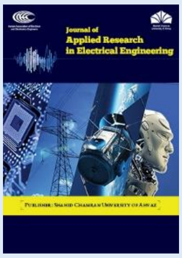
Shahid Chamran University of Ahvaz

Iranian Association of Electrical and Electronics Engineers

Journal of Applied Research in Electrical Engineering

E-ISSN: 2783-2864
P-ISSN: 2717-414X
Homepage: https://jaree.scu.ac.ir/



Research Article

Consensus-Based Algorithm for Distributed Continuous-Time Convex Optimization Over Undirected and Directed Networks

Ehsan Nazemorroaya [1], **Mohsen Shafieirad** [1,*], **Mahdi Majidi** [1], **and Mahdieh Adeli** [2]

[1] *Department of Electrical and Computer Engineering, University of Kashan, Kashan, Iran*

[2] *Electrical Engineering Department, Kermanshah University of Technology, Kermanshah, Iran*

* *Corresponding Author: m.shafieirad@kashanu.ac.ir*

Abstract: In this article, the distributed continuous-time convex Optimization Problem (OP) is investigated over undirected and balanced directed graphs. The cost function of the distributed convex OP is determined as the sum of local convex functions where each of them is known only for one agent. The proposed algorithm consists of two main steps. The first step is a consensus-based scheme which is in combination with the gradient descent method. Employing the Lyapunov theory and LaSalle's invariance principle, the convergence to the Optimal Solution (OS) is analyzed. Moreover, inspired by the average consensus, in the second step the Optimal Value (OV) of the distributed convex OP is calculated. Using consensus concepts, converges to the OV is substantiated in the second step. Therefore, the offered algorithm can calculate the OS and the OV of the distributed convex OP with no need for the strong convexity assumption. Beyond the theoretical findings, the results from simulations are also showcased to demonstrate the efficiency and accuracy of the proposed algorithm.

Keywords: Distributed convex optimization, multi-agent system, consensus, convergence analysis, gradient descent method.

Article history (Please do not write in this space)
Received 16 March 2024; Revised 19 July 2024; Accepted 13 October 2024; Published online 23 October 2024.

How to cite this article
E. Nazemorroaya, M. Shafieirad, M. Majidi, and M. Adeli, "Consensus-Based Algorithm for Distributed Continuous-Time Convex Optimization Over Undirected and Directed Networks," *J. Appl. Res. Electr. Eng.*, Vol. 3, No. 1, pp. 74-82, 2024. DOI: 10.22055/jaree.2024.46401.1113

1. Introduction

Over the last few years, a lot of research has been done on Multi-Agent Systems (MASs) [1-4]. Researchers have shown a keen interest in MASs due to the agents' ability to reach consensus on a value through their mutual interactions [5]. It should be mentioned that this value, named consensus value, is the mean of the initial conditions of agents.

Due to the extensive application of MASs, the design and analysis of distributed algorithms have emerged as a significant area of research [6]. Distributed algorithms achieve the specified objectives by relying solely on local calculations and the exchange of information between each agent and its neighbors the distributed algorithms based on consensus are the predominant ones. The complexities of distributed consensus algorithms are enormous. Therefore, designing an appropriate distributed consensus algorithm is a critical and challenging problem [7]. Algorithms that utilize distributed consensus are widely implemented to solve the distributed Optimization Problem (OP). The objective of the distributed OP is to determine the Optimal Solution (OS) for the cost function through the application of distributed algorithms. It should be noted that the distributed optimization does not need a long-distance communication system and a data fusion center; thus, the load balance for the network becomes better [8]. Also, due to the time-varying communication topology, energy constraints, the large size of the network and privacy problems, distributed optimization methods have received much attention [8-10]. The distributed OP has been used in distributed estimation [8], power distribution systems [11], economic dispatch in power systems [12, 13], federated learning [14] and formation control [15].

In general, the OPs are divided into constrained OP [16, 17] and unconstrained OP [16, 18]. It should be pointed out that the unconstrained OP is discussed in this article. An unconstrained OP is convex if and only if the cost function is convex. The cost function in the distributed convex OP consists of the sum of local convex functions, each of them is known only for one agent. It is worth mentioning that the goal of the distributed convex OP is to minimize the total cost function by using local information and computation.

Subgradient-based approaches are commonly employed to solve the distributed convex OP [19]. In [20], the distributed convex OP under non-smooth and smooth cost functions has been investigated. In the mentioned reference, the communication network has been considered to be undirected. By combining the gradient algorithm and consensus algorithm, a novel distributed algorithm for the distributed convex optimization over the directed graph has been designed in [21]. The designed algorithm has been named as push-pull gradient algorithm because all agents push and pull information (gradient information and the optimization variables) through the communication network. For the distributed convex OP, an asynchronous algorithm has been designed by Zhang and You [22]. In the asynchronous algorithm, each agent iteratively calculates the OS by using the currently available local information, but in the synchronous algorithm, each agent waits for information updates from other agents and then goes to the next iteration of the algorithm. It should be noted that asynchrony causes difficulties in optimization algorithm design. In all the above-mentioned studies, the distributed discrete-time OP has been investigated. On the other hand, the studies about the distributed continuous-time OP are discussed in the next paragraph. It should be mentioned that for solving the distributed continuous-time OP, an algorithm is often designed using control theory such as Lyapunov theory and LaSalle's invariance principle.

In [23], a novel consensus-based algorithm has been designed for the distributed convex OP. In the mentioned reference, the convergence of the designed algorithm to the OS has been guaranteed over the weight-balanced directed graph, by using the Lyapunov theory. For undirected graphs, the distributed convex OP has been studied by Li et al. [24]. Therefore, an event-triggered controller has been designed to solve the OP. It should be noted that the linear MAS with heterogeneous dynamics has been studied in the mentioned reference. In heterogeneous MASs, unlike homogeneous MASs, the agents do not have the same dynamics. Based on the event-triggered mechanism, a distributed optimization algorithm has been designed by Shi et al. [25]. The major advantage of this algorithm is communication resource saving. Inspired by second-order consensus, a novel subgradient-based algorithm for distributed convex OP is presented in [26]. The proposed algorithm can only calculate the OS of the distributed convex OP. In [27], a proposed algorithm addresses the distributed convex OP, and the communication network is assumed to be undirected. Furthermore, by applying the Lyapunov theory, the exponential convergence of the suggested algorithm has been proved with no need for the strong convexity assumption.

The calculation of the Optimal Value (OV) has not been investigated in any of the above-mentioned studies. It should be emphasized that the design of the distributed algorithm for calculating the OV is an important problem with complexity because each agent has access to only a part of the total cost function. Based on the explanations provided, this paper presents an algorithm that integrates the gradient method with the consensus algorithm of the MAS. Two theorems for the convergence of the offered algorithm to the OS and the OV are presented and by application of mathematical tools from control theory and convex optimization they are proved.

The main contribution to the article is listed as follows:

- Achieving higher accuracy than similar algorithms;
- Efficiency of the algorithm for balanced directed graphs;
- Calculating both the OS and the OV.

The subsequent sections of the paper are organized as follows. Section 2 introduces the graph of the MAS and the distributed OP along with several beneficial assumptions and lemmas. Section 3 presents the proposed algorithm and examines its convergence to the OS and OV, utilizing the Lyapunov theory and the consensus concepts. Then, in Section 4, the simulation results are given to show the optimization ability of the offered algorithm. In the end, Section 5 concludes the article.

Notations. $\mathbb{R}^+$ and $\mathbb{R}$ denote the positive real numbers set and the real numbers set $\mathbb{R}^+$ and $\mathbb{R}$, respectively. The Euclidean norm and the Kronecker product are represented by $\|.\|$ and $\otimes$, respectively.

2. Basic Concepts and Problem Statement

This section begins by presenting the communication network model and the required concepts and assumptions for the communication network are stated. Then, the OP is presented and the necessary concepts pertaining to the convex optimization are outlined.

2.1. Communication Network

In this article, to model the communication network, the principles of the MAS and graph theory are employed. To do that, a graph is used to model the communication between the agents. It is assumed that the graph of the MAS is undirected and balanced directed. Thus, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ is defined as a graph that includes the set of nodes $\mathcal{V} = [\mathcal{V}_1, \mathcal{V}_2, \cdots, \mathcal{V}_n]$, the set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ and the adjacency matrix $\mathcal{A} = \{a_{ij}\}_{i,j=1,\ldots,n} \in \mathbb{R}^{n\times n}$ where n is the number of agents. If the agent i is capable of receiving the data sent by the agent j, then $a_{ij} > 0$, otherwise $a_{ij} = 0$.

Definition 1 [28]. A path is defined as a series of nodes connected in such a way that each pair of adjacent nodes is linked by an edge. In graph theory, the path can be directed or undirected.

Definition 2 [28]. An undirected graph is called connected when there is an undirected path between any two distinct nodes.

Definition 3 [28]. In directed graphs, if there is a directed path between any two arbitrary nodes; then, the graph is called to be strongly connected.

An indispensable condition for the consensus of MAS is that the graph is connected or strongly connected [28]. Hence, Assumptions 1 and 2 are presented.

Assumption 1. The graph $\mathcal{G}$ is undirected and connected.

Assumption 2. The graph $\mathcal{G}$ is assumed to be directed and strongly connected.

The definition of the Laplacian matrix corresponding to graph $\mathcal{G}$ is as follows:

$$\mathcal{L} = \{\ell_{i,j}\}_{i,j=1,..,n} \in \mathbb{R}^{n\times n}, \quad \ell_{i,j} = \begin{cases} -a_{ij}, & if\ i \neq j \\ \sum_{j=1}^{n} a_{ij}, & if\ i = j \end{cases} \tag{1}$$

It is apparent that $\mathbf{1}_n = [1,1,\dots,1]^T \in \mathbb{R}^n$ serves as a right eigenvector of the Laplacian matrix; thus, one has $\mathcal{L}\mathbf{1}_n = 0$.

Definition 4 [28]. The graph is termed balanced if $d_{in}(\mathcal{V}_i) = d_{out}(\mathcal{V}_i)$ holds for all nodes where $d_{in}(\mathcal{V}_i) = \sum_{j=1}^{n} a_{ij}$ and $d_{out}(\mathcal{V}_i) = \sum_{j=1}^{n} a_{ji}$.

Assumption 3 [29]. The graph $\mathcal{G}$ is supposed to be balanced.

2.2. Distributed OP Formulation

In this article, the distributed unconstrained OP is investigated. Therefore, the OP is defined as the following equation.

$$\min_{x\in\mathbb{R}^N} f(x) = \sum_{i=1}^{n} f_i(x) \tag{2}$$

where $f(x)$: $\mathbb{R}^N \to \mathbb{R}$ is the total cost function, $x \in \mathbb{R}^N$ is the optimization variable and $f_i(x)$: $\mathbb{R}^N \to \mathbb{R}$ is the local cost function Every local cost function is both convex and differentiable. Each local cost function is exclusively known to a single agent. It seems obvious that the total cost function is convex because it consists of the sum of convex functions.

The vector $x^* \in \mathbb{R}^N$ and the scalar $f^* = f(x^*)$ represent the OS and the OV of the OP in (2), respectively.

Two lemmas are given in the following which will be used in Section 3 in the design and convergence analysis of the offered algorithm.

Lemma 1 [30]. If the following inequality holds for every $x \in \mathbb{R}^N$ and $y \in \mathbb{R}^N$, then the function $f(x)$: $\mathbb{R}^N \to \mathbb{R}$ is convex.

$$(\nabla f(x))^T (y - x) \leq f(y) - f(x) \tag{3}$$

where $\nabla f(x)$: $\mathbb{R}^N \to \mathbb{R}^N$ denotes the gradient of the function $f(x)$.

Lemma 2 [30]. For each convex function $f(x)$: $\mathbb{R}^N \to \mathbb{R}$ and for each $x, y \in \mathbb{R}^N$, the subsequent inequality holds.

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geq 0 \tag{4}$$

Remark 1. If $f(x) - \frac{m}{2}\|x\|_2^2$ is convex, then $f(x)$ is termed strongly convex [20]. In [8], [20], [24] and [25], for the design of the optimization algorithm, it has been assumed that the cost function is strongly convex. Unlike the mentioned references, the strong convexity assumption is not required for the convergence of the offered algorithm in this article.

3. Design and Analysis of the Distributed Optimization Algorithm

In this section, an innovative two-step consensus-based algorithm is introduced that combines calculating OS and OV. Then, the convergence analysis of this algorithm is performed using the concepts of MAS and consensus. First, based on the concept of consensus, the OS is calculated, and the convergence is investigated using the Lyapunov theory. Then, another consensus-based scheme is proposed to calculate the OV and the convergence analysis of this step is performed using the concepts of the MAS and consensus. At the end of this section, the algorithm resulting from the combination of these two steps is fully introduced, and its features are stated. Fig. 1 shows the general scheme of the algorithm.

Step 1: Computing OS

Based on the concept of average consensus, for a MAS with n agents, where the optimization variable of each agent is called $x_i, i = 1, \dots, n$, the following relations is used to compute the OS.

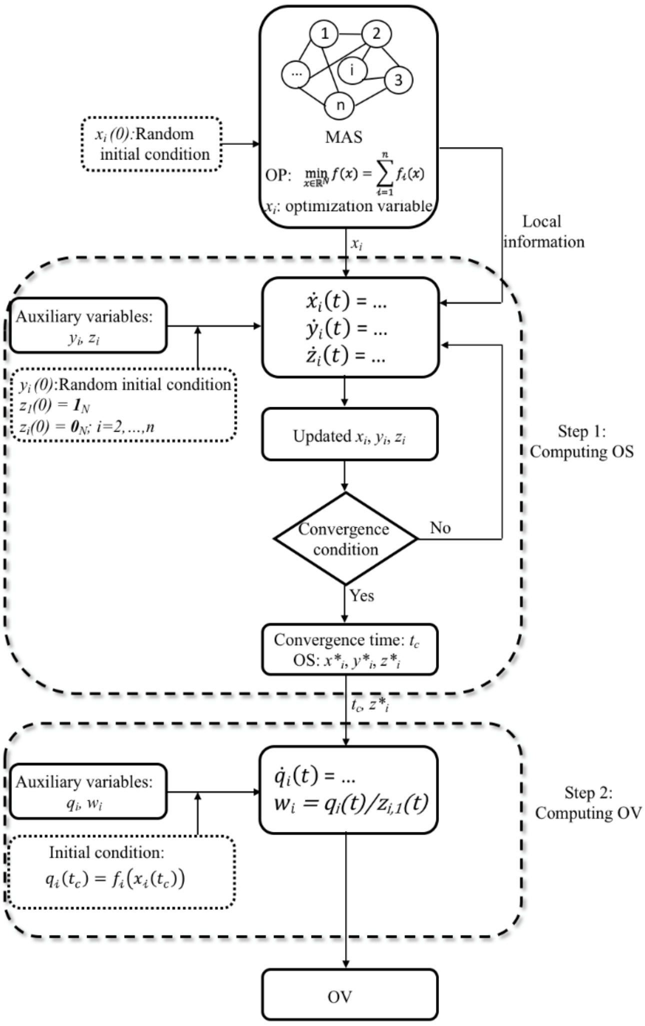


Fig. 1: The method overview.

$$\dot{x}_i(t) = \sum_{j=1}^{n} a_{ij}\left(x_j(t) - x_i(t)\right) + \varpi \sum_{j=1}^{n} a_{ij}\left(y_j(t) - y_i(t)\right) - \varrho g_i\left(x_i(t)\right) \quad (5)$$

$$\dot{y}_i(t) = \varpi x_i(t) \quad (6)$$

$$\dot{z}_i(t) = \sum_{j=1}^{n} a_{ij}\left(z_j(t) - z_i(t)\right) \quad (7)$$

Parameters $\varpi \in R$ and $\varrho \in R^+$ are constant and $g_i\left(x_i(t)\right)$ is the gradient of the function $f_i(x)$ at $x_i(t)$. The initial conditions for the variables $x_i(t) = \left[x_{i,1}(t), x_{i,2}(t), \dots, x_{i,N}(t)\right]^T \in \mathbb{R}^N$ and $y_i(t) = \left[y_{i,1}(t), y_{i,2}(t), \dots, y_{i,N}(t)\right]^T \in \mathbb{R}^N$ are chosen randomly and the initial conditions for the variable $z_i(t) = \left[z_{i,1}(t), z_{i,2}(t), \dots, z_{i,N}(t)\right]^T \in \mathbb{R}^N$ are selected as $z_1(0) = \mathbf{1}_N, z_2(0) = \mathbf{0}_N, \cdots, z_n(0) = \mathbf{0}_N,$ where $\mathbf{1}_N = [1 \quad \dots \quad 1]_{1\times N}^T$ and $\mathbf{0}_N = [0 \quad \dots \quad 0]_{1\times N}^T$.

Remark 2. The variable $z_i(t)$ has no relationship to the variables $x_i(t)$ and $y_i(t)$ because (7) is a first-order consensus for the variable $z_i(t)$. It should be pointed out that in the step 2, the consensus values of variable $z_i(t)$ will be used. Therefore, the variable $z_i(t)$ should be defined in the first step so that it has converged to the consensus values before executing the second step.

Considering (5) and (6) for all agents and using the matrix form, step 1 is written as follows:

$$\dot{X}(t) = -\mathcal{L}_I X(t) - \varpi \mathcal{L}_I Y(t) - \varrho G\left(X(t)\right) \quad (8)$$

$$\dot{Y}(t) = \varpi X(t) \quad (9)$$

where $X(t) = [x_1^T(t), \dots, x_n^T(t)]^T$, $Y(t) = [y_1^T(t), \dots, y_n^T(t)]^T$, $G\left(X(t)\right) = \left[g_1^T\left(x_1(t)\right), \dots, g_n^T\left(x_n(t)\right)\right]^T$ and $\mathcal{L}_I = \mathcal{L} \otimes I_N$.

On the other hand, for the OSs $X^* \in \mathbb{R}^{Nn}$ and $Y^* \in \mathbb{R}^{Nn}$ the following equations can be written. It should be noted that the OS of all agents is collected in the vector X^* which is defined as $X^* = \mathbf{1}_n \otimes x^*$.

$$\dot{X}^* = -\mathcal{L}_I X^* - \varpi \mathcal{L}_I Y^* - \varrho G(X^*) \quad (10)$$

$$\dot{Y}^* = \varpi X^* \quad (11)$$

According to the property $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$, one has

$$\begin{aligned} \mathcal{L}_I X^* &= (\mathcal{L} \otimes I_N)(\mathbf{1}_n \otimes x^*) \\ &= (\mathcal{L}\mathbf{1}_n) \otimes (I_N x^*) \\ &= \mathbf{0}_{Nn} \end{aligned} \quad (12)$$

where $\mathbf{0}_{Nn} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix}_{N\times n}$.

Finally, according to (8)-(12), the convergence error of the variables is given by:

$$\dot{e}_x(t) = -\mathcal{L}_I e_x(t) - \varpi \mathcal{L}_I e_y(t) - \varrho\left(G\left(X(t)\right) - G(X^*)\right)$$

$$\dot{e}_y(t) = \varpi e_x(t) \quad (14)$$

Theorem 1 [26]. Under Assumption 1, and according to Lemma 1 and 2, the convergence of the algorithm defined in (5) and (6) to the OS of the distributed unconstrained OP defined in (2) is guaranteed.

The proof can be followed by [26]. ∎

Step 2: Computing OV

For calculating the OV, the following relations are introduced:

$$\dot{q}_i(t) = \sum_{j=1}^{n} a_{ij}\left(q_j(t) - q_i(t)\right), \qquad \text{for } t \geq t_c \quad (15)$$

$$w_i(t) = \frac{q_i(t)}{z_{i,1}(t)}, \qquad \text{for } t \geq t_c \quad (16)$$

where $q_i \in \mathbb{R}$, $w_i \in \mathbb{R}$ and t_c is the time that all agents converge to OS x^*. In this algorithm, initial conditions for variables $q_i(t)$ are chosen as $q_i(t_c) = f_i\left(x_i(t_c)\right)$. It should be mentioned that $f_i\left(x_i(t_c)\right) = f_i(x^*)$.

It is important to define a method for each agent to calculate the time t_c based on the local information. For each agent, this method is defined as follows:

$$\text{if} \begin{cases} \sum_{j=1}^{n} a_{ij}\left(x_j(t_k) - x_i(t_k)\right) \leq \varepsilon \\ \& \\ x_i(t_k) - x_i(t_{k-1}) \leq \varepsilon \end{cases} \Rightarrow t_k = t_c \quad (17)$$

where ε is a very small positive constant.

Remark 3. It should be noted that at time t_c the differences between the variable $x_i(t)$ and the optimization variable of its neighbors are zero and the value of the variable $x_i(t)$ does not change because it has been converged to the OS. Therefore, in (17), the value of constant ε should be considered zero and inequalities should be replaced by equalities, but due to the computational errors in the simulation software and the communication noise in the practical implementation, the parameter ε is assumed to be a very small positive constant. If the constant ε is selected smaller, then the proposed algorithm has a higher accuracy. For example, $\varepsilon = 10^{-5}$ is a good choice for the simulation.

Theorem 2. Under Assumption 1, the optimization algorithm defined in (15) and (16) converges to the OV of the problem defined in (2).

Proof. Based on the first-order consensus algorithm and according to [28], all agents reach the consensus in the second algorithm and $q_i(t)$ for $i = 1,2, \dots, n$ converges to $\frac{1}{n}\sum_{i=1}^{n} q_i(t_c)$. As mentioned before, the initial conditions are selected as $q_i(t_c) = f_i(x^*)$. Thus, the consensus value of variables $q_i(t)$ for $i = 1,2, \dots, n$ is $\frac{1}{n}\sum_{i=1}^{n} f_i(x^*)$. On the other hand, according to (7), $z_i(t)$ for $i = 1,2, \dots, n$ reach the consensus with the consensus vector $\frac{1}{n}\sum_{i=1}^{n} z_i(0)$ where in the first algorithm, initial conditions are chosen as $z_1(0) =$

$\mathbf{1}_N, z_2(0) = \mathbf{0}_N, \cdots, z_n(0) = \mathbf{0}_N$. Therefore, $z_i(t)$ for $i = 1,2,\ldots,n$ converges to $\frac{1}{n} \times \mathbf{1}_N$ or in other words, the consensus value for $z_{i,1}(t)$ is $\frac{1}{n}$. Finally, according to (16), parameters $w_i(t)$ for $i = 1,2,\ldots,n$ converges to $\sum_{i=1}^{n} f_i(x^*)$. Hence, it is proved that in the second algorithm, all agents reach the OV. ∎

Remark 4. Inspired by [23], by changing the coordinate, Theorem 1 can be proved for balanced directed networks. In addition, the proof of Theorem 2 is also true for balanced directed networks. Therefore, under Assumptions 2 and 3, the algorithm presented in this article is effective for directed networks and the simulation outcomes of Example 2 in Section 4 indicate the correctness of this claim.

Remark 5. In [23] and [27], the distributed continuous-time OP has been investigated over the undirected graph and the balanced directed graph. The algorithm presented in the mentioned references is the same, but different methods have been used to analyze the convergence of the algorithm. It should be emphasized that the algorithm designed in the mentioned references only calculates the OS, while the offered algorithm in this article has the ability to calculate the OV, too. Therefore, the proposed algorithm in this article has an advantage over the algorithm designed in [23] and [27].

Remark 6. In [8], the distributed discrete-time OP has been studied on undirected networks and an algorithm has been presented for calculating the OS and OV. It should be noted that in [8], the OP has been considered as $\min_{x\in\mathbb{R}^N} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$. According to (2), it is clear that the OP considered in this article is more comprehensive.

At the end of this section, a new algorithm is presented in the following (Algorithm 1) by combining the two mentioned algorithms. This algorithm has the ability to calculate the OS and OV.

Algorithm 1. The offered optimization algorithm

1. Each agent initializes the variables $\boldsymbol{x_i(t)}$, $\boldsymbol{y_i(t)}$ and $\boldsymbol{z_i(t)}$.
2. Each agent updates the optimization variable $\boldsymbol{x_i(t)}$ by (5).
3. Each agent updates the variable $\boldsymbol{y_i(t)}$ by (6).
4. Each agent updates the variable $\boldsymbol{z_i(t)}$ by (7).
5. Each agent evaluates the condition (17)1 until this condition is satisfied.
6. **if $\boldsymbol{t \geq t_c}$, then**
7. Each agent updates the variable $\boldsymbol{q_i(t)}$ by (15).
8. Each agent calculates the variable $\boldsymbol{w_i(t)}$ by (16).
9. **end if.**

4. Numerical Simulation

In this section, two numerical examples are simulated by MATLAB software and the outcomes of simulation are given to show the ability of the offered algorithm in the distributed continuous-time convex optimization on undirected and balanced directed graphs.

Example 1. For wireless-sensor networks, a distributed parameter estimation problem is considered as a numerical example. For a sensor network, the parameter estimation problem is defined as [8]

$$\min_{x\in\mathbb{R}} f(x) = \sum_{i=1}^{10} \frac{|x - v_i|^2}{10} \tag{18}$$

where $v = [0.1, 0.2, \ldots, 1]$. In this example, the OS is $x^* = 0.55$ and the OV is $f^* = 0.0825$. The algorithm parameters are selected as

$$\varpi = 2, \qquad \varrho = 4, \qquad \varepsilon = 10^{-5} \tag{19}$$

The adjacency matrix is determined as follows:

$$\mathcal{A} = \begin{bmatrix} 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{20}$$

The simulation outcomes of this example are shown in Figs. 2-4. The variable $x_i(t)$ for all agents is shown in Fig. 2 and the absolute value of convergence errors are shown in Fig. 3. According to Figs. 2 and 3, it is clear that the optimization variable $x_i(t)$ for all agents has been converged to $x^* = 0.55$.

In order to evaluate the offered algori3hm, this example is also simulated using the algorithm presented in [27]. Thus, the outcomes of simulation related to the algorithm designed in [27] are shown in Figs. 5 and 6. First of all, it should be noted that the calculation of the OV has not been studied in [27]. Therefore, our offered algorithm has an advantage due to the calculation of the OV. On the other hand, by comparing Figs. 2 and 5, it can be seen that the offered algorithm has a faster rate of convergence than the algorithm designed in [27].

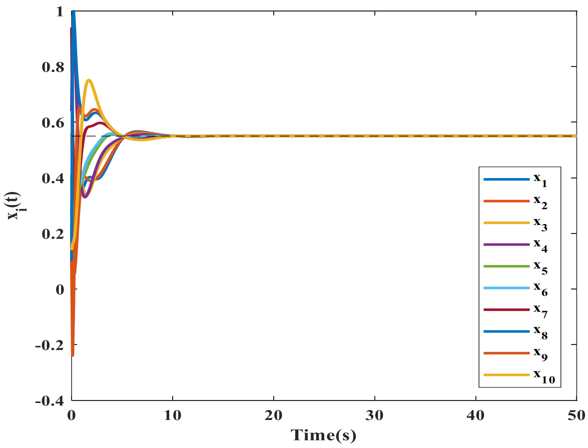


Fig. 2: Trajectories of variable $x_i(t)$.

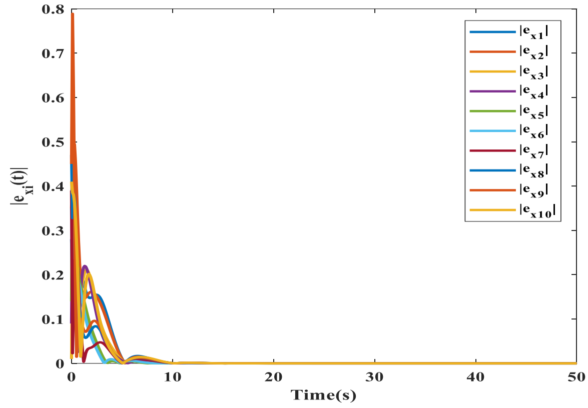


Fig. 3: Convergence errors using the offered algorithm.

In Fig. 4, the OV is shown by the dashed line. As shown in Fig. 4, all the agents have reached a consensus on the OV.

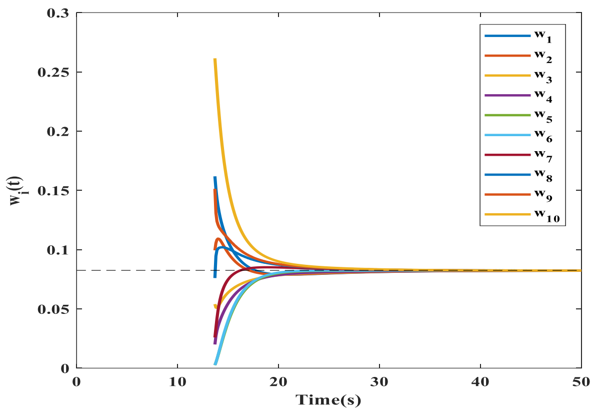


Fig. 4: Trajectories of variable $w_i(t)$.

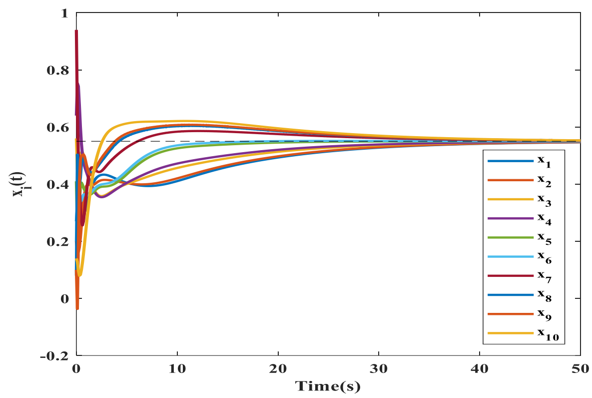


Fig. 5: Trajectories of variable $x_i(t)$ [27].

Example 2. The distributed OP is defined as

$$\min_{x\in\mathbb{R}^N} f(x) = \sum_{i=1}^{5} f_i(x) \tag{21}$$

where $f_i(x)$ for $i = 1, 2, \dots, 5$ are given by

$$f_1(x) = x - 2, \qquad f_2(x) = x^2 + 2,$$
$$f_3(x) = (x+1)^2, \qquad f_4(x) = 4x^4, \tag{22}$$
$$f_5(x) = e^{x-1}$$

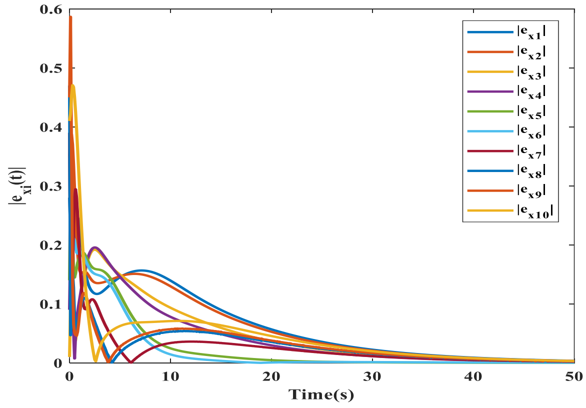


Fig. 6: Convergence errors using the algorithm presented in [27].

The OS and the OV are $x^* = -0.4483$ and $f^* = 0.4536$, respectively. The algorithm parameters are chosen as $\varpi = 3$, $\varrho = 2$ and $\varepsilon = 10^{-5}$.

The communication network model is shown in Fig. 7. In Figs. 8-10, the simulation outcomes of this example are shown. In Fig. 8, the optimization variable $x_i(t)$ for all agents is shown and the dashed line shows the OS.

As shown in Fig. 9, the convergence errors for all agents have been converged to zero. In Fig. 10, the variable $w_i(t)$ is shown. It should be noted that the OV is shown by the dashed line in Fig. 10. In Fig. 10, the variable $w_i(t)$ for all agents has been converged to the OV $f^* = 0.4536$.

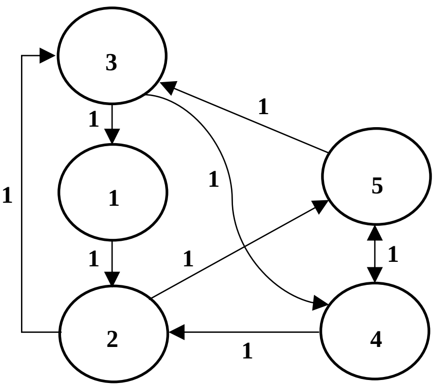


Fig. 7: Communication network model.

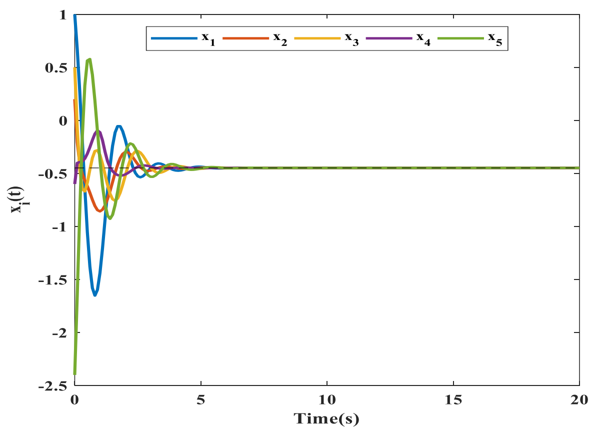


Fig. 8: Trajectories of variable $x_i(t)$.

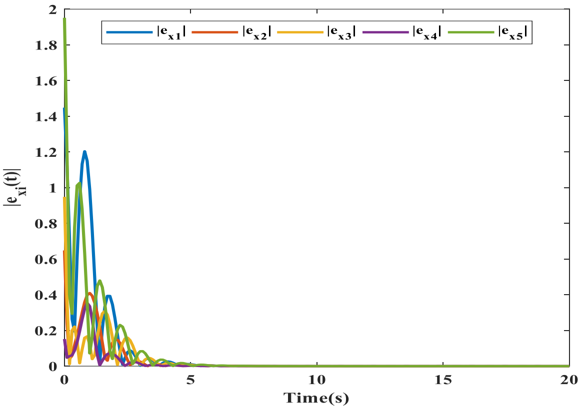


Fig. 9: Convergence errors.

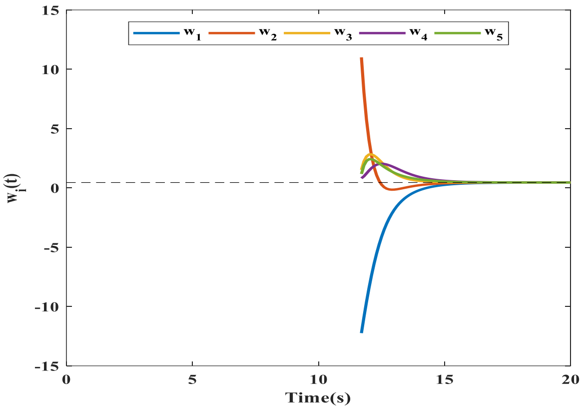


Fig. 10: Trajectories of variable $w_i(t)$.

5. Conclusion

The distributed continuous-time convex OP for undirected and directed balanced networks is the main focus of research in this article. Unlike other references, this article tries to address the calculation of OV in the distributed convex OP. By using the concepts of consensus and gradient descent method, a novel algorithm has been proposed to calculate the OS and the OV. The convergence of the offered algorithm to the OS and the OV has been ensured using the Lyapunov theory, LaSalle's invariance principle and average consensus concepts. Compared to the presented results in [27] where OV is not calculated, the offered algorithm has faster convergence for the same example. The convergence time in [27] is 50 seconds and the convergence time of the offered algorithm is about 10 seconds. Eventually, the offered algorithm provides a new perspective for researchers in solving distributed continuous-time convex OPs.

Credit Authorship Contribution Statement

Ehsan Nazemorroaya: Conceptualization, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Roles/Writing - original draft, Writing - review & editing. **Mohsen Shafieirad**: Methodology, Project administration, Supervision, Validation, Roles/Writing - original draft, Writing - review & editing. **Mahdi Majidi**: Methodology, Project administration, Supervision, Validation, Roles/Writing - original draft, Writing - review & editing. **Mahdieh Adeli:** Methodology, Supervision, Roles/Writing - original draft, Writing - review & editing.

Declaration Of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. The ethical issues; including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, redundancy has been completely observed by the authors.

References

[1] E. Nazemorroaya and M. Hashemi, "Adaptive sliding mode controller design for the bipartite consensus tracking of multi-agent systems with actuator faults and disturbances," *International Journal of Automation and Control*, vol. 16, no. 3-4, pp. 288-302, 2022.

[2] E. Nazemorroaya, M. Shafieirad, and S. A. Zahiripour, "Mean square consensus of heterogeneous multi-agent systems in the presence of actuator faults," *Transactions of the Institute of Measurement and Control*, vol. 45, no. 6, pp. 1158-1168, 2023.

[3] A. H. A. Mehra, M. Shafieirad, and I. Zamani, "Leader-following consensus considering effects of agents on each other via impulsive control: A topology dependent average dwell time approach," *Journal of the Franklin Institute*, vol. 359, no. 16, pp. 8639-8668, 2022.

[4] F. Azadmanesh and R. Ghasemi, "Different Types of Distributed Optimal Leader-Follower Consensus Protocol Design for a Class of High-Order Multi-Agent Systems," *Journal of Applied Research in Electrical Engineering*, 2023.

[5] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215-233, 2007.

[6] M. Adeli, M. Hajatipour, M. J. Yazdanpanah, M. Shafieirad, and H. Hashemi‐Dezaki, "Distributed trust‐based unscented Kalman filter for non‐linear state estimation under cyber‐attacks: The application of manoeuvring target tracking over wireless sensor networks," *IET Control Theory & Applications*, vol. 15, no. 15, pp. 1987-1998, 2021.

[7] W. Zou, Z. Xiang, and C. K. Ahn, "Mean square leader–following consensus of second-order nonlinear multiagent systems with noises and unmodeled dynamics," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 12, pp. 2478-2486, 2018.

[8] H. Li, H. Zhang, Z. Wang, Y. Zhu, and Q. Han, "Distributed consensus-based multi-agent convex optimization via gradient tracking technique," *Journal of the Franklin Institute*, vol. 356, no. 6, pp. 3733-3761, 2019.

[9] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601-615, 2014.

[10] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1245-1260, 2017.

[11] M. Al-Saffar and P. Musilek, "Distributed optimization for distribution grids with stochastic der using multi-agent deep reinforcement learning," *IEEE access*, vol. 9, pp. 63059-63072, 2021.

[12] X. Chang, Y. Xu, H. Sun, and I. Khan, "A distributed robust optimization approach for the economic dispatch of flexible resources," *International Journal of Electrical Power & Energy Systems*, vol. 124, p. 106360, 2021.

[13] N. Mezhoud and M. Amarouayache, "Multi-Objective Optimal Power Flow Based Combined Non-Convex Economic Dispatch with Valve-Point Effects and Emission Using Gravitation Search Algorithm," *Journal of Applied Research in Electrical Engineering*, vol. 2, no. 1, pp. 26-36, 2023.

[14] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1146-1159, 2019.

[15] L. Liu and J. Shan, "Distributed formation control of networked Euler–Lagrange systems with fault diagnosis," *Journal of the Franklin Institute*, vol. 352, no. 3, pp. 952-973, 2015.

[16] E. Twumasi, Y. S. Abdul-Fatawu, and E. A. Frimpong, "Optimal Sizing and Placement of Series Capacitors in Distribution Networks Using Modified Elephant Herding Optimization Algorithm," *Journal of Applied Research in Electrical Engineering*, 2023.

[17] N. Song *et al.*, "Design and optimization of halbach permanent magnet array with rectangle section and trapezoid section," *International Journal of Engineering*, vol. 34, no. 11, pp. 2379-2386, 2021.

[18] P. Maghzi, M. Mohammadi, S. Pasandideh, and B. Naderi, "Operating room scheduling optimization based on a fuzzy uncertainty approach and metaheuristic algorithms," *International Journal of Engineering*, vol. 35, no. 2, pp. 258-275, 2022.

[19] C. Gu, L. Jiang, J. Li, and Z. Wu, "Privacy-Preserving Dual Stochastic Push-Sum Algorithm for Distributed Constrained Optimization," *Journal of Optimization Theory and Applications*, vol. 197, no. 1, pp. 22-50, 2023.

[20] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié, "Optimal convergence rates for convex distributed optimization in networks," *Journal of Machine Learning Research*, vol. 20, pp. 1-31, 2019.

[21] S. Pu, W. Shi, J. Xu, and A. Nedić, "Push–pull gradient methods for distributed optimization in networks," *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 1-16, 2020.

[22] J. Zhang and K. You, "AsySPA: An exact asynchronous algorithm for convex optimization over digraphs," *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2494-2509, 2019.

[23] B. Gharesifard and J. Cortés, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 781-786, 2013.

[24] Z. Li, Z. Wu, Z. Li, and Z. Ding, "Distributed optimal coordination for heterogeneous linear multiagent systems with event-triggered mechanisms," *IEEE Transactions on Automatic Control*, vol. 65, no. 4, pp. 1763-1770, 2019.

[25] X. Shi, Z. Lin, R. Zheng, and X. Wang, "Distributed dynamic event-triggered algorithm with positive minimum inter-event time for convex optimisation problem," *International Journal of Control*, vol. 95, no. 5, pp. 1363-1370, 2022.

[26] E. Nazemorroaya, M. Shafieirad, and M. Majidi, "Consensus-based algorithm for distributed convex optimization," in *4th Conference on Computational Algebra, Computational Number Theory and Applications*, 2023.

[27] S. Liang and G. Yin, "Exponential convergence of distributed primal–dual convex optimization algorithm without strong convexity," *Automatica*, vol. 105, pp. 298-306, 2019.

[28] F. L. Lewis, H. Zhang, K. Hengster-Movric, and A. Das, *Cooperative control of multi-agent systems: optimal and adaptive design approaches*. Springer Science & Business Media, 2013.

[29] X. Meng and Q. Liu, "A consensus algorithm based on multi-agent system with state noise and gradient disturbance for distributed convex optimization," *Neurocomputing*, vol. 519, pp. 148-157, 2023.

[30] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

BIOGRAPHY

Ehsan Nazemorroaya was born in Isfahan, Iran, in 1996. He received the B.Sc degree in Electrical Engineering from the Department of Electrical Engineering, Arak University of Technology, Arak, Iran, in 2018. Also, he received the M.Sc. degree in Electrical Engineering from the Department of Electrical Engineering, Islamic Azad University, Najafabad Branch, Najafabad, Iran, in 2020. Currently, he is a Ph.D. candidate in Electrical Engineering at University of Kashan, Iran. His research interests include multi-agent systems, optimization, fractional-order systems, adaptive control, and stochastic systems. (Email: ehsan.nzm74@gmail.com)

Mohsen Shafieirad received a B.Sc. degree in control engineering from Isfahan University of Technology, Iran, in 2005. Also, he received the M.S. and Ph.D. degrees in control engineering from Amirkabir University of Technology, Iran, in 2007 and 2013, respectively. He is currently an assistant professor of Control at the University of Kashan, Iran. M.Shafieirad overall research output has culminated in more than 50 publications. His research interests include multi-agent systems, biological systems, system identification, ad-hoc/sensor networks, and multi-dimensional systems. (Email: m.shafieirad@kashanu.ac.ir)

Mahdi Majidi received the B.Sc. degree in electrical engineering from Isfahan University of Technology, Isfahan, Iran, in 2004, and the M.S. and Ph.D. degrees in electrical engineering from Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, in 2007 and 2014, respectively. In 2012, he joined the Communications and Networks Laboratory, Department of Electrical and Computer Engineering, National University of Singapore (NUS), Singapore, as a Visiting Ph.D. Student. In 2015, he worked as a researcher at the Iran Telecommunication Research Center (ITRC). Since 2016, he is an assistant professor of the Department of Electrical and Computer Engineering, University of Kashan, Iran. Also, since 2023, he works as a senior communication system designer for the Qualinx BV in Netherlands. His research interests include numerical and analytical optimization methods, Intelligent reflecting surfaces, machine learning, wireless transceiver design, LPWAN, and GNSS. (Email: m.majidi@kashanu.ac.ir)

Mahdieh Adeli received the B.Sc. and M.S. degrees in control engineering from Imam Khomeini International University, Iran, in 2008 and 2012, respectively. Also, she received a Ph.D. degree in control engineering from Kashan University, Iran, in 2022. She is currently an assistant professor of Control at the Kermanshah University of Technology, Iran. Her research interests include optimization, cyber-physical systems, and distributed estimation. (Email: m.adeli@kut.ac.ir)