BYZANTINE-RESILIENT DECENTRALIZED RESOURCE ALLOCATION

Runhua Wang[1] *Yaohua Liu*[2] *Qing Ling*[1]

[1]Sun Yat-Sen University [2]Nanjing University of Information Science and Technology

ABSTRACT

This paper considers the resource allocation problem in a decentralized multi-agent network at presence of Byzantine agents. Compared with its centralized counterpart, a decentralized algorithm enjoys better scalability when the network is large-scale, but is more vulnerable when some of the agents are malicious and send wrong messages during the optimization process. We use the Byzantine attack model to describe these malicious actions, and propose a novel Byzantine-resilient decentralized resource allocation algorithm, abbreviated as BREDA. At each iteration of BREDA, each honest agent receives messages from its neighbors, uses coordinate-wise trimmed mean (CTM) to aggregate these messages, and then updates its local primal and dual variables with gradient descent and ascent, respectively. Numerical experiments demonstrate the resilience of BREDA to various Byzantine attacks.

***Index Terms*—** Resource allocation, decentralized multi-agent network, Byzantine-resilience

1. INTRODUCTION

Resource allocation, which aims at assigning a limited amount of resources among a group of users (also known as agents) to maximize their utility or minimize their cost, has become one of the key issues in network optimization. Resource allocation has been extensively investigated in recent decades, and found broad applications in various fields, such as smart grids, wireless networks, etc.

Resource Allocation Algorithms. In general, resource allocation over a network can be modeled as maximizing the average utility or minimizing the average cost of the agents, subject to global and local resource constraints [1]. Centralized algorithms require a master node to coordinate all the agents, and are unscalable to network size [2]. Hence, decentralized algorithms that rely on coordination among neighboring agents have received much more attention. The main challenge in the decentralized algorithms is to handle the global resource constraints that couple all the agents. For instance, [3] and [4] propose decentralized, weighted gradient algorithms. The communication graph is assumed to be fixed in [3], and dynamic in [4]. Primal-dual algorithms with constant step sizes are introduced in [5] and [6]. Several techniques are developed to achieve faster convergence. One is using both gradients and Hessians of the objective function [7, 8], and another is using gradients of the objective function as well as the past iterates when computing the future ones [9]. To overcome the requirement of re-initialization in the optimization process, [10] proposes an initialization-free continuous-time algorithm and establishes its linear convergence rate. An asynchronous resource allocation algorithm is proposed in [11], utilizing delayed gradient information to carry out updates and hence suitable for heterogeneous networks. Non-smooth resource allocation problems are investigated in [12, 13]. Online convex optimization approaches have been proposed in [14, 15] to solve resource allocation problems with time-varying and coupled global resource constraints.

Byzantine-resilience. Most of the aforementioned works assume that all the agents are reliable and strictly follow the algorithms. However, in the real world, some of the agents might be unreliable in either computing or communicating, and even can be malfunctioning. These agents deviate from the expected optimization process and send wrong messages to their neighbors, yielding biased results. We characterize these behaviors with the classical Byzantine attack model [16], in which the malicious agents (also called the Byzantine agents) can collude and arbitrarily modify the messages sent to their neighbors. Such an attack model imposes no restrictions on the Byzantine agents and is worst-case. To the best of our knowledge, so far there has been no work considering Byzantine attacks in decentralized resource allocation. The work of [17] takes noisy observations and communication uncertainties into consideration, and designs a stochastic approximation algorithm to solve the decentralized resource allocation problem. Nevertheless, the noise and uncertainties in [17] are not worst-case, in contrast to the Byzantine attack model studied here. The work of [2] investigates Byzantine-resilient resource allocation, but the proposed algorithm is centralized.

Byzantine-resilient optimization is now a popular topic in federated learning, where a master node coordinates the learning process of geographically distributed agents [18–20]. It has also attracted attention in decentralized consensus optimization, where decentralized agents collaboratively minimize the average cost subject to consensus constraints – namely, all the local iterates must reach the same value [21–27]. Below we briefly survey the existing Byzantine-resilient decentralized consensus optimization algorithms. The work of [21] proposes ByRDiE, in which at every iteration each honest agent chooses a coordinate, uses trimmed mean to aggregate the coordinate of the local iterates received from its neighbors, followed by coordinate gradient descent to update its own local iterate. In the trimmed mean, a given number of b largest and b smallest values are trimmed, and the rest are averaged. However, it is inefficient to screen only one coordinate of the received local iterates in each iteration. To addresses this issue, [22] devises BRIDGE, which allows each honest agent to aggregate the received local iterates in all coordinates at every iteration with coordinate-wise trimmed mean (CTM) and then perform gradient descent. A variant of CTM is proposed in [24], in which for each coordinate, each honest agent trims up to b received values larger than its own value, and up to b received values smaller than its own value. In [23], each honest agent discards a given number of received local iterates which may increase the loss of its own objective. Total variation regularization is adopted in [25,26] to drive the local iterates of the honest agents to be close, for the sake of tolerating Byzantine attacks. The idea of total variation regularization has been extended to decentralized stochas-

Work of Yaohua Liu is supported by NSF Jiangsu grant BK20210642. Work of Qing Ling (corresponding author) is supported by NSF China Grant 61973324, Guangdong Basic and Applied Basic Research Foundation Grant 2021B1515020094, and Guangdong Province Key Laboratory of Computational Science Grant 2020B1212060032.

tic consensus optimization [27]. Despite the success of Byzantine-resilient decentralized consensus optimization algorithms, the ideas therein cannot be directly applied to Byzantine-resilient decentralized resource allocation. The point is that, in the latter, the optimal local variables of the honest agents are not necessarily consensual. This way, naively applying the existing approaches such as trimmed mean, CTM and total variation regularization no longer works.

Our Contributions. We investigate the almost untouched territory of Byzantine-resilient decentralized resource allocation. The Byzantine agents send wrong messages to their neighbors so as to bias the optimization process. For example, they can collude to manipulate the optimization process so that the honest agents are allocated with less resources than needed. As we have indicated above, directly applying the decentralized robust aggregation rules here is infeasible. To address these issues, we make the following contributions.

C1) We propose a primal-dual Byzantine-resilient decentralized resource allocation (BREDA) algorithm, where the primal and dual variables are updated locally. We introduce a local auxiliary variable to each agent for approximating the average amount of required resources, which is used in updating the local dual variable. A first-order decentralized dynamic average consensus method equipped with CTM is then applied to update the local auxiliary variable in a Byzantine-resilient manner.

C2) We theoretically prove that BREDA converges to a neighborhood of the saddle point of a regularized Lagrangian function. We also conduct extensive numerical experiments on a decentralized resource allocation problem. The experimental results show the resilience of BREDA to various Byzantine attacks.

2. PROBLEM FORMULATION

We consider a connected network of N agents, modeled as an undirected, static graph $\mathcal{G}(\mathcal{J}, \mathcal{E})$. The set of vertices $\mathcal{J} := \{1, ..., N\}$ represents the agents in the network and the set of edges $\mathcal{E}$ represents the communication links between the agents. If $(i, j) \in \mathcal{E}$, then agents i and j are neighbors and can communicate with each other. Let $\mathcal{N}_i = \{j \mid (i, j) \in \mathcal{E}\}$ be the set of neighbors of agent i. Resources are allocated among the agents, and our objective is to find an optimal allocation that minimizes the average cost under resource constraints. This decentralized resource allocation problem is as

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & f(\boldsymbol{\theta}) := \frac{1}{N}\sum_{i=1}^{N} f_i(\boldsymbol{\theta}_i), \\ s.t. \quad & \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\theta}_i \le \boldsymbol{s}, \quad \boldsymbol{\theta}_i \in \boldsymbol{C}_i,\ \forall i \in \mathcal{J}. \end{aligned} \tag{1}$$

Therein, $\boldsymbol{\theta}_i \in \mathbb{R}^D$ is the local optimization variable of agent i, representing the amount of allocated resources, and $f_i(\cdot)$ is the continuously differentiable and convex cost function of agent i. The average amount of allocated resources $\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\theta}_i$ is upper bounded by a constant vector $\boldsymbol{s} \in \mathbb{R}^D$ such that $N\boldsymbol{s}$ corresponds to the total amount of resources. Each $\boldsymbol{\theta}_i$ is also confined to a bounded, convex set $\boldsymbol{C}_i$. For simplicity, we collect all the local optimization variables in a vector $\boldsymbol{\theta} := [\boldsymbol{\theta}_1; ...; \boldsymbol{\theta}_N] \in \mathbb{R}^{ND}$.

To obtain an optimal allocation to (1), the agents must communicate with their neighbors and collaboratively optimize their local optimization variables. However, not all the agents in the decentralized network are honest. Some of them might be malfunctioning or even malicious. These agents can arbitrarily deviate from the expected optimization process and send wrong messages to their neighbors. We define them as Byzantine agents. Let $\mathcal{B}$ be the set of Byzantine agents and $\mathcal{H} := \mathcal{J} \setminus \mathcal{B}$ be the set of honest agents. The numbers of Byzantine and regular agents are $|\mathcal{B}|$ and $|\mathcal{H}|$, respectively.

The objectives of Byzantine agents may vary in different scenarios. One of them is that the Byzantine agents simply disturb the expected optimization process by sending random messages. Another is that the Byzantine agents collude to send crafted messages such that they can occupy more resources than needed. In turn, the honest agents shall be allocated with less resources. We consider the worst case in which the Byzantine agents are arbitrarily malicious. Therefore, an oracle goal for the honest agents is to solve

$$\begin{aligned} \min_{\hat{\boldsymbol{\theta}}} \quad & f\left(\hat{\boldsymbol{\theta}}\right) := \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} f_i(\boldsymbol{\theta}_i), \\ s.t. \quad & \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \boldsymbol{\theta}_i \le \boldsymbol{s}, \quad \boldsymbol{\theta}_i \in \boldsymbol{C}_i,\ \forall i \in \mathcal{H}. \end{aligned} \tag{2}$$

where $\hat{\boldsymbol{\theta}} \in \mathcal{R}^{|\mathcal{H}|D}$ collects all $\boldsymbol{\theta}_i$ of the regular agents $i \in \mathcal{H}$.

However, solving (2) is nontrivial since the number and identities of Byzantine workers are unknown in advance. In this paper, we aim at developing a Byzantine-resilient decentralized resource allocation algorithm that approximately solves (2).

3. DECENTRALIZED RESOURCE ALLOCATION WITHOUT BYZANTINE ATTACKS

Algorithm Development. When there are no Byzantine attacks, in this section we introduce a decentralized resource allocation (DRA) algorithm to solve (1) as the baseline.

Define the regularized Lagrangian function [5] of (1) as

$$\mathcal{L}_v(\boldsymbol{\theta}; \boldsymbol{\lambda}) := \mathcal{L}(\boldsymbol{\theta}; \boldsymbol{\lambda}) + \frac{v}{2}\sum_{i=1}^{N} \|\boldsymbol{\theta}_i\|^2 - \frac{v}{2}\|\boldsymbol{\lambda}\|^2, \tag{3}$$

where the Lagrangian function of (1) is

$$\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{\lambda}) := \frac{1}{N}\sum_{i=1}^{N} f_i(\boldsymbol{\theta}_i) + \left\langle \boldsymbol{\lambda}, \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\theta}_i - \boldsymbol{s} \right\rangle, \tag{4}$$

$\boldsymbol{\lambda} \in \mathbb{R}^D$ is the dual variable, and $v > 0$ is a regularization parameter. Thus, $\mathcal{L}_v(\cdot)$ is v-strongly convex and v-strongly concave in $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$, respectively.

Let k be the iteration index and γ^k be the step size at iteration k. To solve (1), at iteration k, each agent i performs the following projected gradient descent step on the primal variable and projected gradient ascent on the dual variable, as

$$\boldsymbol{\theta}_i^{k+1} = \mathcal{P}_{\boldsymbol{C}_i}\left(\boldsymbol{\theta}_i^k - \gamma^k \nabla_{\boldsymbol{\theta}_i} \mathcal{L}_v\left(\boldsymbol{\theta}^k; \boldsymbol{\lambda}^k\right)\right), \tag{5}$$

$$\boldsymbol{\lambda}^{k+1} = \left[\boldsymbol{\lambda}^k + \gamma^k \nabla_{\boldsymbol{\lambda}} \mathcal{L}_v\left(\boldsymbol{\theta}^k; \boldsymbol{\lambda}^k\right)\right]_+, \tag{6}$$

The gradients with respect to the primal and dual variables are respectively given by

$$\nabla_{\boldsymbol{\theta}_i} \mathcal{L}_v\left(\boldsymbol{\theta}^k; \boldsymbol{\lambda}^k\right) = \frac{1}{N}\left(\nabla_{\boldsymbol{\theta}_i} f_i\left(\boldsymbol{\theta}_i^k\right) + \boldsymbol{\lambda}^k\right) + v\boldsymbol{\theta}_i^k, \tag{7}$$

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L}_v\left(\boldsymbol{\theta}^k; \boldsymbol{\lambda}^k\right) = \left(\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\theta}_i^k - \boldsymbol{s}\right) - v\boldsymbol{\lambda}^k. \tag{8}$$

However, (5) and (6) cannot be implemented in a decentralized manner, since the dual variable is global, and the computation of dual gradient in (8) involves the average of all the local primal variables $\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\theta}_i^k$. To address these issues, we first let each agent i hold a

local dual variable $\boldsymbol{\lambda}_i \in \mathbb{R}^D$. Then, we assign each agent i an auxiliary variable $\boldsymbol{x}_i \in \mathbb{R}^D$ to track the value of $\frac{1}{N}\sum_{i=1}^N \boldsymbol{\theta}_i$. Motivated by the first-order decentralized dynamic average consensus method proposed in [28], $\boldsymbol{x}_i$ tracks $\frac{1}{N}\sum_{i=1}^N \boldsymbol{\theta}_i$ according to

$$\boldsymbol{x}_i^{k+1} = \sum_{j\in\mathcal{N}_i\cup\{i\}} w_{ij}\boldsymbol{x}_j^k + \triangle\boldsymbol{\theta}_i^{k+1}, \tag{9}$$

where w_{ij} is the weight of agents i and j, and $\triangle\boldsymbol{\theta}_i^{k+1} := \boldsymbol{\theta}_i^{k+1} - \boldsymbol{\theta}_i^k$. Collecting the weights in a doubly stochastic matrix $\boldsymbol{W} = [w_{ij}] \in \mathbb{R}^{N\times N}$, we require that $w_{ij} > 0$ if and only if $(i,j) \in \mathcal{E}$ or $i = j$.

With these introduced local variables $\boldsymbol{\lambda}_i$ and $\boldsymbol{x}_i$, in DRA each agent i modifies (5) and (6) to

$$\boldsymbol{\theta}_i^{k+1} = \mathcal{P}_{\boldsymbol{C}_i}\left(\boldsymbol{\theta}_i^k - \gamma^k\nabla_{\boldsymbol{\theta}_i}\mathcal{L}_v\left(\boldsymbol{\theta}^k;\boldsymbol{\lambda}_i^k\right)\right), \tag{10}$$

$$\boldsymbol{\lambda}_i^{k+1} = \left[\boldsymbol{\lambda}_i^k + \gamma^k\nabla_{\boldsymbol{\lambda}_i}\mathcal{L}_v\left(\boldsymbol{\theta}^k;\boldsymbol{\lambda}_i^k\right)|_{\frac{1}{N}\sum_{i=1}^N \boldsymbol{\theta}_i^k = \boldsymbol{x}_i^k}\right]_+. \tag{11}$$

$$\boldsymbol{x}_i^{k+1} = \sum_{j\in\mathcal{N}_i\cup\{i\}} w_{ij}\boldsymbol{x}_j^k + \triangle\boldsymbol{\theta}_i^{k+1}, \tag{12}$$

Failure of DRA under Byzantine Attacks. As we will demonstrate with numerical experiments, when the communication stage is normal, DRA can approximately solve (1) in a decentralized manner. However, applying DRA at the presence of Byzantine attacks will lead to undesirable outcomes. In iteration k of DRA, each agent i updates $\boldsymbol{x}_i^{k+1}$ to track $\frac{1}{N}\sum_{i=1}^N \boldsymbol{\theta}_i^{k+1}$ based on the messages $\boldsymbol{x}_j^k$ from its neighbors. Honest agent $j \in \mathcal{H}$ shall broadcast the true $\boldsymbol{x}_j^k$ to its neighbors. Yet, Byzantine agent $j \in \mathcal{B}$ broadcasts an arbitrary, wrong message $\boldsymbol{z}_j^k \in \mathbb{R}^D$, instead of $\boldsymbol{x}_j^k$. We formally define the message received by agent i from its neighbor j as

$$\boldsymbol{y}_j^k = \begin{cases} \boldsymbol{x}_j^k, \text{ if } j \in \mathcal{H}\cap\mathcal{N}_i, \\ \boldsymbol{z}_j^k, \text{ if } j \in \mathcal{B}\cap\mathcal{N}_i. \end{cases} \tag{13}$$

The existence of wrong messages will prevent the honest agents from obtaining desirable resource allocation strategies. For example, if Byzantine agent $j \in \mathcal{B}$ sends to its honest neighbor i a wrong message $\boldsymbol{z}_j^k$, which is much larger than the average amount of resource $\boldsymbol{s}$, then $\boldsymbol{x}_i^{k+1}$ computed by honest agent i from (12) is larger than $\boldsymbol{s}$. Therefore, its local dual variable $\boldsymbol{\lambda}_i^{k+2}$ will be larger than normal according to (11), and consequently, its primal variable $\boldsymbol{\theta}_i^{k+3}$ will be smaller than normal according to (10).

4. BYZANTINE-RESILIENT DECENTRALIZED RESOURCE ALLOCATION

As we have discussed in Section 3, when there exist Byzantine agents, their wrong messages will affect the optimization process. To address this issue, we propose a novel BREDA algorithm. Instead of directly aggregating the received messages with weighted average in DRA, BREDA adopts the coordinate-wise trimmed mean (CTM) [29], which is able to tolerate Byzantine attacks, to aggregate the received messages. To be specific, CTM requires to roughly estimate an upper bound b for the number of Byzantine neighbors of each honest agent. Then for each coordinate, each honest agent eliminates the smallest b and the largest b values in the messages received from their neighbors and average the remaining values for aggregation. This way, at iteration k and for coordinate d, honest agent i separates its set of neighbors $\mathcal{N}_i$ into three subsets, as

$$\mathcal{N}_{i,d}^{k^{\min}} = \underset{\mathcal{X}:\{\mathcal{X}\in\mathcal{N}_i,|\mathcal{X}|=b\}}{\arg\min} \sum_{j\in\mathcal{X}} \left[\boldsymbol{y}_j^k\right]_d, \tag{14}$$

$$\mathcal{N}_{i,d}^{k^{\max}} = \underset{\mathcal{X}:\{\mathcal{X}\in\mathcal{N}_i,|\mathcal{X}|=b\}}{\arg\max} \sum_{j\in\mathcal{X}} \left[\boldsymbol{y}_j^k\right]_d, \tag{15}$$

$$\mathcal{N}_{i,d}^k = \mathcal{N}_i \setminus \mathcal{N}_{i,d}^{k^{min}} \setminus \mathcal{N}_{i,d}^{k^{max}}. \tag{16}$$

Therefore, for each regular agent $i \in \mathcal{H}$, the updates of BREDA are given by

$$\boldsymbol{\theta}_i^{k+1} = \mathcal{P}_{\boldsymbol{C}_i}\left(\boldsymbol{\theta}_i^k - \gamma^k\nabla_{\boldsymbol{\theta}_i}\mathcal{L}_v\left(\boldsymbol{\theta}^k;\boldsymbol{\lambda}_i^k\right)\right), \tag{17}$$

$$\boldsymbol{\lambda}_i^{k+1} = \left[\boldsymbol{\lambda}_i^k + \gamma^k\nabla_{\boldsymbol{\lambda}_i}\mathcal{L}_v\left(\boldsymbol{\theta}^k;\boldsymbol{\lambda}_i^k\right)|_{\frac{1}{N}\sum_{i=1}^N \boldsymbol{\theta}_i^k = \boldsymbol{x}_i^k}\right]_+, \tag{18}$$

$$\left[\boldsymbol{x}_i^{k+1}\right]_d = \frac{1}{|\mathcal{N}_i| - 2b + 1}\sum_{j\in\mathcal{N}_{i,d}^k\cup\{i\}} \left[\boldsymbol{y}_j^k\right]_d + \left[\triangle\boldsymbol{\theta}_i^{k+1}\right]_d, \tag{19}$$

where (19) applies to $d = 1, \cdots, D$ in a coordinate-wise manner.

With particular note, we will show with numerical experiments that in BREDA, the local optimization variables of the honest agents will be close to their Byzantine-free optima. However, the Byzantine agents are still able to manipulate their local optimization variables, asking for more resources than needed. Therefore, after the agents reach their resource allocation strategies, the resource provider can collect all the local optimization variables, satisfy those with the smallest resource requirements, and partially satisfy those with the largest resource requirements. Although this extra step needs global information collections, the incurred communication burden is low and the local cost functions are kept private.

5. CONVERGENCE ANALYSIS

Due to the page limit, we only give the convergence results of DRA and BREDA, and leave the proofs to an extended version.

Assumption 1. *The sets $\boldsymbol{C}_i$ are closed, convex and bounded. The feasible sets of* (1) *and* (2) *are non-empty. The functions $f_i(\boldsymbol{\theta}_i)$ are convex, and have Lipschitz continuous gradients with constant L.*

Assumption 2. *The undirected graph $\mathcal{G}(\mathcal{J},\mathcal{E})$ is connected. The weight matrix $\boldsymbol{W}$ is doubly stochastic.*

Assumption 3. *For any regular agent $i \in \mathcal{H}$, denote N_i and B_i as its numbers of neighbors and Byzantine neighbors, and suppose $B_i \le b < \frac{N_i}{3}$. Consider a graph set $\mathcal{H}_\mathcal{G}$ whose elements are the subgraphs of $\mathcal{G}$ obtained by removing all edges of Byzantine agents, and removing any additional b incoming edges at each honest agent. Any subgraph $\mathcal{G}' \in \mathcal{H}_\mathcal{G}$ has at least one agent i^* which has a directed path to all agents in $\mathcal{G}'$. The path length is no more than $\tau_\mathcal{G}$.*

Assumption 1 is common in constrained optimization and applies to both DRA and BREDA. Assumption 2 is common in decentralized optimization and applies to DRA. Assumption 3 is for BREDA, requiring that the networks of the regular agents, even after CTM, are still able to disseminate messages [30, Assumption 4].

Denote $(\boldsymbol{\theta}_v^*, \boldsymbol{\lambda}_v^*)$ as the saddle point of (3). The following Theorem shows that DRA converges to the saddle point.

Theorem 1. *Consider the DRA updates* (10)*,* (12) *and* (11)*. Define a column vector*

$$V^k := \left[\|\boldsymbol{\theta}^k - \boldsymbol{\theta}_v^*\|; \sqrt{\sum_{i=1}^N \|\boldsymbol{x}_i^k - \frac{1}{N}\sum_{i=1}^N \boldsymbol{\theta}_i^k\|^2}; \sqrt{\sum_{i=1}^N \|\boldsymbol{\lambda}_i^k - \boldsymbol{\lambda}_v^*\|^2}\right].$$

If Assumptions 1 and 2 hold, then with a proper constant step size $\gamma^k = \gamma$, we have

$$\lim_{k\to+\infty} V^k = \mathbf{0}. \tag{20}$$

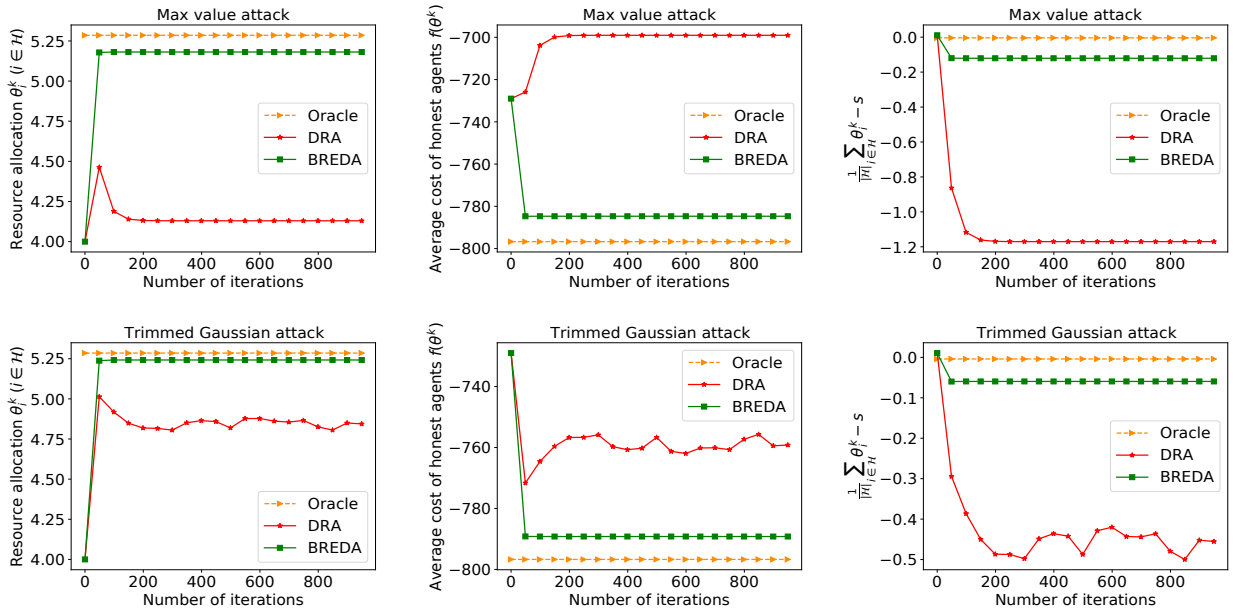

Fig. 1. Oracle without Byzantine attacks and DRA and BREDA under Byzantine attacks, when β_i are randomly distributed within $[1, 2]$. From top to bottom: max value attacks and trimmed Gaussian attacks. From left to right: resource allocation of a randomly chosen honest agent i, average cost of honest agents, and constraint violation of honest agents.

Define the regularized Lagrangian function [5] of (2) as

$$\hat{\mathcal{L}}_v\left(\hat{\boldsymbol{\theta}};\boldsymbol{\lambda}\right) := \hat{\mathcal{L}}\left(\hat{\boldsymbol{\theta}};\boldsymbol{\lambda}\right) + \frac{v}{2}\sum_{i\in\mathcal{H}}\|\boldsymbol{\theta}_i\|^2 - \frac{v}{2}\|\boldsymbol{\lambda}\|^2, \tag{21}$$

where the Lagrangian function of (1) is

$$\hat{\mathcal{L}}\left(\hat{\boldsymbol{\theta}};\boldsymbol{\lambda}\right) := \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} f_i\left(\boldsymbol{\theta}_i\right) + \left\langle \boldsymbol{\lambda}, \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\boldsymbol{\theta}_i - \boldsymbol{s}\right\rangle. \tag{22}$$

Denote $(\hat{\boldsymbol{\theta}}_v^*, \hat{\boldsymbol{\lambda}}_v^*)$ as the saddle point of (21). BREDA converges to a neighborhood of the saddle point.

Theorem 2. *Consider the BREDA updates* (17)*,* (19) *and* (18)*. Define a column vector*

$$V^k = \left[\|\hat{\boldsymbol{\theta}}^k - \hat{\boldsymbol{\theta}}_v^*\|; \sqrt{\sum_{i\in\mathcal{H}}\|\boldsymbol{\lambda}_i^k - \hat{\boldsymbol{\lambda}}_v^*\|^2}\right].$$

If Assumptions 1 and 3 hold, then with a proper two-stage diminishing step size γ^k, we have

$$\limsup_{k\to+\infty}\|V^k\| \le \phi, \tag{23}$$

where $\phi \in \mathbb{R}$ is determined by $\tau_{\mathcal{G}}$.

6. NUMERICAL EXPERIMENTS

This section presents numerical experiments to demonstrate the resilience of BREDA to various Byzantine attacks. More results on different cost functions and different fractions of Byzantine agents are left to the extended version. We consider the one-dimensional case such that $D = 1$. The upper bound of average resource is $\boldsymbol{s} = 5$. The local constraint of agent i is $\boldsymbol{\theta}_i \in \boldsymbol{C}_i = [0, 10]$. The local cost function of agent i is in the form of $f_i\left(\boldsymbol{\theta}_i\right) = -\alpha\beta_i \log\left(1 + \boldsymbol{\theta}_i\right)$ where $\alpha = 300$ is a constant and different agents i have different β_i. The variance of β_i reflects the heterogeneity of local cost functions.

We generate a connected random network consisting of $N = 100$ agents, but letting each agent have 15 neighbors. The maximum available amount of resources is 500, meaning that $\boldsymbol{s} = 5$. We randomly select $|\mathcal{B}|$ Byzantine agents, with $|\mathcal{B}| = 6$ by default. We test the performance of BREDA under two typical Byzantine attacks: max value and trimmed Gaussian attacks. With max value attacks, Byzantine agent $i \in \mathcal{B}$ sets its message as $z_i^k = 10$. For trimmed Gaussian attacks, Byzantine agent $i \in \mathcal{B}$ draws its message z_i^k from a Gaussian distribution with mean 7 and variance 9, followed by being trimmed to the range of $[0, 10]$. We set the lower and upper bounds since all the agents have the same local constraint $\boldsymbol{\theta}_i \in \boldsymbol{C}_i = [0, 10]$ such that the average demand of resource should also be within this range. We consider two baselines, Oracle and DRA. Oracle means that the Byzantine agents behave honestly, while DRA is subject to Byzantine attacks. In DRA, the weights $w_{ij} = \frac{1}{16}$ if and only if $(i, j) \in \mathcal{E}$ or $i = j$. The parameters γ and v are tuned to the best for Oracle, and then applied to DRA and BREDA. When β_i are randomly distributed within $[1, 2]$, $\gamma = 0.620$ and $v = 0.146$. The parameter of CTM is set as $b = 6$. Performance metrics include resource allocation strategy $\boldsymbol{\theta}_i^k$ for a randomly chosen honest agent i, average cost of honest agents $f(\hat{\boldsymbol{\theta}}^k)$, and constraint violation of honest agents $\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\boldsymbol{\theta}_i^k - \boldsymbol{s}$.

Fig. 1 depicts the performance of Oracle, DRA and BREDA when β_i are randomly distributed within $[1, 2]$. Observe that the resource allocation strategy of DRA is far from its oracle value, while that of BREDA is much closer, under both Byzantine attacks. In terms of average cost and constraint violation of honest agents, the gaps between BREDA and Oracle are acceptable, while those between DRA and Oracle are significant. For the two Byzantine attacks, max value attacks are stronger than trimmed Gaussian attacks.

7. REFERENCES

[1] F. P. Kelly, A. K. Maulloo, and D. K. Tan, Rate control for communication networks: Shadow prices, proportional fairness, and stability, Journal of the Operational Research Society, vol. 49, no. 3, pp. 237–252, 1998.

[2] B. Turan, C. A. Uribe, H. Wai, and M. Alizadeh, Resilient primal-dual optimization algorithms for distributed resource allocation, IEEE Transactions on Control of Network Systems, vol. 8, no. 1, pp. 282–294, 2021.

[3] L. Xiao and S. Boyd, Optimal scaling of a gradient method for distributed resource allocation, Journal of Optimization Theory and Applications, vol. 129, no. 3, pp. 469–488, 2006.

[4] H. Lakshmanan and D. P. De Farias, Decentralized resource allocation in dynamic networks of agents, SIAM Journal on Optimization, vol. 19, no. 2, pp. 911–940, 2008.

[5] J. Koshal, A. Nedic, and U. V. Shanbhag, Multiuser optimization: Distributed algorithms and error analysis, SIAM Journal on Optimization, vol. 21, no. 3, pp. 1046–1081, 2011.

[6] W. Lin, Y. Wang, C. Li, and X. Yu, Distributed resource allocation: An indirect dual ascent method with an exponential convergence rate, Nonlinear Dynamics, vol. 102, pp. 1685–1699, 2020.

[7] M. Zargham, A. Ribeiro, A. Ozdaglar, and A. Jadbabaie, Accelerated dual descent for network flow optimization, IEEE Transactions on Automatic Control, vol. 59, no. 4, pp. 905–920, 2014.

[8] E. Wei, A. Ozdaglar, and A. Jadbabaie, A distributed Newton method for network utility maximization – I: Algorithm, IEEE Transactions on Automatic Control, vol. 58, no. 9, pp. 2162–2175, 2013.

[9] E. Ghadimi, I. Shames, and M. Johansson, Multi-step gradient methods for networked optimization, IEEE Transactions on Signal Processing, vol. 61, no. 21, pp. 5417–5429, 2013.

[10] P. Yi, Y. Hong, and F. Liu, Initialization-free distributed algorithms for optimal resource allocation with feasibility constraints and application to economic dispatch of power systems, Automatica, vol. 74, pp. 259–269, 2016.

[11] A. Bedi and K. Rajawat, Asynchronous incremental stochastic dual descent algorithm for network resource allocation, IEEE Transactions on Signal Processing, vol. 66, no. 9, pp. 2229–2244, 2018.

[12] Z. Deng, S. Liang, and Y. Hong, Distributed continuous-time algorithms for resource allocation problems over weight-balanced digraphs, IEEE Transaction on Cybernetics, vol. 48, no. 11, pp. 3116–3125, 2018.

[13] Z. Deng, X. Nian, and C. Hu, Distributed algorithm design for nonsmooth resource allocation problems, IEEE Transactions on Cybernetics, vol. 50, no. 7, pp. 3208–3217, 2019.

[14] T. Chen, Q. Ling, and G. B. Giannakis, An online convex optimization approach to proactive network resource allocation, IEEE Transactions on Signal Processing, vol. 65, no. 24, pp. 6350–6364, 2017.

[15] X. Yi, X. Li, L. Xie, and K. Johansson, Distributed online convex optimization with time-varying coupled inequality constraints, IEEE Transactions on Signal Processing, vol. 68, pp. 731–746, 2020.

[16] L. Lamport, R. E. Shostak, and M. C. Pease, The Byzantine generals problem, ACM Transactions on Programming Languages and Systems, vol.4, no. 3, pp. 382–401, 1982.

[17] P. Yi, J. Lei, and Y. Hong, Distributed resource allocation over random networks based on stochastic approximation, Systems and Control Letters, vol. 114, pp. 44–51, 2018.

[18] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos, DRACO: Byzantine-resilient distributed training via redundant gradients, arXiv preprint arXiv:1803.09877, 2018.

[19] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In Proceedings of AAAI, 2019.

[20] Z. Wu, Q. Ling, T. Chen, and G. B. Giannakis, Federated variance-reduced stochastic gradient descent with robustness to Byzantine attacks, IEEE Transactions on Signal Processing, vol. 68, pp. 4583–4596, 2020.

[21] Z. Yang and W. U. Bajwa, ByRDiE: Byzantine-resilient distributed coordinate descent for decentralized learning, IEEE Transactions on Signal and Information Processing over Networks, vol. 5, no. 4, pp. 611–627, 2019.

[22] Z. Yang and W. U. Bajwa, BRIDGE: Byzantine-resilient decentralized gradient descent, arXiv preprint arXiv:1908.08098 2019.

[23] J. Li, W. Abbas, and X. Koutsoukos, Resilient distributed diffusion in networks with adversaries, IEEE Transactions on Signal and Information Processing over Networks, vol. 6, pp. 1–7, 2019.

[24] S. Sundaram and B. Gharesifard, Distributed optimization under adversarial nodes, IEEE Transactions on Automatic Control, vol. 64, no. 3, pp. 1063–1076, 2018.

[25] W. Ben-Ameur, P. Bianchi, and J. Jakubowicz, Robust distributed consensus using total variation, IEEE Transactions on Automatic Control, vol. 61, no. 6, pp. 1550–1564, 2015.

[26] W. Xu, Z. Li, and Q. Ling, Robust decentralized dynamic optimization at presence of malfunctioning agents, Signal Processing, vol. 153, pp. 24–33, 2018.

[27] J. Peng, W. Li, and Q. Ling, Byzantine-robust decentralized stochastic optimization over static and time-varying networks, Signal Processing, vol. 183, no. 108020, 2021.

[28] M. Zhu and S. Martinez, Discrete-time dynamic average consensus, Automatica, vol. 46, no. 2, pp. 322–329, 2014.

[29] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, Byzantine-robust distributed learning: Towards optimal statistical rates, In Proceedings of ICML, 2018.

[30] Z. Wu, H. Shen, T. Chen, and Q. Ling, Byzantine-resilient decentralized policy evaluation with linear function approximation, IEEE Transactions on Signal Processing, vol. 69, pp. 3839–3853, 2021.