ and $M := \max\{M_1, \ldots, M_N\}$. Suppose that $\kappa, \phi > 0$ satisfy $\phi > 2M$ and

$$\xi =: 4\kappa\phi\lambda_2 - 5\phi^2 + 4\phi + 4\mu\phi - 8\mu M > 0, \quad (5a)$$

$$\gamma =: 4\kappa\phi\lambda_2 - 7\phi^2 - 4\phi - 9 > 0. \quad (5b)$$

Then, for each $i \in \mathcal{V}$, starting from $\mathbf{x}_i(0), \mathbf{z}_i(0), \mathbf{v}_i(0) \in \mathbb{R}^d$ with $\sum_{i=1}^N \mathbf{v}_i(0) = \mathbf{0}_d$, the trajectory of (3) exponentially converges to $\mathbf{x}^*$ with rate no less than

$$\eta \min\{\frac{\xi}{4}, \frac{\gamma}{4}, \frac{1}{5}\}/\lambda_{\mathbf{B}}, \quad (6)$$

where $\lambda_{\mathbf{B}}$ is the maximum eigenvalue of

$$\mathbf{B} = \begin{bmatrix} \frac{\phi}{2} + \frac{\phi\kappa}{2}(\mathbf{L}\otimes\mathbf{I}_d) & \frac{\psi}{2}\mathbf{I}_{Nd} & \mathbf{0}_{Nd} \\ \frac{\psi}{2}\mathbf{I}_{Nd} & \frac{(\phi+1)}{2}\mathbf{I}_{Nd} & \frac{1}{2}\mathbf{I}_{Nd} \\ \mathbf{0}_{Nd} & \frac{1}{2}\mathbf{I}_{Nd} & \frac{1}{2}\mathbf{I}_{Nd} \end{bmatrix}.$$

Proof. Consider the following Lyapunov function

$$V = \underbrace{\kappa\frac{\phi}{2}\mathbf{x}^\top(\mathbf{L}\otimes\mathbf{I}_d)\mathbf{x} + \frac{\phi}{2}\|\mathbf{x}+\mathbf{z}\|^2}_{=:V_1} + \underbrace{\frac{1}{2}\|\mathbf{v}+\mathbf{z}\|^2}_{=:V_2}$$

with $\phi > 0$ as in the statement. The Lie derivative of V_1 along (3) can be written as $\dot{V}_1 = \eta\dot{W}_1$ where

$$\begin{aligned}\dot{W}_1 =& -\kappa\phi\mathbf{x}^\top(\mathbf{L}\otimes\mathbf{I}_d)\mathbf{x} + \underbrace{\kappa\phi\mathbf{x}^\top((\mathbf{L}+\mathbf{L}^\top)/2\otimes\mathbf{I}_d)\mathbf{z}}_{=:r_1} \\ &+ \underbrace{\phi\mathbf{x}^\top(\mathbf{z}-\mathbf{x})}_{=:r_2} - \phi\mathbf{x}^\top\mathbf{h} \underbrace{-\kappa\phi\mathbf{x}^\top(\mathbf{L}\otimes\mathbf{I}_d)\mathbf{z}}_{=:r_3} \underbrace{-\phi\mathbf{x}^\top\mathbf{v}}_{=:r_4} \\ &+ \underbrace{\phi\mathbf{z}^\top(\mathbf{z}-\mathbf{x})}_{=:r_5} \underbrace{-\phi\mathbf{z}^\top\mathbf{h}}_{=:r_6} - \kappa\phi\mathbf{z}^\top(\mathbf{L}\otimes\mathbf{I}_d)\mathbf{z} \underbrace{-\phi\mathbf{z}^\top\mathbf{v}}_{=:r_7}.\end{aligned}$$

Note that r_1 and r_3 cancels each other due to the symmetry property of $\mathbf{L}$. Furthermore, $r_2 + r_5 = -\phi\|\mathbf{x}\|^2 + \phi\|\mathbf{z}\|^2$, $r_4 \le \frac{5\phi^2}{4}\|\mathbf{x}\|^2 + \frac{1}{5}\|\mathbf{v}\|^2$, $r_6 \le \frac{\phi^2}{2}\|\mathbf{z}\|^2 + \frac{1}{2}\|\mathbf{h}\|^2$ and $r_7 \le \frac{5\phi^2}{4}\|\mathbf{z}\|^2 + \frac{1}{5}\|\mathbf{v}\|^2$ hold true. Substituting this to $\dot{V}_1$ yields

$$\begin{aligned}\dot{W}_1 \le& -\kappa\phi\mathbf{x}^\top(\mathbf{L}\otimes\mathbf{I}_d)\mathbf{x} - \kappa\phi\mathbf{z}^\top(\mathbf{L}\otimes\mathbf{I}_d)\mathbf{z} - \phi\mathbf{x}^\top\mathbf{h} + \frac{1}{2}\|\mathbf{h}\|^2 \\ &+ \frac{7\phi^2}{4}\|\mathbf{z}\|^2 + \frac{5\phi^2}{4}\|\mathbf{x}\|^2 + \phi\|\mathbf{z}\|^2 - \phi\|\mathbf{x}\|^2 + \frac{2}{5}\|\mathbf{v}\|^2.\end{aligned}$$

Furthermore, we have the Lie derivative of $\dot{V}_2 = \eta\dot{W}_2$ as follows

$$\begin{aligned}\dot{W}_2 =& \underbrace{\kappa\mathbf{v}^\top(\mathbf{L}\otimes\mathbf{I}_d)\mathbf{z}}_{=:s_1} \underbrace{-\mathbf{v}^\top\mathbf{h}}_{=:s_2} \underbrace{-\kappa\mathbf{v}^\top(\mathbf{L}\otimes\mathbf{I}_d)\mathbf{z}}_{=:s_3} - \mathbf{v}^\top\mathbf{v} \\ &+ \underbrace{\kappa\mathbf{z}^\top(\mathbf{L}\otimes\mathbf{I}_d)\mathbf{z}}_{=:s_4} \underbrace{-\mathbf{z}^\top\mathbf{h}}_{=:s_5} \underbrace{-\kappa\mathbf{z}^\top(\mathbf{L}\otimes\mathbf{I}_d)\mathbf{z}}_{=:s_6} \underbrace{-\mathbf{z}^\top\mathbf{v}}_{=:s_7}.\end{aligned}$$

Notice that $s_2 \le \frac{1}{5}\|\mathbf{v}\|^2 + \frac{5}{4}\|\mathbf{h}\|^2$, $s_5 \le \|\mathbf{z}\|^2 + \frac{1}{4}\|\mathbf{h}\|^2$ and $s_7 \le \frac{9}{4}\|\mathbf{z}\|^2 + \frac{1}{9}\|\mathbf{v}\|^2$. By using this, and the fact that s_1, s_3, s_4, and s_6 can be eliminated, $\dot{W}_2$ can be rewritten as follows

$$\dot{W}_2 \le -\frac{3}{5}\mathbf{v}^\top\mathbf{v} + \frac{9}{4}\|\mathbf{z}\|^2 + \frac{6}{4}\|\mathbf{h}\|^2.$$

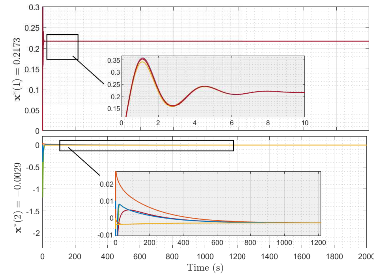

Fig. 2. Trajectory of $\mathbf{x}$ converge to the optimal point.

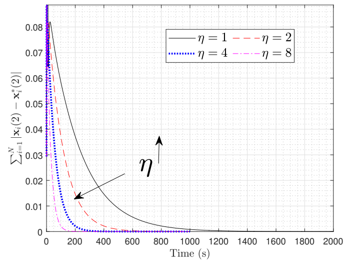

Fig. 3. The trajectory $\sum_{i=1}^N |\mathbf{x}_i(2) - \mathbf{x}^*(2)|$ tends to zero. Faster convergence is obtained as η increases.

By using $\dot{V}_1$ and $\dot{V}_2$ above, we obtain $\dot{V} = \eta\dot{W}_1 + \eta\dot{W}_2 =: \eta\dot{W}$, where $\dot{W}$ is given by

$$\begin{aligned}\dot{W} =& -\kappa\phi\mathbf{x}^\top(\mathbf{L}\otimes\mathbf{I}_d)\mathbf{x} - \kappa\phi\mathbf{z}^\top(\mathbf{L}\otimes\mathbf{I}_d)\mathbf{z} - \phi\mathbf{x}^\top\mathbf{h} - \frac{1}{5}\|\mathbf{v}\|^2 \\ &+ \left(\frac{7\phi^2+4\phi+9}{4}\right)\|\mathbf{z}\|^2 + \left(\frac{5\phi^2-4\phi}{4}\right)\|\mathbf{x}\|^2 + 2\underbrace{\|\mathbf{h}\|^2}_{\le M\mathbf{x}^\top\mathbf{h}} \\ \le& -\left(4\kappa\phi\lambda_2 - 7\phi^2 - 4\phi^2 - 7\right)\frac{\|\mathbf{z}\|^2}{4} - \frac{1}{5}\|\mathbf{v}\|^2 \\ &- \left(4\kappa\phi\lambda_2 - 5\phi^2 + 4\phi\right)\frac{\|\mathbf{x}\|^2}{4} - (\phi - 2M)\,\mathbf{x}^\top\mathbf{h}\end{aligned}$$

Due to $\phi > 2M$ and $\mathbf{x}^\top\mathbf{h} \ge \mu\|\mathbf{x}\|^2$, it follows that

$$\begin{aligned}\dot{W} \le& -\underbrace{\left(4\kappa\phi\lambda_2 - 7\phi^2 - 4\phi - 9\right)}_{=\gamma}\frac{\|\mathbf{z}\|^2}{4} - \frac{1}{5}\|\mathbf{v}\|^2 \\ &- \underbrace{\left(4\kappa\phi\lambda_2 - 5\phi^2 + 4\phi + 4\mu\phi - 8\mu M\right)}_{=\xi}\frac{\|\mathbf{x}\|^2}{4}.\end{aligned}$$

Observe that $\xi > 0$ and $\gamma > 0$ in the statement ensure that $V(t)$ is bounded and $\dot{V} < -\eta\min\{\frac{\xi}{4}, \frac{\gamma}{4}, \frac{1}{5}\}\|\mathbf{y}\|^2 < 0$ where $\mathbf{y} = (\mathbf{x}^\top, \mathbf{z}^\top, \mathbf{v}^\top)^\top$. Thus, $\mathbf{x}$ converges to $\mathbf{x}^*$ exponentially with rate no less than (6). The rate of convergence follows Theorem 4.10 in Khalil (2002).

Remark 3. Note that there always exists κ satisfying $\gamma, \xi > 0$. As a matter of fact, (5) can be ensured by selecting a sufficiently large κ. Moreover, it is evident here that faster convergence can be obtain by enlarging the value of η.

In next section, numerical simulations are given to validate the proposed algorithm.

Presented as late breaking results contribution
21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

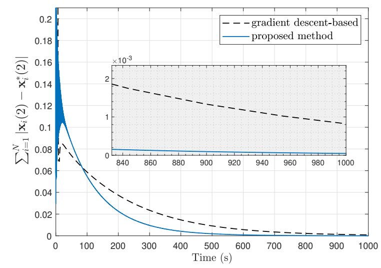

Fig. 4. The proposed method shows improved convergence property than the existing algorithm.

5. SIMULATION

Consider a network of ten agents illustrated in Figure 1. All $f_i(\mathbf{x})$, $i = 1, \ldots, 10$, with $\mathbf{x} \in \mathbb{R}^2$ in (1) are defined as follows

$f_1 = 0.5e^{-0.5\mathbf{x}(1)} + 0.5e^{0.3\mathbf{x}(2)} + 0.4e^{5\mathbf{x}(1)}$,
$f_2 = (\mathbf{x}(1)-4)^2 + (10^2\mathbf{x}(2)-4)^2$, $f_3 = 0.5(1+\mathbf{x}(2)^2)+\mathbf{x}(1)^2$,
$f_4 = \mathbf{x}^2(1) + e^{0.1\mathbf{x}(2)}$, $f_5 = e^{-0.1\mathbf{x}(1)} + 10^3e^{0.3\mathbf{x}(2)}$,
$f_6 = \mathbf{x}^4(1) + \mathbf{x}^2(2)$, $f_7 = 0.2e^{-0.2\mathbf{x}(1)} + 0.4e^{10^2\mathbf{x}(2)}$,
$f_8 = \mathbf{x}^4(1) + 2\mathbf{x}^2(2) + 2$, $f_9 = e^{-0.2\mathbf{x}(1)} + 5 \times 10^2\mathbf{x}^2(2)$,
$f_{10} = e^{-0.2\mathbf{x}(1)} + (5 \times 10^2\mathbf{x}(2)+2)^2$.

The proposed distributed algorithm is employed with $\eta = 1$. Figure 2 shows that equilibrium points of all agents reach to the same equilibrium point. The equilibrium point is the optimal solution as it is the same as the value given by a centralized algorithm: $\mathbf{x}^* = [0.2173; -0.0029]$. Note that the convergence of $\mathbf{x}(2)$ is slower than $\mathbf{x}(1)$ because the order of magnitude of $\mathbf{x}(2)$ in the global objective function is smaller than that of $\mathbf{x}(1)$. Figure 3 implies that the trajectory of $\sum_{i=1}^{N} |\mathbf{x}_i(2) - \mathbf{x}^*(2)|$, i.e. error trajectory correspond to $\mathbf{x}(2)$, tends to zero as $t \to \infty$. Moreover, enhanced convergence property is obtained by enlarging the value of η . As depicted in Figure 4, the proposed method with $\eta = 2$ shows better convergence property compared to the existing gradient-based distributed algorithm Kia et al. (2015) .

6. CONCLUSION

A novel continuous-time coordination algorithm for an undirected graph is proposed. The method yields exponential convergence, with the rate linearly dependent on the value of parameter design η. The main advantage of the method is that the convergence can be easily accelerated to some point by choosing a larger η. The parallel result for a directed and strongly connected graph will be published separately. Furthermore, possible future research includes an accelerated distributed algorithm that exactly mimics the continuous-time Nesterov accelerated gradient method, in which it is parametrized by increasing functions instead of constant η.

ACKNOWLEDGEMENTS

This work was supported by the Human Resources Development of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) Grant Funded by the Ministry of Trade, Industry and Energy of the Korean Government under Grant 20154030200720.

REFERENCES

Boyd, S.P., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1), 1–122.

Cherukuri, A. and Cortes, J. (2015). Distributed generator coordination for initialization and anytime optimization in economic dispatch. *IEEE Transactions on Control of Network Systems*, 2(3), 226–237.

Gharesifard, B. and Cortes, J. (2014). Distributed continuous-time convex optimization on weight-balanced digraphs. *IEEE Transactions on Automatic Control*, 59(3), 781–786.

Johansson, B., Rabi, M., and Johansson, M. (2010). A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Optimization*, 20(3), 1157–1170.

Khalil, H.K. (2002). *Nonlinear Systems, Third Edition*. New York.

Kia, S.S., Cortes, J., and Martinez, S. (2015). Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication. *Automatica*, 55, 254 – 264.

Li, Z., Ding, Z., Sun, J., and Li, Z. (2018). Distributed adaptive convex optimization on directed graphs via continuous-time algorithms. *IEEE Transactions on Automatic Control*, 63(5), 1434–1441.

Nedic, A. and Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1), 48–61.

Nesterov, Y. (1983). A method for solving the convex programming problem with convergence rate O(1/kˆ2). *Soviet Mathematics Doklady*, 27, 372–376.

Polyak, B.T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5), 1–17.

Rabbat, M. and Nowak, R. (2004). Distributed optimization in sensor networks. In *Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004*, 20–27.

Ram, S.S., Veeravalli, V.V., and Nedic, A. (2010). Distributed and recursive parameter estimation in parametrized linear state-space models. *IEEE Transactions on Automatic Control*, 55(2), 488–492.

Song, Y., Song, J., and Ermon, S. (2018). Accelerating natural gradient with higher-order invariance. In *35th International Conference on Machine Learning, ICML 2018*, 4720–4729.

Varagnolo, D., Zanella, F., Cenedese, A., Pillonetto, G., and Schenato, L. (2016). Newton-raphson consensus for distributed convex optimization. *IEEE Transactions on Automatic Control*, 61(4), 994–1009. doi: 10.1109/TAC.2015.2449811.

Wibisono, A., Wilson, A.C., and Jordan, M.I. (2016). A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 133, E7351–E7358.

Wilson, A.C., Recht, B., and Jordan, M.I. (2016). A lyapunov analysis of momentum methods in optimization.