A Distributed Optimization Approach for Multi-Agent Systems over Delaying Networks

Farshad Rahimi*
Department of Electrical Engineering
Sahand University of Technology
Tabriz, Iran.
fa_rahimi@sut.ac.ir

Received: 15 August 2021 - Accepted: 14 October 2021

Abstract—This paper investigates a novel method to solve distributed optimization problems in the presence of communication delays between the networked agents that cooperate together to find an optimal solution of a global cost function composed of local ones. In the problem of distributed optimization in a network of multi-agent because of existing phenomena such as communication delay, deriving approaches having appropriate performance so that the states of all agents converge to the same value always has been a substantial challenge. Delay-dependent conditions in the form of linear matrix inequities are derived to analyze the convergence of the introduced scheme to the optimal solution. It is demonstrated that the maximum allowable time delay in the network and convergence rate of the optimization procedure are increased by the suggested strategy. Finally, comparative simulation results are considered to illustrate the superior performance of the introduced scheme compared to a rival one in the literature.

Keywords: Distributed optimization; Communication delay; Linear Matrix Inequality; Multi-agent systems.

Article type: Research Article

© The Author(s).
Publisher: ICT Research Institute

I. INTRODUCTION

In the problem of distributed optimization, a global cost function composed of several local ones is optimized by the cooperation of networked agents. Each agent which has access only to one local objective function contributes to the solution of the problem through local computations and information exchange with the neighbors. Due to widespread applications of distributed optimization in real-world systems such as distributed estimation in sensor networks, motion planning, distributed predictive control, resource allocation over networks, and the economic power dispatch problem, this issue has attracted much attention in recent years [1-6].

The majority of works in the area of distributed optimization for multi-agent systems are focused on discrete-time formulations [7-11]. However, because of
the simple application of the Lyapunov stability theorem for convergence analysis and also potential applications in continuous-time physical systems, much attention has been recently concentrated on the study of continuous-time distributed optimization [12-16].

In most of the aforementioned publications, communication channels between agents were assumed to be ideal, for instance, see the work of Weng et al [17], while time delays are inevitable in data exchange between agents in the networks due to a finite bandwidth of the communication medium and limited speed of agents for calculating and sending their outputs [18, 19]. Also, in practical systems such as unmanned aerial vehicles (UAVs), when the information would be transmitted by an UAV to others, occurring time delay is inevitable. Compensation of destructive effect of communication delays on the performance and stability of solution process is an important challenge that is addressed only by a few papers in the literature. Therefore, it is beneficial to propose the approaches that consider communication delay in multi agent systems [20].

Authors in [21] by utilizing the mirror descent method, derived a distributed algorithm to solve a distributed optimization problem, which phenomenon of time delays in a multi-agent system are considered and analyzed the effects of delay on convergence rate. In [22], based on the dual averaging notion, an algorithm was derived to solve distributed cooperative optimization problems subject to delayed sub-gradient data in a networked system with delay. In [23], the problem of distributed optimization in the presence of inter-agent communication delays was solved via a proportional-integral consensus algorithm in a passivity-based framework. It was proved that the transmission delays can be handled while ensuring the convergence property using scattering transformation. Authors have developed a distributed optimization approach for a continuous-time multi-agent system subject to communication delays in [24]. Sufficient conditions in the form of linear matrix inequities were presented for convergence to analysis. In the paper by Lin et al [25], distributed optimization problem was solved by implementing a sub-gradient projection algorithm for a networked system subject to communication delays and nonidentical constraints. Moreover, a distributed optimization problem of multi-agent systems with delayed sampled data is considered in [26], then based on Lyapunov theory and graph the convergence of all the agents to the optimal solution was proved.

In this paper, inspired by [24] and [25], a novel technique is developed to solve efficiently the distributed optimization problem in the presence of communication delays utilizing the sub-gradient projection idea. The key idea to improve the approaches of [24] and [25] is that, for each agent based on the communication graph, a weighted information of its neighbors in the gradient term is implemented to update the value of the state of each agent. The main contributions of the proposed approach can be highlighted as follows:

1. Convergence of the proposed algorithm is assured using the Lyapunov-Krasovskii stability argument;
2. It is demonstrated that compared to the rival method in [24], the convergence rate is increased by the proposed strategy; moreover, convergence to the optimal solution is achieved for higher values of transmission delay.

The remainder of this paper proceeds as follows. In section 2, first, the necessary background materials are called form literature and then the problem of distributed optimization for the system with multi-agent is modeled. The proposed algorithm is given in Section 3, then, the convergence of the optimization process is analyzed. In Section 5, comparative simulation results are presented to verify the superiority of the suggested scheme compared to the rival method in the literature. Ultimately, the conclusion is given in 6.

II. PROBLEM STATEMENT AND PRELIMINARIES

Considering a network of $\Lambda$ agents interacting over a directed graph $\Xi(V,E)$ that consists of nodes (agents) set, $V=[1, \ldots, \Lambda]$ and an edge set, $E \subseteq V \times V$. An edge from $v_i$ to $v_j$, described by $(v_i, v_j)$, denotes that $v_j$ can obtain data from $v_i$; then $v_j$ is known a neighbor of $v_i$. The set $\Lambda_i = \{ j \mid (i, j) \in E \}$ denotes all neighbors of the agent $i$. The considered graph is assumed to be strongly connected; namely, for every pair of nodes, there is a directed path connecting them. The adjacency matrix of the graph $\Xi(V,E)$ is represented by $A$ which is a $\Lambda \times \Lambda$ matrix, whose entries $a_{ij}$ are given as:

$$a_{ij} = \begin{cases} 1 & \text{if } i \neq j \text{ and } (i, j) \in E, \\ 0 & \text{otherwise}. \end{cases}$$ (1)

The out-degree and in-degree of a node $v_i$ in the graph are $\tilde{d}_i = \sum_{j=1}^{\Lambda} a_{ji}$ and $\tilde{d}_i = \sum_{j=1}^{\Lambda} a_{ij}$, respectively.

The considered graph $\Xi$ is supposed to be weighted-balanced; that is, $\tilde{d}_i = \tilde{d}_i$ for any agent $v_i \in V$ [27, 28]. The overall cost function of the minimization problem is considered as follows:

$$F(x) = \sum_{i=1}^{\Lambda} f_i(x)$$ (2)

wherein the agent $i$ only accesses to its local cost, $f_i(x)$ which $m_i$ strongly is convex in $x$. That is,

$$\nabla f_i(z) - \nabla f_i(x) \leq l_i (z-x)^T (\nabla f_i(z) - \nabla f_i(x)) \quad \text{and} \quad \nabla f_i = l_i - \text{Lipchitz over } \mathbb{R}^n \quad \text{that is for all } x,z \in C \subset \mathbb{R}^n .$$

The aim is to find variables $x=(x_1, x_2, \ldots, x_\Lambda) \in \mathbb{R}^\Lambda$ such that the general objective function $F(x)$ in (2) attains its minimal value. For minimization of (2), the agents, $V$ in the network coordinate their decisions through a set of communication links. To formulate this fact, the coupling among agents is transformed to a set of
The constraints [29]. Therefore, the minimization of structure (2) is defined as:

\[
\text{minimize } F(x) = \sum_{i=1}^{\Lambda} f_i(x)
\]

s.t. \( x_i = x_j, \ i \in [1,...,\Lambda], \ j \in \Lambda, \)

where \( x \in \mathbb{R}^n \) denotes the state of the agent \( i \). The role of each agent \( i \) is to contribute to finding the optimal solution of the problem (3) via cooperation with other agents. Inspired by the distributed optimization techniques in literature such as dual-decomposition and consensus fashion [15, 30, 31] the following networked system is obtained to find the solution of the optimization problem.

\[
\dot{x}_i = \gamma \sum_{j \neq i} a_{ij} [x_i - x_j],
\]

\[
\dot{x}_i = -\alpha \nabla f_i(x_i) - \beta \sum_{j \neq i} a_{ij} [x_i - x_j],
\]

where \( \alpha, \beta \) and \( \gamma \) are positive weights and the weighting that agent \( i \) associates to the agent \( j \) is \( a_{ij} \).

Note that the system (4) inherits the properties of dual-decomposition and consensus methods in solving the distributed optimization problem; namely, the fast transient behavior of consensus and desirable convergence of dual-decomposition.

Since the agents exchange information together through a communication graph subject to time-delay, the information of neighbors are not immediately in hand to be used in (4), i.e. at step time \( t \), the amount of \( x_i(t - d(t)) \) is available instead of \( x_i(t) \), where \( d(t) \) is assigned to demonstrate time-varying delay among agents \( i \) and \( j \), which is bounded as \( 0 < d(t) \leq \bar{d}, \) with \( d(t) < 1 \). In the next section, system (4) is refined to tackle this issue efficiently. Before proceeding, some useful facts which will be employed in the derivation of our results are recalled from the literature.

Lemma 1. [15]. Laplace matrix concerned with the graph \( \Xi \) is described as the following representation:

\[
L = \left[ \ell_{ij} \right] \in \mathbb{R}^{N \times N}, \ \ell_{ij} = \begin{cases} \sum_{\alpha \beta} a_{ij} & i = j \\ -a_{ij} & i \neq j \end{cases}
\]

(5)

It is alternatively can be construed as \( L = D_{out} - A \), with \( D_{out} = \text{diag} \left\{ \bar{d}_{11}, \bar{d}_{22}, \ldots, \bar{d}_{nn} \right\} \in \mathbb{R}^{N \times N} \). The Laplacian \( L \) of graph \( \Xi \) is a positive semi-definite matrix and for a connected graph, the Laplacian has a single zero eigenvalue and the corresponding eigenvector is a vector of ones, defined by \( \left\{ k_1: k \in \mathbb{R}^n \right\} \), which \( 1_n \in \mathbb{R}^n \).

Lemma 2. [32]. For an undirected connected graph with symmetric \( L \), there exist \( \Lambda - 1 \) real eigenvalues in the open right half plane (RHP) \( 0 = \Lambda_1 < \Lambda_2 \leq \ldots \leq \Lambda_{\Lambda - 1} \). The matrix \( L \) can be diagonalized through an orthogonal transformation as the following:

\[
L = RJR^T
\]

where the diagonal matrix \( J \) is as

\[
J = \begin{bmatrix} 0 & 0 & \ldots & 0 \\ 0 & \Lambda_{\Omega(1)} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \Lambda_{\Omega(1)} \end{bmatrix}
\]

in which \( Y \) has a \((\Lambda-1) \times (\Lambda-1)\) diagonal matrix containing the mentioned RHP eigenvalues of \( L \) and \( R \) is constructed using the right eigenvectors of \( L \) as \( R = [r_1, r_2, \ldots, r_{\Lambda}] \), wherein \( r_i, i \in [1,2,\ldots,\Lambda] \), defining the right eigenvectors of \( L \) where \( r_i = 1 / \sqrt{\Lambda} \). Lemma 3. [33]. For any function \( x(t) \in L_2[-s,0] \) and any positive matrix \( \Psi \), the following inequality holds

\[
\int_{-s}^{0} x^T(t) \Psi x(t) dt \geq \frac{1}{\bar{d}} \left( \int_{-s}^{0} x^T(t) dt \right) \Psi \left( \int_{-s}^{0} x(t) dt \right)
\]

Lemma 4 [34]. Assume that we have \( Q_i \in \mathbb{R}^{n \times n}, \ldots, Q_{\Lambda} \in \mathbb{R}^{n \times n} \) been positive matrices. For all \( a_{ij} > 0 \) with \( \frac{\sum_{i} a_{ij}}{a_{ij}} = 1 \) and for all \( S_{ij} \in \mathbb{R}^{n \times n}, \)

\[
\sum_{i=1}^{\Lambda} e_i^T Q_i e_i \geq \frac{1}{\bar{d}} \left( \sum_{i=1}^{\Lambda} \sum_{j=1}^{\Lambda} e_i^T Q_{ij} e_j \right)
\]

The following relation holds:

\[
\sum_{i=1}^{\Lambda} e_i^T Q_i e_i \geq \begin{bmatrix} e_1^T & \ldots & e_\Lambda^T \end{bmatrix} \begin{bmatrix} Q_1 & S_{12} & \ldots & S_{1\Lambda} \\ S_{21} & Q_2 & \ldots & S_{2\Lambda} \\ \vdots & \vdots & \ddots & \vdots \\ S_{\Lambda 1} & S_{\Lambda 2} & \ldots & Q_{\Lambda} \end{bmatrix} \begin{bmatrix} e_1 \\ \vdots \\ e_\Lambda \end{bmatrix}
\]

for all \( e_1 \in \mathbb{R}^n, \ldots, e_\Lambda \in \mathbb{R}^n \).

III. MAIN RESULTS

In this section, a novel continuous-time system is provided to be replaced with (4) in the case of delayed information; then convergence of the proposed method is analyzed. The merits of the proposed method are illustrated in the next section by simulation. The delayed subgradient information received at time step \( t \) is used to update evolution of \( x_i(t) \) in the agent \( i \) can be defined as the following form:

\[
\dot{x}_i(t) = \gamma \sum_{j=1}^{\Lambda} a_{ij} \left[ x_j(t - d(t)) - x_i(t - d(t)) \right]
\]

\[
\dot{x}_i(t) = -\alpha \nabla f_i(x_i(t)) + \sum_{j=1}^{\Lambda} w_{ij} x_j(t - d(t)) \]

\[
- \beta \sum_{j=1}^{\Lambda} a_{ij} \left[ x_i(t - d(t)) - x_j(t - d(t)) \right]
\]

(6)

for \( i \)th agent and its neighbors, the weight coefficients \( w_{ij}^c \) and \( w_{ij}^d \) are devoted, respectively; such that matrices \( w_{ij}^c \) and \( w_{ij}^d \) satisfy:

\[
w_{ij}^c = \text{diag}[w_{ij}^c]_{1 \times \Lambda}, \quad w_{ij}^d = \text{diag}[w_{ij}^d]_{1 \times \Lambda}
\]

with \( w_{ij}^d = 1, \ i = 1,...,\Lambda \). Now, we can rewrite system (6) in the compact form as the following representation:
\[ z(t) = \gamma Lx(t - d(t)) \]
\[ x(t) = -\alpha \nabla F \left( w_x(t) + w_x(t - d(t)) \right) - \beta Lx(t - d(t)), \] (7)

where \( x = (x_1^T, x_2^T, \ldots, x_n^T) \),
\[ \nabla F(x(t)) = (\nabla f_1(x_1(t)), \nabla f_2(x_2(t)), \ldots, \nabla f_n(x_n(t)))^T, \]
\[ z = (z_1, z_2, \ldots, z_k)^T, \]

\( L = L \otimes I_n \in \mathbb{R}^{m \times m} \) and \( \otimes \) represents the Kronecker product. Note that time delay appears in the \( x \) component. Therefore, the initial value \( x \) needs to be known as a function \( x_0 \) in \( C([-d, 0], \mathbb{R}^m) \) which is the space of continuous functions mapping the interval \([-d, 0]\) into \( \mathbb{R}^m \) subject to norm \( \varphi = \sup_{-d, x, \delta} \varphi(\delta) \).

A. Optimality Analysis

In the following theorem, it is proven that the system (7) converges to the optimal solution of (3), subject to delayed transmission. Let \( f_i \in \mathcal{V} \) is differentiable and \( m_t \) -strongly convex, and its gradient is \( l_t \) -Lipschitz also, the graph \( \Xi \) is strongly connected and weighted-balanced.

**Theorem 3.1:** For any \( \varepsilon \in \mathbb{R}^n \),
\[ S(\varepsilon) = \{(z, x) \mid (I_n \otimes l_x)^T z = \varepsilon \} \]

is a positive invariant set and \( x^* \) is an optimal solution of (3) if and only if \( \left( -\alpha \nabla F \left( w_x^* + w_x^* \right), x^* \right) \in S(0) \) is an equilibrium of systems (7).

**Proof.** See the appendix.

B. Convergence analysis

Next, in order to find the optimal solution \( x^* \) to system (7), sufficient conditions in the form of LMIs are presented to guarantee the convergence.

**Theorem 3.2:** The System (7) with time-delay \( d(t) \leq \sigma < 1 \) and the initial condition \( S(0) \) is convergent to the optimal solution of (3) if there exist \( (A-1) \times (A-1) \) matrix \( S, P_1, P_2 \), and positive definite matrices \( Q_1, Q_2, Q_3 \) and positive scalars \( \delta, \rho_{11} \) and \( \rho_{22} \) such that:

\[
\begin{pmatrix}
Q_2 & S \\
* & Q_2
\end{pmatrix} > 0
\]
\[
\begin{pmatrix}
\rho_{11} I_{A-1} & P_1 \\
P_1^T & \rho_{22} I_{A-1}
\end{pmatrix} > 0
\]

With
\[
\begin{pmatrix}
\Pi_1 & 0 & -P_2 & \Pi_{14} & \Pi_{15} & 0 \\
* & \Pi_{22} & -p_{22} I_{A-1} & \Pi_{24} & \Pi_{25} & S \\
* & * & \Pi_{33} & 0 & \Pi_{36} & 0 \\
* & * & * & \Pi_{44} & \Pi_{45} & 0 \\
* & * & * & * & \Pi_{55} & \Pi_{56} \\
* & * & * & * & * & *
\end{pmatrix}
\]

\[
\begin{pmatrix}
-\alpha P_0 & 0 & 0 \\
-\alpha P_4^T & 0 \\
-\alpha P_2 & 0 & -\delta Q_2 \\
\Pi_{47} & \Pi_{48} & 0 \\
\Pi_{57} & \Pi_{58} & -\beta \delta Q_2 J < 0 \\
0 & 0 & 0 \\
-\delta I_{A-1} & 0 & -\alpha \delta Q_2 \\
0 & * & -Q_2
\end{pmatrix}
\] (8)
which \( m = \min \{m_1, m_2, \ldots, m_n\} \) that \( m_i \)-strongly is convex, \( \hat{I} = \max \{I_1, I_2, \ldots, I_n\} \) that \( I_i \) is Lipschitz coefficient, \( a_i \in \text{diag} [w_i \Lambda]_{n \times n} \).

\[
\begin{bmatrix}
z_{11} & z_{12} \\
z_{21} & z_{22}
\end{bmatrix} = \begin{bmatrix} w_a w_s; \\
\hat{g}_{11} & \hat{g}_{12} \\
\hat{g}_{21} & \hat{g}_{22}
\end{bmatrix} = w_a^T w_s
\]

\[
\begin{bmatrix}
k_{11} & k_{12} \\
k_{21} & k_{22}
\end{bmatrix} = w_a^T.
\]

**Proof:** see the appendix.

It is noteworthy to mention that as stated in the next section, the values of the weight coefficients have been devised to derive sufficient conditions of the convergence. In other words, \( w_a \) and \( w_s \) affect the feasibility of LMIs (8).

**IV. BENEFIT AND APPLICATION OF THIS KIND OF OPTIMIZATION**

**BENEFIT:** This paper is assigned to investigate the problem of distributed multi-agent optimization over delaying networks. The optimization equation (6) is derived for objective enduring higher values of transmission delay and increasing of the converge rate. Differently from (4) in which the gradient of \( f_i \) is computed only on \( x_i \), in the proposed system (6), the gradient term is evaluated on a weighted sum of \( x_i \) and all delayed \( x_i \). In equation (6), both weight coefficients \( w_a \) and \( w_s \) would be chosen by the designer, and as demonstrated in the proceeding, these weight coefficients are implemented in establishing the sufficient conditions to assure the convergence of the proposed approach (i.e. in Linear Matrix Inequalities (LMIs) (8)). In addition, as shown later in the simulations, this idea leads to improved performance compared to [24].

**APPLICATIONS:** As we know, the distributed optimization technique is the backbone for the learning and formation control of many practical applications. For instance, networked mobile robots are collaborating with each other in order to reach a certain task, therefore, they have to cooperate together to minimize a collective cost such that a central computing station was not employed. As a case study, we can use this proposed optimization approach in [35], where networked mobile robots are connected through a communication graph and are aimed to minimize their speeds and maximize their distance from each other. This kind of optimization approach also could be implemented in problems such as [15], and collaborative multi-robot systems for search and rescue [36].

**V. SIMULATION RESULTS**

In this section, the advantages of the addressed distributed optimization method are provided by two examples.

**Example 1:** In this example, the results of the performance of the proposed scheme are compared to a rival one in the literature. A network including five agents is considered subject to a graph which can be seen in Figure 1. The communication graph is weight-balanced and strongly connected.

Note that in order to easily see, fewer agents are deliberately considered in the network for simulation results. Although if we consider a network with more agents, better results would be obtained for the proposed method. Local cost functions with \( x \in \mathbb{R} \) are as the following

\[ f_1(x) = 0.9(x^2 + 2x + 1), \quad f_2(x) = (x - 4)^2, \]

\[ f_3(x) = 0.5x^2 - 1, \quad f_4(x) = \sin \left( \frac{x}{2} \right) + \frac{x^2}{2}, \quad f_5(x) = (x - 3)^2 \]

which are strongly convex and globally Lipschitz. It can be simply verified that the general cost function \( F = \sum_{i=1}^{5} f_i \), gets its minimum value \( F_{\text{min}} = 16.05 \), at \( x_{\text{min}} = 1.51 \). In the first simulation, the delay value is supposed to be constant \( d = 0.62 \). The values of the weight coefficients are chosen as follows:

\[
\begin{bmatrix}
1.3 & 0 & 0 & 0 & 0 \\
0 & 1.3 & 0 & 0 & 0 \\
0 & 0 & 1.3 & 0 & 0 \\
0 & 0 & 0 & 1.3 & 0 \\
0 & 0 & -0.3 & 0 & 0 \\
-0.3 & 0 & 0 & 0 & 0 \\
0 & 0 & -0.3 & 0 & 0 \\
0 & 0 & 0 & -0.3 & 0
\end{bmatrix}
\]

\[
\begin{bmatrix}
w_a = [1.3 & 0 & 0 & 0 & 0; 0 & 1.3 & 0 & 0 & 0; 0 & 0 & 1.3 & 0 & 0; 0 & 0 & 0 & 1.3 & 0; 0 & 0 & -0.3 & 0 & 0; -0.3 & 0 & 0 & 0 & 0; 0 & 0 & -0.3 & 0 & 0; 0 & 0 & 0 & -0.3 & 0]
\end{bmatrix}
\]

Figure 2 (a) shows the result obtained by the algorithm of [26] and Figure 2(b) depicts the consequence of the proposed algorithm with \( \alpha = \beta = \gamma = 0.6 \). Although, two methods can tolerate the same time delay it is clear that the convergence speed in Figure 2(b) is considerably faster than Figure 2(a).
Fig 2. The agents’ state trajectories for $d = 0.62$. (a): the method of [26] and (b) the proposed method.

Fig 3. The agents’ state trajectories for $d = 0.78$. (a): the method of [26] and (b) the proposed method.

In Figure 3 state trajectories of agents are shown for $d = 0.78$ with rival and suggested schemes to verify that the maximum allowable delay in our method is much larger compared to [24]. To convenient evolution of maximum tolerable delay value, maximum allowable delays are reported in table 1 for the introduced method and rival schemes. It is clear that the converging criterion of the proposed method is less conservative compared to the maximum allowable delay of the rival scheme.

TABLE I. THE PERFORMANCE OF THE PROPOSED APPROACH

<table>
<thead>
<tr>
<th>Approaches</th>
<th>$*$</th>
<th>$**$</th>
</tr>
</thead>
<tbody>
<tr>
<td>The proposed method</td>
<td>0.82</td>
<td>0.8171</td>
</tr>
<tr>
<td>The Method of [24]</td>
<td>0.64</td>
<td>0.6093</td>
</tr>
</tbody>
</table>

$* $ Actual value

$** $ Obtained from stability condition

Example 2: Consider the following local cost functions subject to the communication graph in Figure 4:

$$F(x) = \sum_{i=1}^{k} f_i(x).$$

(11)

where

$$f_1(x) = (x-6)^2, f_2(x) = 0.8x^2 + 2,$$

$$f_3(x) = 0.9(x^2 + 2x + 1),$$

$$f_4(x) = (x-4)^2, f_5(x) = 0.5x^2 - 1,$$

$$f_6(x) = \sin \left(\frac{x}{2}\right) + \frac{x^2}{2},$$

$$f_7(x) = 0.6x^2 + x, f_8(x) = (x - 3)^2$$

(12)

Now, System (7) including eight agents is employed to solve optimization structure (11) subject to (12). It is assumed that we have the communication delays via a determined value. The parameters in system (7) are selected as $\alpha = 0.6, \beta = \gamma = 0.5$. The values of the weight coefficients are chosen as follows:

$$w_a = \begin{bmatrix} 1.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.3 & 0 \end{bmatrix},$$

$$w_n = \begin{bmatrix} 0 & -0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.1 & 0 & 0 & -0.1 & 0 & 0 \\ -0.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.3 & 0 & 0 & 0 \\ 0 & -0.3 & 0 & 0 & 0 & 0 & -0.3 & 0 \end{bmatrix}.$$

Fig 4. The communication topology of eight agents.

Fig 5. The agents’ state trajectories for $d = 0.6$.  

Volume 13- Number 4 – 2021 (18 -27)
According to the communication graph in Figure 4, the simulation results for time-delay $d = 0.60$, $d = 0.72$ and $d = 0.80$ are shown in Figures 5-7, respectively. Figure 7, unlike Figure 5 and 6, $x(t)$ is not convergent.

VI. CONCLUSIONS

In this paper, the problem of distributed multi-agent optimization over delaying networks is investigated. A novel method has been presented to solve a distributed continuous-time optimization problem with a multi-agent system subject to communication delay. Sufficient conditions have been derived in terms of LMIs to check the convergence of the algorithm to the optimal solution utilizing Lyapunov-Krasovskii’s theory. Comparative simulation results have been presented to demonstrate that maximum allowable delay and rate of convergence are both improved compared to the recent rival method in the literature. Many issues are still open for future research, for instance considering uncertainties, and quantization effects [37] in the parameters of a multi-agent system.

VII. APPENDIX

Proof. of Theorem 3.1.

Regarding that communication delay occurs in the $x$ component of the delayed system (7), the initial value of system (7) is $x(0)$. From Lemma 2, we have

$$
(1_\lambda \otimes I_\lambda)\left( L \otimes I_\lambda \right) = (0_\lambda \otimes I_\lambda)^T
$$

So

$$
(1_\lambda \otimes I_\lambda)\left( L \otimes I_\lambda \right) x(t) = \gamma (1_\lambda \otimes I_\lambda)^T x(t)
$$

$$
L x(t - d(t)) = 0
$$

(13)

which states that the set $S(\varepsilon)$ is invariant for any $\varepsilon$, therefore:

$$
(1_\lambda \otimes I_\lambda)\left( L \otimes I_\lambda \right) x(t) = (1_\lambda \otimes I_\lambda)^T x(0)
$$

(14)

which implies $\varepsilon = (1_\lambda \otimes I_\lambda)^T x(0)$.

Additionally, assume $(\mathbf{v}^*, \mathbf{x}^*)$ denotes the equilibrium point of (7) in $S(0_n)$, and recall the Laplace matrix $L$ concerned with a graph, then from Lemma 2, we have

$$
0 = \gamma Lx^* = -\alpha \nabla \mathbf{w} \mathbf{X}^* - \beta Lx^* - z^*
$$

(15)

wherein $\mathbf{X}^* = 1_\lambda \otimes x^*$. By multiplying $(1_\lambda \otimes I_\lambda)^T$ to (15) from the left and regarding (13), we have

$$
-\alpha \nabla \mathbf{w} \mathbf{X}^* = 0
$$

(16)

Regarding weight coefficients assigned to agents in (6), the related $X^* = w_\mathbf{x} x^* + w_\mathbf{X} \mathbf{X}^*$ results in $X^* = \mathbf{x}^*$ and we have the following:

$$
-\alpha \nabla \mathbf{w} \mathbf{X}^* = -\alpha \sum_{i=1}^n \mathbf{w}_i \nabla f_i(\mathbf{x}^*)
$$

(17)

to find the optimal condition of $X^*$. In addition, due to $0 = \gamma Lx^*$ and $0 = -\alpha \nabla \mathbf{w} \mathbf{X}^* - \beta Lx^* - z^*$, we can obtain $z^* = -\alpha \nabla \mathbf{w} \mathbf{X}^*$. The proof of converse is uncomplicated. Note that the initial value for solving the problem (6) requires to be located in $S(0_n)$ otherwise, if $\varepsilon = (1_\lambda \otimes I_\lambda)^T z(0) \neq 0$, due to (15) we have $\alpha (1_\lambda \otimes I_\lambda)^T \nabla f(\mathbf{x}) \neq 0$, that verifies the system (6) may converge to the vector $\mathbf{x}$ that is not the optimal solution.

Proof. of Theorem 3.2. Consider definitions

$$
\overline{\mathbf{x}} = \mathbf{z} - z^* \quad \text{and} \quad \overline{\mathbf{x}} = \mathbf{x} - \mathbf{x}^* ,
$$

the equilibrium point transformers to the origin by replacing $\mathbf{z}$ and $\mathbf{X}$ within the structure (7).

$$
\overline{\mathbf{x}} = (\mathbf{R} \otimes I_\lambda) \mathbf{v},
$$

$$
\overline{\mathbf{x}} = (\mathbf{R} \otimes I_\lambda)(w_\mathbf{X} \mathbf{u}(t)) + (\mathbf{R} \otimes I_\lambda)
$$

(18)

$$
\times (w_\mathbf{x} \mathbf{u}(t - d(t))
$$

$\mathbf{R}$ is in the form of $[\mathbf{R}_1 \mathbf{R}_2]^T$ with $\mathbf{R}_1 = \frac{1}{\sqrt{\mathbf{X}}}$. Let $\mathbf{v}$ be $(v_1, v_2)^T$ with $v_1 \in \mathbb{R}_i$, $v_2 \in \mathbb{R}(\mathfrak{A}_i)$ and in a similar process: $\mathbf{u} = (u_1, u_2)^T$. Since the initial condition belongs to $S(0_n)$, system (7) can be converted to
\[ \dot{v}_i = 0, \]
\[ \dot{v}_{2A} = \gamma J u_{2A} \left( t - d(t) \right) \]
\[ \dot{u}_1 = -\alpha r_1^T h(z) \]
\[ \dot{u}_{2A} = -\alpha R^T_1 h(z) - \beta J u_{2A} \left( t - d(t) \right) \]
\[ -v_{2A} \left( t \right) \]  \hspace{1cm} (10)

wherein \( J = J \otimes I_n \), \( r_i = r_i \otimes I_n \), \( R_2 = R_2 \otimes I_n \) and \( h(z) = \nabla F \left( z + z' \right) - \nabla F \left( z' \right) \)  \hspace{1cm} (19)

Concerning the initial condition and \( v_1 = 0 \), we can have \( v_1 = 0 \). Then, we only need to focus on the other three relations of (18).

\[ V = V_1 + V_2 + V_3 + V_4 \]  \hspace{1cm} (20)

with

\[ V_i = \eta^T P \eta V_i = \int_{t - \tau}^{t} u_{2A}^T(s) Q u_{2A}(s) ds \]

\[ = \frac{\int_{t - \tau}^{t} u_{2A}^T(s) Q u_{2A}(s) ds d\theta}{\int_{t - \tau}^{t} u_{2A}^T(s) Q u_{2A}(s) ds} \]  \hspace{1cm} (21)

In which \( Q = Q \otimes I_n \) such that:

\[ P = \begin{pmatrix} P_1 & P_1^T & P_2 & P_2^T & P_3 \\ P_1^T & P_2 & I_{n \times 1} & P_2^T & P_3 \end{pmatrix} \]  \hspace{1cm} (22)

It is obvious that \( V > 0 \). Derivation of \( V_1, V_2, V_3 \) and \( V_4 \) are calculated along the trajectories of systems (21) as the following representation:

\[ \dot{V}_1 = -2\alpha p_1 u_1^T(t) (r_1^T h(z)) - 2\alpha p_{22} u_{2A}^T(t) R_1^T h(z) \\
-2\beta u_{2A}^T(t) P_1 J u_{2A} - 2\alpha u_1^T(t) P_2 R_1^T h(z) \\
-2\beta u_1^T(t) P_3 J u_{2A} - 2\alpha u_1^T(t) P_3 R_1^T h(z) \\
-2p_{22} u_{2A}^T(t) v_{2A}(t) + 2\gamma u_{2A}^T(t) P_1 J u_{2A} - 2\alpha v_{2A}^T(t) P_2 R_1^T h(z) \\
-2v_{2A}^T P_3 J u_{2A} - 2\alpha v_{2A}^T P_3 R_1^T h(z) + 2\gamma v_{2A}^T(t) P_1 J u_{2A} - 2\alpha v_{2A}^T(t) P_3 R_1^T h(z) \\
\dot{V}_2 = u_{2A}^T(t) Q u_{2A}(t) \\
- u_1^T(t) Q u_{2A}(t) - u_{2A}^T(t - d(t)) Q u_{2A}(t - d(t)) \\
\dot{V}_4 = u_{2A}^T(t) Q u_{2A}(t) - u_1^T(t) u_{2A}^T(t - d(t)) Q u_{2A}(t - d(t)) - u_{2A}^T(t) Q u_{2A}(t) - (1 - d(t)) u_{2A}^T(t - d(t)) Q u_{2A}(t - d(t)) \]  \hspace{1cm} (26)

\( \dot{V}_4 = u_{2A}^T(t) Q u_{2A}(t) - u_1^T(t) u_{2A}^T(t - d(t)) Q u_{2A}(t - d(t)) - u_{2A}^T(t) Q u_{2A}(t) - (1 - d(t)) u_{2A}^T(t - d(t)) Q u_{2A}(t - d(t)) \)

\[ \times Q u_{2A}(t - d(t)) \).

Now, consider the first two expressions in \( \dot{V}_i \), the following relations can be obtained:

\[ \dot{\Theta} = -2\alpha p_1 u_1^T(t) (r_1^T h(z)) - 2\alpha p_{22} u_{2A}^T(t) R_1^T h(z) \]  \hspace{1cm} (27)

\[ -2\beta u_{2A}^T(t) P_1 J u_{2A} - 2\alpha u_1^T(t) P_2 R_1^T h(z) \\
-2\beta u_1^T(t) P_3 J u_{2A} - 2\alpha u_1^T(t) P_3 R_1^T h(z) \]

\[ -2p_{22} u_{2A}^T(t) v_{2A}(t) + 2\gamma u_{2A}^T(t) P_1 J u_{2A} - 2\alpha v_{2A}^T(t) P_2 R_1^T h(z) \\
-2v_{2A}^T P_3 J u_{2A} - 2\alpha v_{2A}^T P_3 R_1^T h(z) + 2\gamma v_{2A}^T(t) P_1 J u_{2A} - 2\alpha v_{2A}^T(t) P_3 R_1^T h(z) \]

\[ \dot{V}_4 = \frac{1}{w_a} u^T(w_a R^T + \frac{1}{w_a} u^T(t - d) w_a^T R^T)} \]  \hspace{1cm} (28)

Where \( u \) refers to \( u(t) \), replacing (28) in (27) and using \( w_a = \frac{1}{w_a} \), \( w_a^T = I_{n \times n} \) results in the equation:

\[ \dot{\Theta} = -2\alpha p_1 u_1^T(t) (r_1^T h(z)) - 2\alpha p_{22} u_{2A}^T(t) R_1^T h(z) \]  \hspace{1cm} (29)

As regards \( F \) is strongly convex and \( \nabla F \) is Lipschitz, the followings hold:

\[ R_1^T h(z) \geq m \hat{z}'(z) = m u^T u \]  \hspace{1cm} (30)

\[ \dot{\hat{z}}(z) \geq m \hat{z}(z) \]  \hspace{1cm} (31)

In which

\[ u^T u = (u^T w_a R^T + \frac{1}{w_a} u^T(t - d) w_a^T R^T)} (Rw_u + w_a w_a^T (t - d) )) \]

Inequality (31) yields to

\[ \dot{\hat{z}}(z) \geq m \hat{z}(z) \geq m (r_1^T h(z)) \]  \hspace{1cm} (32)

So, finally, we have:

\[ \Theta \leq (2\hat{\Theta} - 2\alpha p_1) \hat{z}' h(z) \]  \hspace{1cm} (33)

\[ + 2\alpha p_1 u^T(t - d) w_a^T R^T h(z) \]

\[ - 2\alpha (p_{22} - p_1) u_{2A}^T(t) R_1^T h(z) \]

\[ - 2\alpha (p_{22} - p_1) u_{2A}^T(t) R_1^T h(z) \]

\[ - (r_1^T h(z))^T (r_1^T h(z)) \leq \eta (\hat{\Theta} - 2\alpha p_1) u^T u \]  \hspace{1cm} (33)
\[ +2 \frac{\alpha}{\alpha_i} p_{ii} u_i^T (t-d) \times w_j^T \mathbf{R}^j h(z) \]
\[ -2 \alpha (p_{2i} - p_{ii}) w_i^T \mathbf{R}^j h(z) \]
\[ - \delta(\mathbf{R}, h(z))^T (\mathbf{R}, h(z)) - \delta(\mathbf{v}, h(z))^T (\mathbf{v}, h(z)). \]

Note that we can have the following equation
\[ u(t) = u^T (t-d) w_j^T w_j u + u^T (t-d) w_j^T w_j u (t-d) \]
and be defined \( u = (u_1^T, u_2^T, \cdots, u_n^T) \) in terms of \( u_1^T \) and \( u_2^T \).

\[ \mathbf{w}_j^T \mathbf{w}_k \mathbf{u} = \begin{bmatrix} u_1^T & u_2^T & \cdots & u_n^T \end{bmatrix} \]
\[ = u_1^T a_1 u_1 + u_2^T a_2 u_1 I A - u_2^T A_1 u_1^T A \]
\[ \text{Let } \mathbf{w}_j^T \mathbf{w}_k = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{bmatrix} \text{ then we can obtain} \]
\[ u(t-d) w_j^T \mathbf{w} _k u = u_1^T (t-d) Y_{ij} u_1 (t-d) \]
\[ + u_2^T (t-d) Y_{ij} u_2 (t-d) \]
\[ + u_1^T (t-d) Y_{ij} u_1 \]
\[ \text{Defining } \mathbf{w}_j^T \mathbf{w}_k = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} \text{, we reach to} \]
\[ u(t-d) \mathbf{w}_j^T \mathbf{w}_k u = u_1^T (t-d) Y_{ij} u_1 (t-d) \]
\[ + u_2^T (t-d) Y_{ij} u_2 (t-d) \]
\[ + u_1^T (t-d) Y_{ij} u_1 \]
\[ \text{and, defining } \mathbf{w}_j^T \mathbf{w}_k = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \text{, we have} \]
\[ u(t-d) \mathbf{w}_j^T \mathbf{w}_k u = u_1^T (t-d) g_{ij} u_1 (t-d) \]
\[ + u_2^T (t-d) g_{ij} u_2 (t-d) \]
\[ + u_1^T (t-d) g_{ij} u_1 \]
\[ \text{by parsing } \mathbf{w}_j^T = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \text{ yields to,} \]
\[ 2 \frac{\alpha}{\alpha_i} p_{ii} u_1^T (t-d) \mathbf{w}_j^T \mathbf{R}^j h(z) = \]
\[ 2 \frac{\alpha}{\alpha_i} u_1^T (t-d) k_{11} \mathbf{R}^j h(z) \]
\[ + 2 \frac{\alpha}{\alpha_i} u_2^T (t-d) k_{12} \mathbf{R}^j h(z) \]
\[ + 2 \frac{\alpha}{\alpha_i} u_1^T (t-d) k_{21} \mathbf{R}^j h(z) \]
\[ + 2 \frac{\alpha}{\alpha_i} u_2^T (t-d) k_{22} \mathbf{R}^j h(z) \]

Regarding (33)-(37), upper-bounded of \( \Omega \) can be acquired that ultimately, it will be used to make final LMI.

Additionally, by substituting
\[ \dot{u}_{2, \lambda} = - \alpha \mathbf{R}, h(z) - \beta \mathbf{R}, u_{2, \lambda} (t-d) - v_{2, \lambda} (t) \]
into the first term of \( \dot{V}_i \), we have:
\[ \dot{V}_i = \frac{\alpha}{\alpha_i} p_{ii} u_1^T (t-d) \mathbf{w}_j^T \mathbf{R}^j h(z) = \]
\[ \frac{\alpha}{\alpha_i} u_1^T (t-d) k_{11} \mathbf{R}^j h(z) \]
\[ + 2 \frac{\alpha}{\alpha_i} u_2^T (t-d) k_{12} \mathbf{R}^j h(z) \]
\[ + 2 \frac{\alpha}{\alpha_i} u_1^T (t-d) k_{21} \mathbf{R}^j h(z) \]
\[ + 2 \frac{\alpha}{\alpha_i} u_2^T (t-d) k_{22} \mathbf{R}^j h(z) \]

\[ \text{REFERENCES} \]


Farshad Rahimi received the B.Sc. degree in Robotic Engineering from Hamedan University of Technology, Iran, in 2015, and the M.Sc. degree from the Department of Electrical Engineering, Sahand University of Technology, Tabriz, Iran. He is a faculty member and the Head of the Advanced Modern Control Laboratory, at Sahand University of Technology, Tabriz, Iran. His research interests includes Distributed Optimization, Multi-Agent Coordination, and Time-Delay Systems.