mathematics

MDPI

Article

R-DOCO: Resilient Distributed Online Convex Optimization Against Adversarial Attacks

Zhixiang Kong [1], Huajian Xu [2] and Chengsheng Pan [1,3,*]

[1] School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China; 218110010211@njust.edu.cn
[2] School of Software, Nanjing University of Information Science and Technology, Nanjing 210044, China; 202212200039@nuist.edu.cn
[3] School of Electronics and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China
* Correspondence: 003150@nuist.edu.cn; Tel.: +86-15940934666

Abstract: This paper addresses the problem of distributed constrained optimization in a multi-agent system where some agents may deviate from the prescribed update rules due to failures or malicious adversarial attacks. The objective is to minimize the collective cost of the unattacked agents while respecting the constraint limitations. To tackle this, we propose a resilient distributed projected gradient descent algorithm for online optimization that achieves sublinear individual regret, defined as the difference between the online and offline solutions. Additionally, we extend the cost function from convex combinations to more general distributed optimization scenarios. The proposed algorithm demonstrates resilience under adversarial conditions, allowing it to handle an unknown number of adversarial nodes while maintaining performance. Compared to existing methods, this approach offers a robust solution to adversarial attacks in constrained distributed optimization problems.

Keywords: distributed projected gradient descent; resilient online optimization; adversarial attack; multi-agent systems

MSC: 68M14; 90C25



Citation: Kong, Z.; Xu, H.; Pan, C. R-DOCO: Resilient Distributed Online Convex Optimization Against Adversarial Attacks. *Mathematics* **2024**, *12*, 3439. https://doi.org/10.3390/math12213439

Academic Editor: Duxin Chen

Received: 2 October 2024
Revised: 29 October 2024
Accepted: 1 November 2024
Published: 3 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Many problems in cyber–physical systems (CPSs) with a large number of interconnected subsystems (agents) can be posed in the framework of distributed optimization, the objective of which is to minimize a global objective function collaboratively [1]. Denoting n and Ω as the number of agents and the closed convex set over the network, respectively, the standard distributed optimization problem with projection constraint can be described as follows:

$$\min_{x\in\Omega}\frac{1}{n}\sum_{i=1}^{n}f_i(x), \tag{1}$$

where $f_i(\cdot)$ is the local objective function of node i and the exact explicit expression is known only to itself. The goal of problem (1) is to minimize the sum of all local objective functions, while all local variables x are consistent within the closed convex set constraint [2,3].

Due to the mechanism of independent information processes and instant communication among agents, external adversarial cyber-attacks against distributed CPSs become exceptionally easy [4,5]. Adversarial attacks are currently one of the most significant threats to CPSs, as they might render existing distributed algorithms vulnerable or even invalid, resulting in system damage and even paralysis [6,7]. Therefore, it is imperative to design

algorithms to resist adversarial attacks or to make the system more resilient. Most of the current distributed optimization algorithms against adversarial attacks are based on outlier filtering rules. A weaker version of the distributed optimization problem formed as a convex combination of local cost functions was defined and a variant of the gradient descent method was proposed in [8] to make the system fault-tolerant. Based on the above weaker version of the distributed optimization problem, an online style of distributed adversarial optimization was put forward in [9], with the assumption that the eigenvectors of the transition matrix are equal to the convex combination coefficients, which is hardly possible due to the time-varying characteristics of adversarial nodes. A new set of outlier-filtering rules, named local filtering dynamics, was presented to guarantee the non-adversarial nodes converge to the convex hull of the minimizers in [10], while the maximum number of adversarial nodes had to be limited to a certain range. Moreover, the RDO-T algorithm endowing a certain connected dominating set (CDS) of nodes with high security was proposed in [11], for when the number of adversarial attacks is large and unknown. Although these above works are able to obtain optimal solutions in the presence of adversarial attacks, they all deal with unconstrained distributed optimization problems. In order to tackle the weaker version of the distributed optimization problem against adversarial attacks with the closed convex set constraint, a collaborative projected gradient descent method was proposed in [12]. In addition, a Byzantine-resilient distributed bandit online convex optimization algorithm named BR-DBOCO that can tolerate up to $\mathcal{B}$ Byzantine agents was developed in [13].

In practice, the objective function of distributed optimization often changes over time, and these changes are unpredictable [14–16]. Therefore, distributed online optimization is of wide practical interest. Based on the push-sum algorithm, the first-order and one-point zeroth-order gradient methods for the unconstrained distributed online optimization problem were presented in [14,17], respectively. Further, to handle the constrained distributed online optimization problem, a family of algorithms based on projections and Lagrange multipliers are studied in [18–21]. However, a constrained distributed online optimization algorithm against adversarial attacks has not been proposed so far.

Therefore, in this paper, we are motivated to design a resilient distributed online convex optimization named R-DOCO, which is based on the projected gradient descent method, for constrained distributed optimization against adversarial attacks. Compared with the resilient distributed optimization algorithms in [8–11], an online version of resilient distributed optimization algorithms is proposed for the constrained distributed optimization problem against adversarial attacks with an arbitrarily large and unknown number of adversarial nodes. Furthermore, we also relax the form of the cost function from convex combination to the general distributed optimization problem.

The remainder of this paper is organized as follows. We begin in Section 2, where we describe the network model and the problem formulation. Section 3 is devoted to outlining the R-DOCO algorithm and giving the properties of the transition matrix. Section 4 states the main convergence results with the key proof. Numerical experiments are presented in Section 5, following which we conclude this study in Section 6.

We use boldface to distinguish between vectors and scalars or vectors of different dimensions. We consider the Euclidean inner product endowed with $\langle \cdot, \cdot \rangle$ and the Euclidean norm $\|x\| = \sqrt{\langle x, x \rangle}$. We use $\mathbf{1}$ to denote a vector of ones, and I for a unit matrix. $P_{\Omega}[x]$ is denoted as the projection of the point x on the set Ω.

2. Network Model and Problem Formulation

2.1. Network Model

Consider a network modeled by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of n agents and $\mathcal{E} = \subset \mathcal{V} \times \mathcal{V}$ is the set of edges, consisting of $n(n \geq 3)$ agents. For the sake of comprehension, agent and node are used interchangeably in the following context. Denote the neighbor set of node i as $N_i = \{j \in \mathcal{V} \mid (j, i) \in \mathcal{E}\}$ and $(i, j) \in \mathcal{E} \Leftrightarrow (j, i) \in \mathcal{E}$ since the undirected graph is $\mathcal{G}$.

The nodes are divided into three types: trusted nodes, normal nodes, and adversarial nodes [11]. Specifically, the trusted nodes have high security, where they will not crash or be compromised by the attacker. The normal nodes may crash or be compromised by the attacker. The adversarial nodes contain malicious and Byzantine nodes that are aware of the global network topology, while they are also acquainted with the update rules of the normal nodes, which can be readily accomplished by using a latent probe over a length of time [22]. If an adversarial node delivers an identical value to all of its out-neighbors at each time step, we call it malicious. In contrast, if an adversarial node delivers different values to different neighbors at each time step, we call it Byzantine [10].

Suppose that each node can recognize the trusted nodes among their neighbors. The sets of trusted nodes, normal nodes, and adversarial nodes are denoted by V_t, V_n, and V_a, respectively. Let $n_t = |V_t|$, $n_n = |V_n|$, and $n_a = |V_a|$ represent their cardinalities, respectively. To simplify the notation, define the set of all the trusted and normal nodes as $n_0 = n_t + n_n$. Different from the topological network without adversarial nodes, let $A \in \mathbb{R}^{n_0 \times n_0}$ be the adjacency matrix consisting of all normal and trusted nodes, where the element $a_{ij} = 1$ if and only if $(i, j) \in \mathcal{E}$, and $a_{ij} = 0$ otherwise. The CDS composed of non-adversarial nodes is essential in the distributed algorithm design against adversarial nodes. The definition of CDS is given as follows.

Definition 1 ([23]). *A set C of graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a CDS if each node which does not belong to C has at least one neighbor in C and all nodes in C form a connected graph.*

Definition 2 ([23]). *$\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ is the subgraph of $\mathcal{G}$, where $\mathcal{V}_1 = V_t \cup V_n$ and $\mathcal{E}_1 \in \mathcal{E}$ includes all edges connecting trusted nodes and all directed edges from trusted nodes to neighboring normal ones. Denote the diameter of graph $\mathcal{G}_1$ as d.*

Supported by the above definition, the following assumption about the network structure for resilient distributed optimization is given.

Assumption 1. *Trusted agents induce a CDS of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.*

Remark 1. *Assumption 1 is particularly relevant in the context of dynamic or large-scale networks, where maintaining efficient communication and connectivity is crucial. The CDS property ensures that every node not included in the set has at least one neighbor in the CDS, which facilitates effective coverage and communication. However, in dynamic environments where nodes frequently join or leave, the ability to continuously maintain a CDS can be challenging. For instance, the mobility of nodes or the introduction of new agents may disrupt existing connections, necessitating a robust mechanism for updating the CDS in real-time to preserve its connectivity.*

2.2. Problem Formulation

The attacker selects certain normal nodes to be compromised so that they become adversarial nodes, which can update their states arbitrarily and broadcast different states to different neighbors. Consider the worst-case scenario in order to endure such attacks, i.e., adversarial nodes alter their local cost functions and follow the iteration protocol like normal nodes and keep communication with all nodes synchronized. In this case, the optimization objective of the overall system is transformed into a sum consisting of all the non-adversarial nodes' local objective functions, i.e., the distributed optimization problem (1) takes the following form:

$$\min_{x \in \Omega} \frac{1}{n_0} \sum_{i \in V_t \cup V_n} f_i(x). \tag{2}$$

Problem (2) cannot be solved perfectly in a distributed manner, i.e., the best solution cannot be attained in a completely distributed way, since the normal or trusted nodes are

unable to detect and identify the compromised nodes through local interactions if each node maintains the confidentiality of its local cost function [8].

The distributed online optimization of this paper differs from the general online optimization form due to the presence of adversarial nodes. The evaluation index for online optimization performance is the regret function, which is defined as the difference between the accumulated cost over time and the cost incurred by the best fixed decision when all functions are known in advance [18,20,24]. Due to the presence of adversarial nodes, the regret for agent j, corresponding to problem (2) from $t = 1$ to T, where $T \in \mathbb{Z}^+$ is the time horizon, is presented as follows:

$$Reg_T^j = \sum_{t=1}^{T} \sum_{i=1}^{n_0} f_t^i(x_j(t)) - \sum_{t=1}^{T} \sum_{i=1}^{n_0} f_t^i(z^\star), \tag{3}$$

where $z^\star \in arg\min_{z\in\Omega} \sum_{t=1}^{T} \sum_{i=1}^{n_0} f_t^i(z)$ is the best fixed offline decision. Throughout this paper, we assume that the minimizer set is nonempty. The objective here is to design a strategy for the compromised multi-agent system so that it achieves a regret that is sublinear in T , that is, $\limsup_{T\to\infty} Reg_T^j/T = 0$, which guarantees that the average regret over time goes to zero [24]. Further, we give the following assumption for the local objective functions, which is common in distributed optimization.

Assumption 2. *Each local objective function satisfies the following assumptions:*

(1) $f_t^i(x)$ *is* μ*-strongly convex and continuously differentiable;* $\forall i \in \mathcal{V}$, $t \in \{1, \dots T\}$.

(2) $f_t^i(x)$ *is Lipschitz continuous, i.e.,* $\|g_i(t)\| \le L$, $\forall i \in \mathcal{V}$, $t \in \{1, \dots T\}$, *where* $g_i(t)$ *denotes the gradient of* $f_t^i(x)$ *and* L *is a positive constant, respectively.*

3. R-DOCO Algorithm

3.1. Algorithm Design

In this subsection, we provide an R-DOCO algorithm in which each normal node only utilizes its neighbor's states, restricted by neighboring trusted nodes and itself, to defeat adversarial attacks. The R-DOCO is divided into two main steps, i.e., the filter step and protocol updates.

Firstly, According to Algorithm 1, each node i identifies the set of all neighboring trusted nodes, i.e., $T_i = \{j \in N_i, j \in V_t\}$, after receiving the messages from its neighbors.

Algorithm 1: R-DOCO: Resilient Distributed Online Convex Optimization

Input: Each node i initializes $x_i(0)$ randomly.

for $t = 1$ *to* T **do**

  Filter Step: Each node i obtains the set $R_i(t)$, i.e.,
  $R_i(t) = \{j \mid x_i^m(t) \le x_j(t) \le x_i^M(t), j \in N_i \cup \{i\}\}$;
  Protocol Updates: Each node i updates $x_i(t+1)$ according to (4):

$$d_i(t) = \frac{1}{|R_i(t)|} \sum_{j\in R_i(t)} x_j(t), \tag{4a}$$

$$x_i(t+1) = P_\Omega[d_i(t) - \alpha(t) g_i(t)], \tag{4b}$$

  where $\alpha(t)$ is the step size and $|R_i(t)|$ denotes the cardinality of $R_i(t)$.

end

Output: $x_i(t), \forall i, j \in V_t \cup V_n$.

Next, for any $j \in T_i$, node i sorts $x_j(t)$ to yield the maximum and minimum states represented by $x_i^M(t) = \max\{x_j(t) \mid j \in T_i \cup \{i\}\}$ and $x_i^m(t) = \min\{x_j(t) \mid j \in T_i \cup \{i\}\}$, respectively. Then, node i iteratively updates $x_i(t)$ based on (4) with the non-negative learning rate $\alpha(t)$ at each time t. Meanwhile, the learning rate $\alpha(t)$ decays with time t

and satisfies $\sum_{t=0}^{\infty}\alpha(t)=\infty$ and $\sum_{t=0}^{\infty}\alpha^2(t)<\infty$, which can guarantee that nodes' states, at steady state, converge to the same optimizer [2]. The projection operator method is used in (4b), where $P_\Omega[x]$ is denoted as the projection of the point x on the closed convex set Ω, i.e.,

$$P_\Omega[x]=arg\min_{y\in\Omega}|x-y|.$$

The above projection has inequality and non-expansiveness properties, i.e., for any x,

$$(P_\Omega[x]-x)^{\mathsf{T}}(y-P_\Omega[x])\geq 0,\forall y\in\Omega, \tag{5}$$
$$(P_\Omega[x]-P_\Omega[y])\leq|x-y|,\forall x,y. \tag{6}$$

The properties listed in the following lemma are also used.

Lemma 1. *Let Ω be a nonempty closed convex set in $\mathbb{R}$. Then, for any $x\in\mathbb{R}$ and for all $y\in\Omega$, we have*

$$(P_\Omega[x]-x)^{\mathsf{T}}(x-y)\leq-|P_\Omega[x]-x|^2, \tag{7}$$
$$|P_\Omega[x]-y|^2\leq|x-y|^2-|P_\Omega[x]-x|^2. \tag{8}$$

3.2. Transition Matrix and Backward Product

This subsection states the properties of the transition matrix and the auxiliary variable that is essential to analyze convergence for R-DOCO. Firstly, define $e_i(t)$ as follows:

$$e_i(t)=P_\Omega[d_i(t)-\alpha(t)g_i(t)]-(d_i(t)-\alpha(t)g_i(t)). \tag{9}$$

Then, the algorithm (4) can be rewritten as

$$x_i(t+1)=d_i(t)-\alpha(t)g_i(t)+e_i(t). \tag{10}$$

The compact form of (10) can be represented as

$$\mathbf{x}(t+1)=\mathbf{M}(t)\mathbf{x}(t)-\alpha(t)\mathbf{g}(t)+\mathbf{e}(t), \tag{11}$$

where $\mathbf{M}(t)$ is the transition matrix at time t. The construction of $\mathbf{M}(t)$ satisfies the requirements of [25], and can obtain the following properties.

Lemma 2. *If Assumption 1 holds, under R-DOCO, there exists $\mathbf{M}(t)=\left[\mathbf{M}_{ij}(t)\right]\in\mathbb{R}^{n_0\times n_0}$ for all t and $i,j\in V_t\cup V_n$, such that*

(1) $\mathbf{M}(t)$ *is a row stochastic matrix, i.e.,* $\sum_{j=1}^{n_0}\mathbf{M}_{ij}(t)=1$;

(2) $\mathbf{M}_{ij}(t)\neq 0$ *if* $(i,j)\in\mathcal{E}_1$ *or* $i=j$;

(3) $\forall\mathbf{M}_{ij}(t)\neq 0,\mathbf{M}_{ij}(t)\geq\gamma=\frac{1}{\max\{|N_i|\}+1}$.

It should be noted that the transition matrix $\mathbf{M}(t)$ can be thought of as a weight matrix if there no attacker invades. Thus, (11) can be further expanded under the transition matrix as follows:

$$\begin{aligned}\mathbf{x}(t+1)&=\mathbf{M}(t)\mathbf{x}(t)-\alpha(t)\mathbf{g}(t)+\mathbf{e}(t)\\&=(\mathbf{M}(t)\mathbf{M}(t-1)\cdots\mathbf{M}(0))\mathbf{x}(0)\\&\quad-\sum_{k=1}^{t+1}(\mathbf{M}(t)\mathbf{M}(t-1)\cdots\mathbf{M}(k))\alpha(k-1)\mathbf{g}(k-1)\\&\quad+\sum_{k=1}^{t+1}(\mathbf{M}(t)\mathbf{M}(t-1)\cdots\mathbf{M}(k))\mathbf{e}(t)\end{aligned}$$

$$= \Phi(t,0)\mathbf{x}(0) - \sum_{k=1}^{t+1} \Phi(t,k)\alpha(k-1)\mathbf{g}(k-1) + \sum_{k=1}^{t+1} \Phi(t,k)\mathbf{e}(k-1), \tag{12}$$

where $\Phi(t,k) = \mathbf{M}(t)\mathbf{M}(t-1)\cdots\mathbf{M}(k), \forall k \leq t+1$ is denoted as the backward product, with $\Phi(t,t) = \mathbf{M}(t)$ and $\Phi(t,t+1) = I_{n_0}$. We can directly obtain the following lemma for the backward product via [8,11,26].

Lemma 3. *If Assumption 1 holds, under R-DOCO, $\Phi(t,k)$ has the following properties for all $k \geq t$.*

(1) For $\Phi(t,k)$, it holds that $\lim_{t\to\infty} \Phi(t,k) = \mathbf{1}\theta^{\mathrm{T}}(k)$, where $\theta(k)$ is a stochastic vector dependent on k and elements in $\theta(k)$ all satisfy $\theta_i(k) \leq 1$;

(2) For any $\Phi(t,k)$, $|\Phi(t,k) - \theta_i(k)| \leq C(1-\gamma)^{t-k}$, where $C > 0$ and $\gamma \in (0,1]$;

(3) For any fixed k, n_2 nonzero entries in $\theta(k)$ are lower-bounded by γ^{n_0}, i.e., there are n_2 nonzero entries $i \in \{1,\ldots,n_0\}$ such that $\theta(k) \geq \gamma^{n_0}$.

Through some properties of the above transition matrix and its backward product, we transform the analysis of distributed optimization with adversarial attacks into the time-series mode. Nevertheless, our goal is to obtain the sublinear convergence of the network regret and agent's regret, respectively. Before all convergence results are given, we need to give the auxiliary variable $y(t)$, which represents the local variable $x_i(t+1)$ will converge to. The main idea of the following analysis is inspired by [11,12]. Thus, R-DOCO is based on the projected gradient descent method against different Byzantine adversarial attacks.

Assuming that for $t \geq \bar{t}, g_i(t) = 0, i \in V_t \cup V_n$, we can obtain that from (12)

$$\begin{aligned}
&\mathbf{x}(t+\bar{t}) \\
&= \Phi(t,\bar{t}-1)\mathbf{x}(\bar{t}-1) + \sum_{k=1}^{\bar{t}} \Phi(t,k)\mathbf{e}(k-1) - \sum_{k=1}^{\bar{t}} \Phi(t,k)\alpha(k-1)\mathbf{g}(k-1) \\
&= \Phi(t,\bar{t})\Big[\Phi(\bar{t}-1,0)\mathbf{x}(0) + \sum_{k=1}^{\bar{t}} \Phi(\bar{t}-1,k)\mathbf{e}(k-1) \\
&- \sum_{k=1}^{\bar{t}} \Phi(\bar{t}-1,k)\alpha(k-1)\mathbf{g}(k-1)\Big] = \Phi(t,\bar{t})\mathbf{x}(\bar{t}).
\end{aligned} \tag{13}$$

As $t \to \infty$, we have

$$\begin{aligned}
&\lim_{t\to\infty} \mathbf{x}(t+\bar{t}) \\
&= \lim_{t\to\infty} \Big(\Phi(t,0)\mathbf{x}(0) + \sum_{k=1}^{\bar{t}} \Phi(t,k)\mathbf{e}(k-1) - \sum_{k=1}^{\bar{t}} \Phi(t,k)\alpha(k-1)\mathbf{g}(k-1)\Big) \\
&= \lim_{t\to\infty} \Phi(t,0)\mathbf{x}(0) + \sum_{k=1}^{\bar{t}} \lim_{t\to\infty} \Phi(t,k)\mathbf{e}(k-1) \\
&- \sum_{k=1}^{\bar{t}} \lim_{t\to\infty} \Phi(t,k)\alpha(k-1)\mathbf{g}(k-1).
\end{aligned} \tag{14}$$

From Lemma 3, (14) can be further derived as

$$\begin{aligned}
&\lim_{t\to\infty} \mathbf{x}(t+\bar{t}) \\
&= \mathbf{1}\theta^{\mathrm{T}}(0)\mathbf{x}(0) + \sum_{k=1}^{\bar{t}} \mathbf{1}\theta^{\mathrm{T}}(0)\mathbf{e}(k-1) - \sum_{k=1}^{\bar{t}} \mathbf{1}\theta^{\mathrm{T}}(0)\alpha(k-1)\mathbf{g}(k-1)
\end{aligned}$$

$$= \Big(\langle\theta(0), \mathbf{x}(0)\rangle + \sum_{k=1}^{\bar{t}} \langle\theta(0), \mathbf{e}(k-1)\rangle - \sum_{k=1}^{\bar{t}} \alpha(k-1)\langle\theta(0), \mathbf{g}(k-1)\rangle\Big)\mathbf{1}. \tag{15}$$

Let $\mathbf{y}(\bar{t})$ denote the limiting vector of $\mathbf{x}(t+\bar{t})$, which is a constant related to $\mathbf{x}(0)$. Since all entries in the limiting vector are identical, $\mathbf{y}(\bar{t}) = [y(\bar{t}), \ldots, y(\bar{t})]^{\mathrm{T}}$. Thus, from (15), we have

$$\begin{aligned} y(\bar{t}+1) &= \langle\theta(0), \mathbf{x}(0)\rangle - \sum_{k=1}^{\bar{t}+1} \alpha(k-1)\langle\theta(k), \mathbf{g}(k-1)\rangle + \sum_{k=1}^{\bar{t}+1} \langle\theta(k), \mathbf{e}(k-1)\rangle \\ &= y(\bar{t}) + \langle\theta(\bar{t}+1), \mathbf{e}(\bar{t})\rangle - \alpha(\bar{t})\langle\theta(\bar{t}+1), \mathbf{g}(\bar{t})\rangle. \end{aligned} \tag{16}$$

Since the subsequent content has no effect on the relationship between t and $\bar{t}$, we use t to replace $\bar{t}$, yielding

$$y(t+1) = y(t) - \alpha(t)\langle\theta(t+1), \mathbf{g}(t)\rangle + \langle\theta(t+1), \mathbf{e}(t)\rangle. \tag{17}$$

In the following section, we provide $y(k)$ to which the local variable $x_i(k)$ will converge, and then, prove that the individual agent's regret has a sublinear rate.

4. Main Results

In this section, we show how the R-DOCO algorithm (4) can be used to bound the individual regret defined in (3). The following theorem is the main result of this paper.

Theorem 1. (Agent's Regret) *Suppose Assumptions 1 and 2 hold, the agents use R-DOCO to generate the sequence* $\{x_i(t)\}$, $i \in \{1, 2, \ldots, n_0\}$, *with step size* $\alpha(t) = 1/(n_0\mu t\gamma^{n_0})$. *Then, for each* $j \in n_0$, *we have*

$$Reg_T^j \leq H_1 + H_2(1 + \ln T) + H_3(1 + \ln(T))^2. \tag{18}$$

where

$$\begin{aligned} H_1 &= \frac{n_0\mu}{2}\|y(0) - z^\star\| + n_0CL\frac{3\gamma^{n_0}+1}{\gamma^{n_0+1}}\|x_i(0)\| \\ &\quad + \frac{\mu C n_0}{\gamma}\sum_{i=1}^{n_0}\|x_i(0)\|\|y(0) - z^\star\|, \\ H_2 &= 2n_0L\frac{C+2\gamma}{\gamma^{n_0+1}}L\|y(0) - z^\star\| + \frac{2n_0LC}{\gamma^{n_0+1}}\sum_{i=1}^{n_0}\|x_i(0)\| \\ &\quad + M_1, \\ H_3 &= 4L^2\frac{n_0C\mu + \mu + 1}{\mu^2\gamma^{2n_0}}(1 + \ln(T))^2, \end{aligned}$$

with

$$\begin{aligned} M_1 &= \frac{L^2}{2n_0\mu\gamma^{2n_0}} + 2L^2\frac{\gamma + C}{\mu\gamma^{n_0+1}} + \frac{4n_0\gamma L^2 + 2n_0CL^2}{\mu\gamma^{2n_0+1}} \\ &\quad + \frac{2L}{n_0\mu\gamma^{n_0}} + \frac{2CL}{\mu\gamma^{n_0+1}}, \end{aligned}$$

is a constant.

Before proving this result, we first define another type of regret, known as network regret, to simplify the process of studying the convergence result for the individual regret in (3).

$$Reg_T = \sum_{t=1}^{T}\sum_{i=1}^{n_0} f_t^i(x_i(t)) - \sum_{t=1}^{T}\sum_{i=1}^{n_0} f_t^i(z^\star). \tag{19}$$

Then, the statement for the performance of network regret is as follows.

Theorem 2 (Network Regret). *Suppose Assumptions 1 and 2 hold, the sequence* $\{x_i(t)\}$, $i \in \{1, 2, \ldots, n_0\}$, *generated by R-DOCO with step size* $\alpha(t) = 1/(n_0 \mu t \gamma^{n_0})$, *satisfies the network regret bound*

$$Reg_T \leq \widetilde{H}_1 + \widetilde{H}_2(1 + \ln T) + \widetilde{H}_3(1 + \ln(T))^2, \tag{20}$$

where

$$\begin{aligned}
\widetilde{H}_1 &= \frac{n_0\mu}{2}\|y(0) - z^\star\|^2 + n_0 CL\frac{1+\gamma^{n_0}}{\gamma^{n_0+1}}\sum_{i=1}^{n_0}\|x_i(0)\| \\
&\quad + \frac{\mu C n_0}{\gamma}\sum_{i=1}^{n_0}\|x_i(0)\|\|y(0) - z^\star\|, \\
\widetilde{H}_2 &= 2n_0 L\frac{C+2\gamma}{\gamma^{n_0+1}}L\|y(0) - z^\star\| + \frac{2n_0 LC}{\gamma^{n_0+1}}\sum_{i=1}^{n_0}\|x_i(0)\| \\
&\quad + M_2, \\
\widetilde{H}_3 &= 4L^2\frac{n_0 C\mu + \mu + 1}{\mu^2\gamma^{2n_0}}(1 + \ln(T))^2,
\end{aligned}$$

with

$$M_2 = \frac{L^2}{2n_0\mu\gamma^{2n_0}} + 2L^2\frac{\gamma + C}{\mu\gamma^{n_0+1}} + \frac{4n_0\gamma L^2 + 2n_0 CL^2}{\mu\gamma^{2n_0+1}},$$

is a constant.

Remark 2. *R-DOCO employs a projected gradient descent method specifically designed for constrained distributed optimization problems, offering enhanced resilience compared to existing algorithms. Unlike previous approaches, such as those by [9–11], R-DOCO proposes an online version that effectively handles an arbitrarily large and unknown number of adversarial nodes. This capability significantly increases its practical applicability in real-world scenarios where the number of adversarial agents may vary. Furthermore, R-DOCO relaxes the cost function's structure from a convex combination to encompass a broader range of distributed optimization problems, thereby enhancing its versatility. Overall, the unique features of R-DOCO position it as a substantial advancement in the field of resilient distributed optimization.*

Proof of the Main Result

The main concept is to first bound the network regret Reg_T, then utilize the relationship between $\sum_{t=1}^{T}\sum_{i=1}^{n_0} f_t^i(x_j(t))$ and $\sum_{t=1}^{T}\sum_{i=1}^{n_0} f_t^i(x_i(t))$.

Before giving the proof of Theorem 2, some lemmas need to be given first, which will be used during the proof. First, we give the convergence bound for $x_i(t)$ with the auxiliary variable $y(t)$ as the measurement criteria.

Lemma 4. *Under Assumptions 1 and 2, the sequence* $\{x_i(t)\}$, $i \in \{1, 2, \ldots, n_0\}$ *is generated by R-DOCO. Then, for step size* $\alpha(t) = 1/(n_0 \mu t \gamma^{n_0})$, $t \geq 1$, *and any finite horizon* T, *we have*

$$\begin{aligned} &\sum_{t=1}^{T} \sum_{i=1}^{n_0} \|x_i(t) - y(t)\| \\ &\leq \frac{n_0 C}{\gamma} \sum_{i=1}^{n_0} \|x_i(0)\| + 2L \frac{\gamma + C}{\mu \gamma^{n_0+1}} (1 + \ln T). \end{aligned} \tag{21}$$

Proof. From (12), we have that the $x_i(t)$ for each node i can be written as

$$\begin{aligned} x_i(t) = &\sum_{j=1}^{n_0} \Phi_{ij}(t-1, 0) x_j(0) - \sum_{k=1}^{t} \sum_{j=1}^{n_0} \Phi_{ij}(t, k) e_j(k-1) \\ &+ \sum_{k=1}^{t} \sum_{j=1}^{n_0} \Phi_{ij}(t, k) \alpha(k-1) g_j(k-1). \end{aligned} \tag{22}$$

The same as for the online optimization analysis in [14,17], we set $\alpha(0) = 0$ to eliminate the influence caused by the absence of the step size $\alpha(t)$ when $t = 0$. Then, from [2] and Lemma 3, we subtract $x_i(t)$ with $y(t)$ to obtain

$$\begin{aligned} \|x_i(t) - y(t)\| \leq &\sum_{k=1}^{t} \Big(\alpha(k-1) \sum_{j=1}^{n_0} \big\| (\Phi_{ij}(t, k) - \theta_j(k)) \big\| \big\| g_j(k-1) \big\| \Big) \\ &+ \sum_{k=1}^{t} \sum_{j=1}^{n_0} \big\| (\Phi_{ij}(t, k) - \theta_j(k)) \big\| \big\| e_j(k-1) \big\| \\ &+ \sum_{j=1}^{n_0} \big\| (\Phi_{ij}(t-1, 0) - \theta_j(0)) \big\| \big\| x_j(0) \big\| \\ &+ \alpha(t) \sum_{j=1}^{n_0} \theta_j(t) \big\| g_i(t) - g_j(t) \big\| \\ &\leq C(1-\gamma)^{t-1} \sum_{j=1}^{n_0} \big\| x_j(0) \big\| + 2 n_0 L \alpha(t) \\ &+ 2 n_0 C L \sum_{k=1}^{t} \alpha(k-1) (1-\gamma)^{t-k}. \end{aligned} \tag{23}$$

Considering the step size $1/(n_0 \mu t \gamma^{n_0})$ and summing from $t = 1$ to T, (23) yields

$$\begin{aligned} &\sum_{t=1}^{T} \|x_i(t) - y(t)\| \leq C \sum_{t=1}^{T} (1-\gamma)^{t-1} \sum_{j=1}^{n_0} \big\| x_j(0) \big\| + 2 n_0 L \sum_{t=1}^{T} \alpha(t) \\ &+ 2 n_0 C L \sum_{t=1}^{T} \sum_{k=1}^{t} \alpha(k-1) (1-\gamma)^{t-k} \\ &\leq \frac{C}{\gamma} \sum_{j=1}^{n_0} \big\| x_j(0) \big\| + 2L \frac{\gamma + C}{\mu \gamma^{n_0+1}} (1 + \ln T), \end{aligned} \tag{24}$$

where the last inequality is due to the fact that $1 - \gamma \in [1, 0)$ and

$$\sum_{t=1}^{T} \sum_{s=1}^{t} \frac{(1-\gamma)^{t-k}}{s} \leq \sum_{s=1}^{T} \frac{1}{s} \sum_{k=1}^{\infty} (1-\gamma)^{k} \leq \frac{(1 + \int_1^T \frac{\mathrm{d}u}{u})}{\gamma} = \frac{1 + \ln(T)}{\gamma}. \tag{25}$$

Summing from $i = 1$ to n_0 yields (21), which completes the proof. □

Lemma 5. *Under Assumptions 1 and 2, the sequence* $\{x_i(t)\}$, $i \in \{1, 2, \ldots, n_0\}$ *is generated by R-DOCO. Then, for step size* $\alpha(t) = 1/(n_0 \mu t \gamma^{n_0})$, $t \geq 1$, *and any finite horizon* T, *we have*

$$\begin{aligned}\sum_{t=1}^{T}\sum_{i=1}^{n_0}\langle g_i(t), y(t) - z^\star\rangle \leq & \left(\frac{L^2}{2n_0\mu\gamma^{2n_0}} + \frac{4n_0\gamma L^2 + 2n_0 C L^2}{\mu\gamma^{2n_0+1}}\right)(1 + \ln(T)) \\ & + \frac{n_0 C L}{\gamma^{n_0+1}}\sum_{i=1}^{n_0}\|x_i(0)\| + \frac{n_0\mu}{2}\sum_{t=1}^{T}\|y(t) - z^\star\|^2 \\ & + \frac{n_0\mu}{2}\|y(T+1) - z^\star\|^2. \end{aligned} \tag{26}$$

Proof. From Lemma 3, we can obtain

$$\begin{aligned}\sum_{t=1}^{T}\sum_{i=1}^{n_0}\langle g_i(t), y(t) - z^\star\rangle &= \sum_{t=1}^{T}\left\langle \sum_{i=1}^{n_0} g_i(t), y(t) - z^\star \right\rangle = \sum_{t=1}^{T}\left\langle \mathbf{1}^{\mathrm{T}}\mathbf{g}(t), y(t) - z^\star \right\rangle \\ &\leq \frac{1}{\gamma^{n_0}}\sum_{t=1}^{T}\langle\langle\theta(t+1), \mathbf{g}(t)\rangle\mathbf{1}, y(t) - z^\star\rangle. \end{aligned} \tag{27}$$

From (17), with Lemma 1 and Assumption 1, we have

$$\begin{aligned}&\|y(t+1) - z^\star\|^2 - \|y(t) - z^\star\|^2 \\ &= \|y(t) - \alpha(t)\langle\theta(t+1), \mathbf{g}(t)\rangle + \langle\theta(t+1), \mathbf{e}(t)\rangle - z^\star\|^2 - \|y(t) - z^\star\|^2 \\ &= \|\alpha(t)\langle\theta(t+1), \mathbf{g}(t)\rangle - \langle\theta(t+1), \mathbf{e}(t)\rangle\|^2 \\ &- 2\left\langle \left(\alpha(t)\langle\theta(t+1), \mathbf{g}(t)\rangle - \langle\theta(t+1), \mathbf{e}(t)\rangle\right)\mathbf{1}, y(t) - z^\star \right\rangle \\ &\leq \alpha^2(t)L^2 - 2\alpha(t)\langle\langle\theta(t+1), \mathbf{g}(t)\rangle\mathbf{1}, y(t) - z^\star\rangle \\ &+ 2\langle\langle\theta(t+1), \mathbf{e}(t)\rangle\mathbf{1}, y(t) - z^\star\rangle. \end{aligned} \tag{28}$$

Due to Lemma 1 with $z^\star$ having to satisfy $z^\star \in \Omega$, we can obtain

$$e_i^{\mathrm{T}}(t)(d_i(t) - \alpha(t)g_i(t) - z^\star) \leq -\|e_i(t)\|^2. \tag{29}$$

Since (11), the left-hand side of (29) can be further obtained:

$$e_i^{\mathrm{T}}(t)((d_i(t) - \alpha(t)g_i(t) - z^\star) = e_i^{\mathrm{T}}(t)(y(t) - z^\star) + e_i^{\mathrm{T}}(t)(x_i(t+1) - e_i(t) - y(t)). \tag{30}$$

Substituting (30) into (29) and rearranging, yields

$$\begin{aligned}e_i^{\mathrm{T}}(t)(y(t) - z^\star) &\leq e_i^{\mathrm{T}}(t)(x_i(t+1) - e_i(t) - y(t)) - \|e_i(t)\|^2 \\ &\leq \|e_i(t)\|\|x_i(t+1) - y(t)\|. \end{aligned} \tag{31}$$

Combining (12) and (23), we can obtain

$$\begin{aligned}&\|x_i(t+1) - y(t)\| \\ &\leq \sum_{j=1}^{n_0}\|(\theta_j(0) - \Phi_{ij}(t-1, 0))\|\|x_j(0)\| + \sum_{k=1}^{t}\sum_{j=1}^{n_0}\|(\theta_j(k) - \Phi_{ij}(t,k))e_j(k-1)\| \\ &+ \sum_{j=1}^{n_0}\|\Phi_{ij}(t, t+1)e_j(k-1)\| + \sum_{j=1}^{n_0}\|\Phi_{ij}(t, t+1)\alpha(k-1)g_j(k-1)\| \\ &+ \sum_{k=1}^{t}\sum_{j=1}^{n_0}\|(\theta_j(k) - \Phi_{ij}(t,k))\alpha(k-1)g_j(k-1)\| \end{aligned}$$

$$\leq C(1-\gamma)^{t-1}\sum_{j=1}^{n_0}\|x_j(0)\| + 4n_0L\alpha(t) + 2n_0CL\sum_{k=1}^{t}\alpha(k-1)(1-\gamma)^{t-k}, \tag{32}$$

where the last inequality is because $\Phi(t,k) = I_{n_0}$ with $t < k$. Then, since Lemma 3, we have

$$\langle\langle\theta(t+1),\mathbf{e}(t)\rangle\mathbf{1}, y(t)-z^\star\rangle \leq n_0L\alpha(t)\Bigg(C(1-\gamma)^{t-1}\sum_{j=1}^{n_0}\|x_j(0)\| + 4n_0L\alpha(t) + 2n_0CL\sum_{k=1}^{t}\alpha(k-1)(1-\gamma)^{t-k}\Bigg). \tag{33}$$

Then, by rearranging (28) with (33) and summing from $t = 1$ to T, we obtain

$$\begin{aligned}
&\sum_{t=1}^{T}\langle\langle\theta(t+1),\mathbf{g}(t)\rangle\mathbf{1}, y(t)-z^\star\rangle\\
&\leq \sum_{t=1}^{T}\left(\frac{\|y(t)-z^\star\|^2}{2\alpha(t)} - \frac{\|y(t+1)-z^\star\|^2}{2\alpha(t)} + \frac{\alpha(t)L^2}{2}\right)\\
&+ \sum_{t=1}^{T}\frac{\langle\langle\theta(t+1),\mathbf{e}(t)\rangle\mathbf{1}, y(t)-z^\star\rangle}{\alpha(t)}\\
&\leq L^2\sum_{t=1}^{T}\frac{\alpha(t)}{2} + \frac{\|y(1)-z^\star\|^2}{2\alpha(1)} + n_0L\sum_{t=1}^{T}\Bigg(4n_0L\alpha(t)\\
&+ C(1-\gamma)^{t-1}\sum_{j=1}^{n_0}\|x_j(0)\| + 2n_0CL\sum_{k=1}^{t}\alpha(k-1)\\
&(1-\gamma)^{t-k}\Bigg) + \sum_{t=1}^{T}\Big(\|y(t+1)-z^\star\|^2\Big)\left(\frac{1}{2\alpha(t+1)} - \frac{1}{2\alpha(t)}\right).
\end{aligned} \tag{34}$$

Setting $\alpha(t) = 1/(n_0\mu t\gamma^{n_0})$, (34) can further obtain

$$\begin{aligned}
&\sum_{t=1}^{T}\langle\langle\theta(t+1),\mathbf{g}(t)\rangle\mathbf{1}, y(t)-z^\star\rangle\\
&\leq \left(\frac{L^2}{2n_0\mu\gamma^{n_0}} + \frac{4n_0\gamma L^2 + 2n_0CL^2}{\mu\gamma^{n_0+1}}\right)(1+\ln(T)) + \frac{n_0CL}{\gamma}\sum_{i=1}^{n_0}\|x_i(0)\|\\
&+ \frac{n_0\mu\gamma^{n_0}}{2}\|y(1)-z^\star\|^2 + \frac{n_0\mu\gamma^{n_0}}{2}\sum_{t=1}^{T}\|y(t+1)-z^\star\|^2.
\end{aligned} \tag{35}$$

The proof then follows immediately using (27) and (35). □

Denote $X = \mu\|x_i(t)-y(t)\|\|y(t)-z^\star\|$, and the final stepping stone in the proof of Theorem 2 is stated next.

Lemma 6. *Under Assumptions 1 and 2, the sequence $\{x_i(t)\}$, $i \in \{1,2,\ldots,n_0\}$ is generated by R-DOCO. Then, for step size $\alpha(t) = 1/(n_0\mu t\gamma^{n_0})$, $t \geq 1$, and any finite horizon T, we have*

$$\begin{aligned}
\sum_{t=1}^{T}\sum_{i=1}^{n_0}X &\leq \frac{\mu Cn_0}{\gamma}\sum_{i=1}^{n_0}\|x_i(0)\|\|y(0)-z^\star\| + 4L^2\frac{n_0C+1}{\mu\gamma^{2n_0}}(1+\ln(T))^2\\
&+ \left[2n_0L\frac{C+\gamma}{\gamma^{n_0+1}}\|y(0)-z^\star\| + \frac{2n_0LC}{\gamma^{n_0+1}}\sum_{i=1}^{n_0}\|x_i(0)\|\right](1+\ln(T)).
\end{aligned} \tag{36}$$

Proof. From (17), we can write

$$y(t+1) = y(0) - \sum_{k=1}^{t} \alpha(k)\langle \theta(k+1), \mathbf{g}(k)\rangle + \langle \theta(k+1), \mathbf{e}(k)\rangle.$$

Subtracting $z^\star$ on both sides and using the triangle law, we obtain

$$\begin{aligned}\|y(t+1) - z^\star\| &= \|y(0) - z^\star\| + \sum_{k=1}^{t}\sum_{j=1}^{n_0} \theta_j(k+1)\|e_j(k)\| \\ &\quad + \sum_{k=1}^{t} \alpha(k) \sum_{j=1}^{n_0} \theta_j(k+1)\|g_j(k)\| \leq \|y(0) - z^\star\| + 2n_0 L \sum_{k=1}^{t} \alpha(k). \end{aligned} \tag{37}$$

Thus, we can directly obtain

$$\begin{aligned}\|y(T+1) - z^\star\| &\leq \|y(0) - z^\star\| + 2n_0 L \sum_{k=1}^{t} \alpha(k) \\ &\leq \|y(0) - z^\star\| + \frac{2L}{\mu\gamma^{n_0}}(1 + \ln(T)). \end{aligned} \tag{38}$$

From (23) and (37) with $\alpha(0) = 0$, we have

$$\begin{aligned} X \leq & \left[2n_0\mu L\alpha(t) + \mu C(1-\gamma)^{t-1} \sum_{j=1}^{n_0}\|x_j(0)\| \right. \\ & \left. + 2n_0\mu CL \sum_{k=1}^{t-1} \alpha(k)(1-\gamma)^{t-k-1}\right]\left[\|y(0) - z^\star\| + 2n_0 L \sum_{k=1}^{t-1}\alpha(k)\right] \\ & = 2n_0\mu L\alpha(t)\|y(0) - z^\star\| + \mu C(1-\gamma)^{t-1}\sum_{j=1}^{n_0}\|x_j(0)\|\|y(0) - z^\star\| \\ & + 2n_0\mu CL\|y(0) - z^\star\| \sum_{k=1}^{t-1}\alpha(k)(1-\gamma)^{t-k-1} + 4n_0^2\mu L^2\alpha(t)\sum_{k=1}^{t-1}\alpha(k) \\ & + 2n_0 L\mu C(1-\gamma)^{t-1}\sum_{j=1}^{n_0}\|x_j(0)\|\sum_{k=1}^{t-1}\alpha(k) \\ & + 4n_0^2\mu CL^2\sum_{k=1}^{t-1}\alpha(k)(1-\gamma)^{t-k-1}\sum_{k=1}^{t-1}\alpha(k). \end{aligned} \tag{39}$$

Recalling $\alpha(t) = 1/(n_0\mu t\gamma^{n_0})$, we have

$$\sum_{t=1}^{T}\sum_{i=1}^{n_0} 2n_0\mu L\alpha(t)\|y(0) - z^\star\| \leq \frac{2n_0 L}{\gamma^{n_0}}\|y(0) - z^\star\|(1 + \ln(T)), \tag{40}$$

$$\sum_{t=1}^{T}\sum_{i=1}^{n_0} 4n_0^2\mu L^2\alpha(t)\sum_{k=1}^{t-1}\alpha(k) \leq \frac{4n_0 L^2}{\mu\gamma^{2n_0}}(1 + \ln(T))^2, \tag{41}$$

where we use the fact that $\sum_{s=1}^{t} 1/s \leq 1 + \ln(T)$. Due to (25), we also have

$$\sum_{t=1}^{T}\sum_{i=1}^{n_0}\mu C(1-\gamma)^{t-1}\sum_{j=1}^{n_0}\|x_j(0)\|\|y(0) - z^\star\| \leq \frac{\mu C n_0}{\gamma}\sum_{j=1}^{n_0}\|x_j(0)\|\|y(0) - z^\star\|, \tag{42}$$

$$\sum_{t=1}^{T}\sum_{i=1}^{n_0} 2n_0 L\mu C(1-\gamma)^{t-1}\sum_{j=1}^{n_0}\|x_j(0)\|\sum_{k=1}^{t-1}\alpha(k) \le \frac{2n_0 CL}{\gamma^{n_0+1}}\sum_{j=1}^{n_0}\|x_j(0)\|(1+\ln(T)). \tag{43}$$

Finally,

$$\sum_{t=1}^{T}\sum_{i=1}^{n_0} 2n_0\mu CL\|y(0)-z^\star\|\sum_{k=1}^{t-1}\alpha(k)(1-\gamma)^{t-k-1} \le \frac{2n_0 CL}{\gamma^{n_0+1}}\|y(0)-z^\star\|(1+\ln(T)) \tag{44}$$

$$\sum_{t=1}^{T}\sum_{i=1}^{n_0} 4n_0^2\mu CL^2\sum_{k=1}^{t-1}\alpha(k)(1-\gamma)^{t-k-1}\sum_{k=1}^{t-1}\alpha(k) \le \frac{4n_0 CL^2}{\mu\gamma^{2n_0}}(1+\ln(T))^2. \tag{45}$$

In (44) and (45), by rearranging the summation, we have $\sum_{t=1}^{T}\sum_{s=1}^{t-1}\left((1-\gamma)^{t-k-1}/s\right) \le ((1+\ln(T))/\gamma)$. Using (40)–(45) in (39) then yields the result. □

We are now in a position to prove Theorem 2.

Proof of Theorem 2. Due to the convexity of f_t^i, it follows that

$$\begin{aligned}
Reg_T &= \sum_{t=1}^{T}\sum_{i=1}^{n_0}\left(f_t^i(x_i(t)) - f_t^i(z^\star)\right) \le \sum_{t=1}^{T}\sum_{i=1}^{n_0}\left(\langle g_i(t), x_i(t)-z^\star\rangle - \frac{\mu}{2}\|x_i(t)-z^\star\|^2\right)\\
&= \sum_{t=1}^{T}\sum_{i=1}^{n_0}\langle g_i(t), x_i(t)-y(t)+y(t)-z^\star\rangle - \frac{\mu}{2}\sum_{t=1}^{T}\sum_{i=1}^{n_0}\|x_i(t)-y(t)+y(t)-z^\star\|^2\\
&\le \sum_{t=1}^{T}\sum_{i=1}^{n_0}\left(L\|x_i(t)-y(t)\| + \langle g_i(t), y(t)-z^\star\rangle - \mu\langle x_i(t)-y(t), y(t)-z^\star\rangle\right)\\
&- \frac{n_0\mu}{2}\sum_{t=1}^{T}\|y(t)-z^\star\|^2.
\end{aligned} \tag{46}$$

Then, from Lemma 5, we have

$$\begin{aligned}
Reg_T &\le \left(\frac{L^2}{2n_0\mu\gamma^{2n_0}} + \frac{4n_0\gamma L^2 + 2n_0 CL^2}{\mu\gamma^{2n_0+1}}\right)(1+\ln(T)) + \frac{n_0 CL}{\gamma^{n_0+1}}\sum_{i=1}^{n_0}\|x_i(0)\|\\
&+ \mu\sum_{t=1}^{T}\sum_{i=1}^{n_0}\|x_i(t)-y(t)\|\|y(t)-z^\star\| + \frac{n_0\mu}{2}\|y(T+1)-z^\star\|^2\\
&+ L\sum_{t=1}^{T}\sum_{i=1}^{n_0}\|x_i(t)-y(t)\|\\
&\le \left(\frac{L^2}{2n_0\mu\gamma^{2n_0}} + \frac{4n_0\gamma L^2 + 2n_0 CL^2}{\mu\gamma^{2n_0+1}}\right)(1+\ln(T)) + \frac{n_0 CL}{\gamma^{n_0+1}}\sum_{i=1}^{n_0}\|x_i(0)\|\\
&+ \frac{n_0\mu}{2}\|y(T+1)-z^\star\|^2 + \mu\sum_{t=1}^{T}\sum_{i=1}^{n_0}\|x_i(t)-y(t)\|\|y(t)-z^\star\|\\
&+ L\sum_{t=1}^{T}\sum_{i=1}^{n_0}\|x_i(t)-y(t)\|.
\end{aligned} \tag{47}$$

From (38), we can directly obtain

$$\begin{aligned}
\|y(T+1)-z^\star\|^2 &\le \left(\|y(0)-z^\star\| + \frac{2L}{\mu\gamma^{n_0}}(1+\ln(T))\right)^2\\
&= \|y(0)-z^\star\|^2 + \frac{4L}{\mu\gamma^{n_0}}\|y(0)-z^\star\|(1+\ln(T)) + \frac{4L^2}{\mu^2\gamma^{2n_0}}(1+\ln(T))^2.
\end{aligned} \tag{48}$$

Substituting Lemma 4, Lemma 6, and (48) into (47), then yields the result. □

In order to demonstrate the proof of Theorem 1, using the preceding result about the network regret, we provide an upper bound on the individual regret.

Proof of Theorem 1. First, from the convexity of cost functions, note that

$$\begin{aligned} Reg_T^j - Reg_T &= \sum_{t=1}^{T}\sum_{i=1}^{n_0} f_t^i(x_j(t)) - \sum_{t=1}^{T}\sum_{i=1}^{n_0} f_t^i(x_i(t)) \\ &\leq \sum_{t=1}^{T}\sum_{i=1}^{n_0}\left\langle g^j(t), x_j(t) - x_i(t)\right\rangle \leq \sum_{t=1}^{T}\sum_{i=1}^{n_0}\left\|g^j(t)\right\|\left\|x_j(t) - x_i(t)\right\|. \end{aligned} \tag{49}$$

The term can be bounded by (23):

$$\begin{aligned} \left\|x_j(t) - x_i(t)\right\|^2 &= \left\|x_j(t) - y(t) + y(t) - x_i(t)\right\|^2 \\ &\leq 4\Bigg(C(1-\gamma)^{t-1}\sum_{j=1}^{n_0}\|x_j(0)\| + 2\alpha(t)L \\ &\quad + 2n_0CL\sum_{k=1}^{t}\alpha(k-1)(1-\gamma)^{t-k}\Bigg)^2. \end{aligned} \tag{50}$$

Now, by choosing the step size as $\alpha(t) = 1/(n_0\mu t\gamma^{n_0})$, from inequalities (49) and (50), we obtain

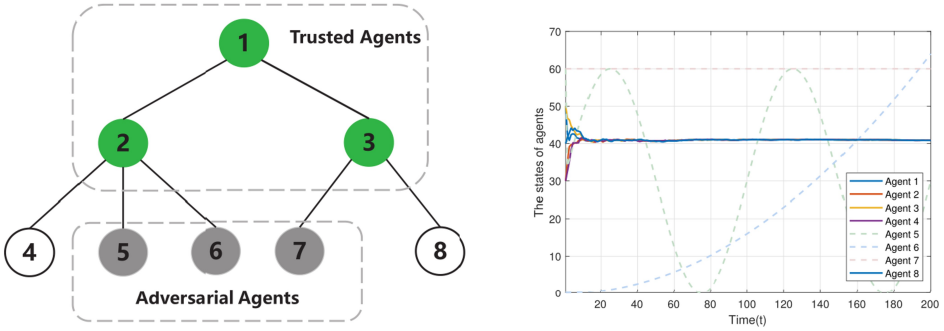
$$\begin{aligned} &Reg_T^j - Reg_T \leq 2n_0L\sum_{t=1}^{T}\Bigg(C(1-\gamma)^{t-1}\sum_{i=1}^{n_0}\|x_i(0)\| + 2\alpha(t)L \\ &+ 2n_0CL\sum_{k=1}^{t}\alpha(k-1)(1-\gamma)^{t-k}\Bigg) \\ &\leq 2n_0L\left(\frac{C}{\gamma}\sum_{i=1}^{n_0}\|x_i(0)\| + \frac{2L}{n_0\mu\gamma^{n_0}}(1+\ln(T)) + \frac{2CL}{\mu\gamma^{n_0+1}}(1+\ln(T))\right) \\ &= \left(\frac{2L}{n_0\mu\gamma^{n_0}} + \frac{2CL}{\mu\gamma^{n_0+1}}\right)(1+\ln(T)) + \frac{2n_0LC}{\gamma}\sum_{i=1}^{n_0}\|x_i(0)\|. \end{aligned} \tag{51}$$

Finally, the proof of Theorem 1 follows by utilizing the network regret bound in Theorem 2 and the bound on the difference between the individual regret and the network regret, obtained above. The proof is complete. □

5. Numerical Results

A simple example is provided to illustrate the performance of our algorithm in this section. Motivated by [11], we consider a sensor network shown in Figure 1a with $n = 8$ nodes, including three adversarial, three trusted, and two normal nodes. These sensors are used to observe a vector $x \in \mathbb{R}^d$, which is randomly sampled. Each sensor $i \in n$ can receive a time-varying measurement vector $z_i(t) \in \mathbb{R}^{d_i}$ at time t. We assume that each measurement $z_i(t)$ is associated with some noise, modeled as a linear function, which can be expressed as $z_i(t) = V_i x + \xi_i$, where $V_i \in \mathbb{R}^{d_i \times d}$ is the observation matrix with a bounded norm and ξ_i is the white noise. We use $\hat{x}_i$ to denote the local estimate for x_i to minimize the cost function

$$f_i(t) = \frac{1}{2}\|z_i(t) - V_i(t)\hat{x}_i\|^2. \tag{52}$$



(a) **(b)**

Figure 1. Network example and performance evaluation with adversarial attacks. (**a**) Example; (**b**) convergence.

We then explore how R-DOCO performs under adversarial attacks. For $t \geq 1$, adversarial nodes update their states as $x_5 = 30\sin(0.02\pi * t) + 30$, $x_6 = (t/25)^2$, and $x_7 = 60$. We select $V_i = 1$. All the states are satisfied in the interval $[20, 50]$. From Figure 1b, we see that all the local variables of normal agents and trusted agents will converge under adversarial attacks.

The convergence performance of R-DOCO is presented in Figure 2, which shows that the average individual regret, (Reg_T^j/T), of sensor $j \in n_0$ tends to zero as T tends to 100. Finally, we present the simulation results of the network regret (Reg_T/T) with R-DOCO and projected distributed gradient descent (P-DGD) in Figure 3. P-DGD is an optimization algorithm that solves large-scale distributed problems by enabling multiple agents to collaboratively update a global parameter vector. Each agent computes gradients based on local data and performs a projection to maintain feasibility within specified constraints. This decentralized approach reduces communication overhead by sharing only updates instead of entire datasets. It is obvious from Figure 3 that the regret of P-DGD cannot even converge, while the network regret under R-DOCO tends to zero fast.

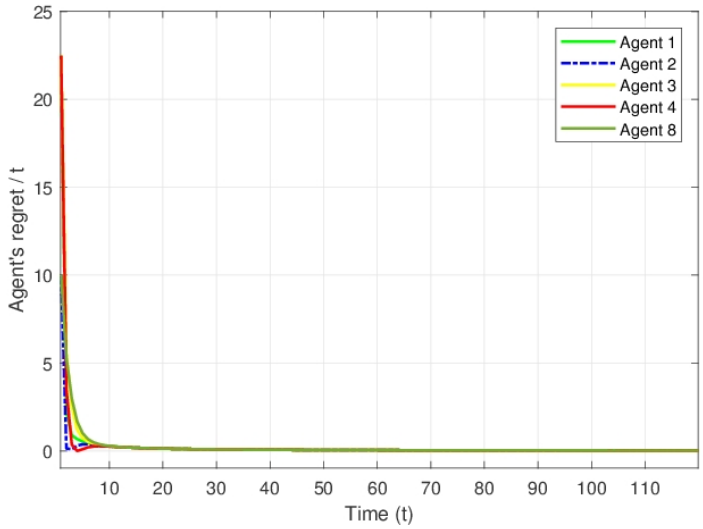


Figure 2. Average individual regret versus the total number of iterations T for all the normal and trusted sensors in the network.

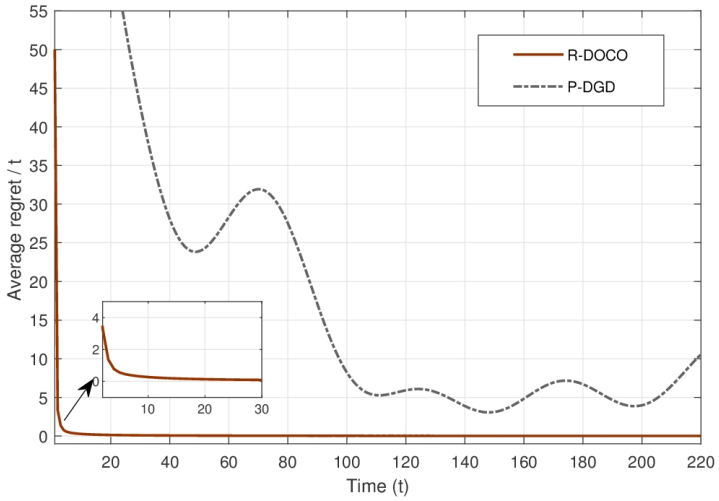


Figure 3. Network regret (Reg_T/T) with R-DOCO and P-DGD.

6. Conclusions

In this article, we analyzed the distributed constrained optimization problem in the presence of adversarial attacks, where certain agents fail or act maliciously, thus not following the update rule. To address this, we proposed a resilient distributed projected gradient descent online optimization algorithm, named R-DOCO, capable of handling large and unknown numbers of adversarial nodes. We provided specific bounds for both individual regret and network regret, demonstrating that the individual regret bound grows linearly in an 8-node network. Potential future directions include improving the regret bounds and extending R-DOCO to non-convex functions, thereby broadening its applicability. Additionally, adapting the algorithm for dynamic networks with node mobility could enhance its performance in applications such as decentralized control, resource allocation, and edge computing. Exploring R-DOCO's impact on specific graph structures, like small-world or scale-free networks, may further reveal insights into its resilience and efficiency across various topologies.

Author Contributions: Z.K. is responsible for the methodology and software design as well as writing the first draft. H.X. is responsible for validation and data management as well as reviewing the paper. C.P. is responsible for related management and providing funding assistance. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China Key Project: Space-Earth Integration Intelligent Network Traffic Theory and Key Technology Fund Project 61931004.

Data Availability Statement: All relevant data are within the manuscript .

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

References

1. Yang, T.; Yi, X.; Wu, J.; Yuan, Y.; Wu, D.; Meng, Z.; Hong, Y.; Wang, H.; Lin, Z.; Johansson, K.H. A survey of distributed optimization. *Annu. Rev. Control* **2019**, *47*, 278–305. [CrossRef]
2. Nedic, A.; Ozdaglar, A.; Parrilo, P.A. Constrained consensus and optimization in multi-agent networks. *IEEE Trans. Autom. Control* **2010**, *55*, 922–938. [CrossRef]
3. Wei, M.; Chen, D.; Yu, W.; Yao, L. Differentially Private Decentralized Weakly Convex Delay-Tolerance Optimization in Robust Phase Retrieval. In Proceedings of the 2023 13th International Conference on Information Science and Technology (ICIST), Cairo, Egypt, 8–14 December 2023; pp. 543–549. [CrossRef]
4. Pasqualetti, F.; Dorfler, F.; Bullo, F. Attack Detection and Identification in Cyber-Physical Systems. *IEEE Trans. Autom. Control* **2013**, *58*, 2715–2729. [CrossRef]
5. Zhang, J.; Yang, D.; Li, W.; Zhang, H.; Li, G.; Gu, P. Resilient Output Control of Multiagent Systems With DoS Attacks and Actuator Faults: Fully Distributed Event-Triggered Approach. *IEEE Trans. Cybern.* **2024**, 1–10. [CrossRef]

6. Sundaram, S.; Gharesifard, B. Consensus-based distributed optimization with malicious nodes. In Proceedings of the 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 29 September–2 October 2015; pp. 244–249. [CrossRef]
7. Liu, Z.; Xiong, X.; Li, Y.; Yu, Y.; Lu, J.; Zhang, S.; Xiong, F. HyGloadAttack: Hard-label black-box textual adversarial attacks via hybrid optimization. *Neural Netw.* **2024**, *178*, 106461. [CrossRef]
8. Su, L.; Vaidya, N. Byzantine multi-agent optimization: Part I. *arXiv* **2015**, arXiv:1506.04681 .
9. Sahoo, S.; Gokhale, A.; Kalaimani, R.K. Distributed Online Optimization with Byzantine Adversarial Agents. *arXiv* **2021**, arXiv:2109.12340.
10. Sundaram, S.; Gharesifard, B. Distributed optimization under adversarial nodes. *IEEE Trans. Autom. Control* **2018**, *64*, 1063–1076. [CrossRef]
11. Zhao, C.; He, J.; Wang, Q.G. Resilient distributed optimization algorithm against adversarial attacks. *IEEE Trans. Autom. Control* **2019**, *65*, 4308–4315. [CrossRef]
12. Su, L.; Vaidya, N.H. Byzantine-Resilient Multiagent Optimization. *IEEE Trans. Autom. Control* **2021**, *66*, 2227–2233. [CrossRef]
13. Wei, M.; Yu, W.; Liu, H.; Chen, D. Byzantine-Resilient Distributed Bandit Online Optimization in Dynamic Environments. *IEEE Trans. Ind. Cyber-Phys. Syst.* **2024**, *2*, 154–165. [CrossRef]
14. Akbari, M.; Gharesifard, B.; Linder, T. Distributed online convex optimization on time-varying directed graphs. *IEEE Trans. Control. Netw. Syst.* **2015**, *4*, 417–428. [CrossRef]
15. Wei, M.; Yu, W.; Liu, H.; Xu, Q. Distributed Weakly Convex Optimization Under Random Time-Delay Interference. *IEEE Trans. Netw. Sci. Eng.* **2024**, *11*, 212–224. [CrossRef]
16. He, L.; Ai, Q.; Yang, X.; Ren, Y.; Wang, Q.; Xu, Z. Boosting adversarial robustness via self-paced adversarial training. *Neural Netw.* **2023**, *167*, 706–714. [CrossRef] [PubMed]
17. Zhao, Z.; Yang, Z.; Jiang, L.; Yang, J.; Ge, Q. Privacy Preserving Distributed Bandit Residual Feedback Online Optimization Over Time-Varying Unbalanced Graphs. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 2284–2297. [CrossRef]
18. Lu, K.; Jing, G.; Wang, L. Online distributed optimization with strongly pseudoconvex-sum cost functions. *IEEE Trans. Autom. Control* **2019**, *65*, 426–433. [CrossRef]
19. Yi, X.; Li, X.; Yang, T.; Xie, L.; Chai, T.; Johansson, K.H. Distributed Bandit Online Convex Optimization With Time-Varying Coupled Inequality Constraints. *IEEE Trans. Autom. Control* **2021**, *66*, 4620–4635. [CrossRef]
20. Li, X.; Yi, X.; Xie, L. Distributed online optimization for multi-agent networks with coupled inequality constraints. *IEEE Trans. Autom. Control* **2020**, *66*, 3575–3591. [CrossRef]
21. Cao, X.; Liu, K.J.R. Online Convex Optimization With Time-Varying Constraints and Bandit Feedback. *IEEE Trans. Autom. Control* **2019**, *64*, 2665–2680. [CrossRef]
22. Huang, K.; Siegel, M.; Madnick, S. Systematically understanding the cyber attack business: A survey. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 1–36. [CrossRef]
23. Caro, Y.; West, D.B.; Yuster, R. Connected domination and spanning trees with many leaves. *SIAM J. Discret. Math.* **2000**, *13*, 202–211. [CrossRef]
24. Hazan, E. Introduction to Online Convex Optimization. *Found. Trends Optim.* **2016**, *2*, 157–325. [CrossRef]
25. Pease, M.; Shostak, R.; Lamport, L. Reaching agreement in the presence of faults. *J. ACM (JACM)* **1980**, *27*, 228–234. [CrossRef]
26. Nedić, A.; Olshevsky, A. Distributed optimization over time-varying directed graphs. *IEEE Trans. Autom. Control* **2014**, *60*, 601–615. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.