UNICA UNIVERSITÀ DEGLI STUDI DI CAGLIARI

IRIS Università di Cagliari

UNICA IRIS Institutional Research Information System

This is the Author's *post-print* manuscript version of the following contribution:

M. Kaheni, E. Usai and M. Franceschelli, "Resilient Constrained Optimization in Multi-Agent Systems With Improved Guarantee on Approximation Bounds," in *IEEE Control Systems Letters*, vol. 6, pp. 2659-2664, 2022, doi: 10.1109/LCSYS.2022.3173495.

Abstract:
This letter considers resilient decentralized constrained optimization in multi-agent systems where some agents due to cyberattacks become adversaries. We show that the proposed method is resilient despite the persistent influence of up to F anonymous adversaries in the complete graphs. Our approach provides a better approximation of the optimal solution than the current literature. If the agents' objectives are 2F redundant, then the algorithm converges to the optimal solution. In addition to current literature, we consider a constrained optimization problem. Finally, we present numerical simulations to corroborate the theoretical analysis.

Keywords: Cyber-physical security, distributed optimization, multi-agent systems, resilient optimization.

- © 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

The publisher's version is available at: https://ieeexplore.ieee.org/document/9770946

When citing, please refer to the published version.

This full text was downloaded from UNICA IRIS https://iris.unica.it/

Resilient Constrained Optimization in Multi-Agent Systems with Improved Guarantee on Approximation Bounds

M. Kaheni, E. Usai, *Member, IEEE*, and M. Franceschelli, *Member, IEEE*

***Abstract*— This paper considers resilient decentralized constrained optimization in multi-agent systems where some agents due to cyberattacks become adversaries. We show that the proposed method is resilient despite the persistent influence of up to F anonymous adversaries in the complete graphs. Our approach provides a better approximation of the optimal solution than the current literature. If the agents' objectives are $2F$ redundant, then the algorithm converges to the optimal solution. In addition to current literature, we consider a constrained optimization problem. Finally, we present numerical simulations to corroborate the theoretical analysis.**

I. Introduction

Distributed optimization in multi-agent systems refers to a group of agents with individual objective functions that cooperatively agree on the global optimizer of the sum of these functions. The agents gather information from their neighbors in a graph and then update their values according to a state update rule [1]–[4].

As discussed in [5], distributed optimization methods can be easily disrupted by adversarial agents that can arise due to faults or cyberattacks. The methods proposed to deal with the destructive effects of adversarial agents on distributed optimization fall into two general categories. The first consists of methods that propose algorithms to detect and isolate the adversarial agents to prevent the outcome from being diverted to the attackers' desired value [6], [7]. The second category consists of methods that converge to a solution that the adversarial agents can not influence, thus without the need to detect and isolate [5], [8], [9]. The second category is referred to as *resilient distributed optimization*.

Many resilient protocols are based on mean-subsequence-reduced (MSR) algorithms. In these algorithms, regular agents filter and discard the most extreme values in their neighborhood before processing their state values. This method was initially introduced to obtain resiliency in consensus problems [10], [11] and is recently being developed to deal with distributed optimization [5], [9].

Consider the constrained optimization problem

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

where $x \in \mathbb{R}$, $i = 1, 2, \ldots, n$ represents agents in the network, X is the set of constraints and $f_i : \mathbb{R} \to \mathbb{R}$ is their corresponding cost function. The optimal solution of (1) in an adversarial environment is to ignore the adversarial agents and find the optimizer among the set V_r of regular agents with nominal behavior

$$\min_{x\in X} \sum_{i\in V_r} f_i(x). \tag{2}$$

The algorithms introduced in [5], [9], [12], instead of solving (2), find an approximate solution which consists in obtaining the optimal solution to the next problem:

$$\min_{x\in X} \sum_{i=1}^{n} \alpha_i f_i(x), \quad \sum_{i\in V_r} \alpha_i = 1, \quad \forall i \in V\backslash V_r \to \alpha_i = 0. \tag{3}$$

where the coefficients α_i are arbitrary and unknown. The problem in (3) provides a good approximation of (2) only when the objective functions of each agent are the same or very similar to each other. On the other hand, if each objective function f_i is different, i.e., if their minimum value occurs for different values of x, then the solution obtained from solving (3) can be significantly different from the solution of (2). We pose the next important question: is it possible to find the solution of (2) despite the presence of adversarial agents?

In [13], the authors define the concept of *redundancy* in distributed optimization problems. They prove that the necessary condition for finding the solution of (2) in a network with the possibility of the presence of up to F adversarial agents is that the agents' cost functions have the *2F-redundancy* property. *2F-redundancy* means that for any subset of regular agents, Ω, where $|\Omega| = n - 2F$ we have

$$\operatorname{argmin}_{x\in X} \sum_{i\in V_r} f_i(x) = \operatorname{argmin}_{x\in X} \sum_{i\in \Omega} f_i(x). \tag{4}$$

In [14], the authors consider 2F-redundant unconstrained problems in complete graphs and propose an algorithm to find the exact optimal solution of (2) in the presence of up to F adversarial agents. In contrast, we propose an algorithm that reaches the same results in constrained problems. In constrained problems, the updated values at each time step must be projected onto the constraint set X. The presence of the nonlinear projection function in the update rule makes the mathematical treatments more challenging.

The main contributions of this paper are

- A novel resilient decentralized algorithm for **constrained** optimization problems in complete graphs. The algorithm has common principles with MSR algorithms but deals instead with constrained optimization problems.

This work was supported by the Fondazione di Sardegna with grant "Formal Methods and Technologies for the Future of Energy Systems", n. F72F20000350007.

This work has been funded by "Fondazione di Sardegna" under the project "ARGOSAT- Microsatellite cluster for the observation of optical transients in Astronomy", CUP: F74I19001070007

Mojtaba Kaheni, Elio Usai, and Mauro Franceschelli are with the Department of Electrical and Electronic Engineering (DIEE), University of Cagliari, Italy. {mojtaba.kaheni, elio.usai, mauro.franceschelli}@unica.it

Mauro Franceschelli is the corresponding author.

- Proof of convergence to an approximate solution guaranteed to be bounded by a combination of $n-2F$ objectives of regular agents.
- Proof of convergence to the **optimal solution of** (2), if the 2F-redundancy property in (4) is satisfied.

The proposed algorithm can also be executed in a different scenario considering a star graph with a central coordinator for comparison purposes with [15].

II. Preliminaries

Let $G=(V,E)$ be a graph where $V=\{1,\ldots,n\}$ is the set of nodes (or agents) and $E\subseteq\{V\times V\}$ is the set of edges which represent interactions among them. If agent i receives information from j, then $(j,i)\in E$. A graph is undirected if $(j,i)\in E\Leftrightarrow(i,j)\in E$ and directed otherwise. A weighted graph is a graph where to each edge (i,j) is associated a weight $a_{ij}\geq 0$. The adjacency matrix is defined as $A=[a_{ij}]\in\mathbb{R}^{n\times n}$ where $a_{ij}>0$ if $(j,i)\in E$ and $a_{ij}=0$ otherwise. The set of in-neighbors of agent i is denoted by $N_{in}^{i}=\{j\in V|\,(j,i)\in E, j\neq i\}$ and the set of out-neighbors of agent i is similarly denoted as $N_{out}^{i}=\{j\in V|\,(i,j)\in E, j\neq i\}$. A path π_{ij} between agents i and j is a set of consecutive edges $\{(i,k_1),(k_1,k_2),\ldots,(k_m,j)\}\in E$ where $\{i,k_1,k_2,\ldots,k_n,j\}\in V$. An agent j is said to be *reachable* from agent i if there exists a directed path π_{ij} from agent i to agent j. A matrix $A\in\mathbb{R}^{n\times n}$ is called row stochastic if $\forall i\in V$, $\sum_{j=1}^{n}a_{ij}=1$ and column stochastic if $\forall j\in V$, $\sum_{i=1}^{n}a_{ij}=1$. If A is jointly row and column stochastic, then A is called doubly stochastic. Finally, a weighted graph $G(k)$ is time-varying if the edge weights $a_{ij}(k)$ are time-varying. Next, we recall some basic notions on graph connectivity.

Definition 1: A graph G is said to be rooted if there exist an agent $i\in V$ such that all agents in V are reachable from i. Agent i is called root agent.

Definition 2: A graph G is strongly connected if there exists a directed path between all pairs $i,j\in V$ of agents.

Definition 3: A graph $G(k)$ is jointly strongly connected if there exists a positive integer B such that for all $k_0\geq 0$, $\left(V,\cup_{k=0}^{B-1}E\left(k_0+k\right)\right)$ is strongly connected.

Definition 4 ([5]): A subset $S\subset V$ of agents is r-reachable if there exists $i\in S$ such that $\left|N_{in}^{i}\backslash S\right|\geq r$.

Definition 5 ([5]): G is said to be r-robust if for all pairs of disjoint nonempty subsets, $S_1,S_2\subset V$, at least one of S_1 or S_2 is r-reachable.

The following result is also helpful in our analysis.

Lemma 1 ([11]): Suppose G is r-robust and G' be a graph obtained by removing $r-1$ or fewer incoming edges from each agent in G. Then G' is rooted. ■

We can define the *transition matrix of a graph* by

$$\Phi\left(k,s\right)=A\left(k\right)A\left(k-1\right)\ldots A\left(s+1\right)A\left(s\right),\qquad(5)$$

for all s and k with $k\geq s$, and $\Phi\left(k,k\right)=A\left(k\right)$ which is an adjacency matrix. We can rewrite the conventional update rule in consensus based multi-agent systems in the form

$$\mathrm{x}\left(k+1\right)=\Phi\left(k,s\right)\mathrm{x}\left(s\right),\qquad(6)$$

where $\mathrm{x}\left(k\right)=\left[x_1\left(k\right),\ldots,x_n\left(k\right)\right]^T$ represents the agents' state at time step k. In general, as shown in (6), if by $\lim_{k\to\infty}\Phi(k,s)$, all the rows of the transition matrix become equal to an identical stochastic vector, convergence is achieved. Lemma 2 describes the conditions for convergence in terms of connectivity and adjacency matrix weights.

Lemma 2 ([16]): Let $a_{ii}\left(k\right)\geq\eta$ for all $i=1,\ldots,n$ and $a_{ij}=0$ or $a_{ij}\left(k\right)\geq\eta$, $i\neq j$ for a scalar η with $0<\eta<1$ holds. If $A\left(k\right)$ is row stochastic and rooted, then there exist constants $B>0$ and $0<\xi<1$, and a stochastic vector $\phi\left(s\right)=\left[\phi_1\left(s\right),\phi_2\left(s\right),\ldots,\phi_n\left(s\right)\right]^T$ such that $\lim_{k\to\infty}\Phi(k,s)=\mathbf{1}_{n\times 1}\phi(s)^T$ and $\left|\left[\Phi\left(k,s\right)\right]_{i,j}-\phi_j\left(s\right)\right|\leq B\xi^{k-s}$. ■

An essential step in constrained distributed optimizations algorithms is to project the updated value of each agent into the constraint space. Projection of a vector x on a closed convex set X is defined as $P_X\left[x\right]=\arg\min_{\bar{x}\in X}\|x-\bar{x}\|$. The following lemma establishes a relation between the projection error vector and the feasible directions of the convex set X at the projection vector.

Lemma 3 ([1]): Let $X\subset\mathbb{R}^n$ be a nonempty closed convex set, we have for any $x\in\mathbb{R}^n$ and $y\in X$

$$\left\|P_X\left[x\right]-y\right\|^2\leq\left\|x-y\right\|^2-\left\|P_X\left[x\right]-x\right\|^2.$$ ■

Our algorithm in this paper improves upon the algorithm introduced in [1] where the authors prove that if the agents start from arbitrary initial states and evolve according to

$$\begin{array}{rcl} v_i\left(k\right) & = & \sum_{j=1}^{n}a_{ij}\left(k\right)x_j\left(k\right),\\ x_i\left(k+1\right) & = & P_X[v_i\left(k\right)-c\left(k\right)d_i\left(k\right)],\end{array}\qquad(7)$$

where $x_i(k)$ is the state value and gradient of agent i at time k, $d_i(k)$ is the gradient of $f_i(x)$ for $x=x_i(k)$, and $c(k)$ is a step size that satisfies

$$c\left(k\right)>0,\quad\sum_{k=0}^{\infty}c\left(k\right)=\infty,\quad\sum_{k=0}^{\infty}c^2\left(k\right)<\infty,\qquad(8)$$

for instance, $c(k)=1/k$, then in convex optimization problems with bounded $d_i(k)$, $x_i\left(k\right)$ converges to the optimal solution of (1) for all $i\in V$. We report this result in the next lemma.

Lemma 4 ([1]): Let $a_{ii}\left(k\right)\geq\eta$ for all $i=1,\ldots,n$ and $a_{ij}\left(k\right)=0$ or $a_{ij}\left(k\right)\geq\eta$, with $i\neq j$, for $\eta\in(0,1)$ and let graph G be jointly strongly connected and associated to a doubly stochastic adjacency matrix. If $\forall i\in V$, f_i is convex, $d_i(k)$ is bounded and X is nonempty, closed and convex, then the agents that evolve according to (7)*, converge to an optimal point of* (1)*, x^*, such that for all $i\in V$, $\lim_{k\to\infty}||x_i\left(k\right)-x^*||=0$.* ■

III. Problem statement and main results

We consider a multi-agent framework to solve the following constrained optimization problem

$$\min_{x\in X}\sum\nolimits_{i=1}^{n}f_i(x),\qquad(9)$$

and we consider up to F agents which distribute false state information at all times. These agents are unknown during the execution of our algorithm. Throughout this paper, we

call these agents as *adversarial agents* and denote their set as V_a. An agent which exhibits adversarial behavior at future time instants is still counted as an adversarial agent in our analysis at all times. We assume that (9) satisfies the next standard assumptions in distributed optimization.

Assumption 1: For all $i \in \{1,\dots,n\}$ the local objective function, f_i, is convex. In addition, The set of constraints, X, is nonempty, closed and convex. ■

Assumption 2: For all $i \in \{1,\dots,n\}$, the gradient $d_i(x)$ of each objective f_i is continuous and bounded by

$$|d_i(x)| \leq L, \quad \forall i \in V. \qquad \blacksquare \qquad (10)$$

Next, we make a specific assumption that appears in several papers on resilient distributed optimization [14], [17].

Assumption 3: Each agent can interact with all others (complete graph). This does not imply that it does because some agents are discarded and thus the state updates actually evolve according to a directed graph. ■

Our main results are Algorithm 1 and its convergence analysis, which solves our stated problem. The main idea behind Algorithm 1 is that each agent discards extreme values received from its in-neighbors. Theorem 1 and Corollary 1 formalize the main results of this paper.

Theorem 1: Consider the optimization problem (9) *and a multi-agent system that executes Algorithm 1. Consider the power set $P(V_r)$ of regular agents V_r, which are indeed unknown during the execution of Algorithm 1, and let $V_\Omega = \{\Omega \in P(V_r) | |\Omega| = n - 2F\}$ be the set of all possible subsets of regular agents of cardinality $n-2F$.*

Let Assumptions 1-3 be satisfied. If the graph is $2F+1$-robust and the number of adversaries is equal or less than F, then the regular agents converge to an approximate solution $\bar{x}$ bounded by the minimum of an objective function that is guaranteed to be a combination of $n-2F$ objectives of regular agents, thus such that $x_m \leq \bar{x} \leq x_M$, where

$$x_m = \min_{\Omega \in V_\Omega} \left\{ \operatorname*{argmin}_{x \in X} \sum_{i \in \Omega} f_i(x) \right\},$$
$$x_M = \max_{\Omega \in V_\Omega} \left\{ \operatorname*{argmin}_{x \in X} \sum_{i \in \Omega} f_i(x) \right\}. \qquad \blacksquare$$

Next, we identify an interesting case where Algorithm 1 converges to the actual optimal solution of (9).

Corollary 1: Under the assumptions of Theorem 1 and the 2F-redundancy condition in (4)*, the solution $\bar{x}$ achieved by executing Algorithm 1 satisfies $\bar{x} \in x^\star$ where $x^\star$ is the set of optimal solutions of* (9). ■

Proof: By the 2F-redundancy condition (4), it follows that $x_m, x_M \in x^\star$. Therefore from Assumption 1, since $x_m \leq \bar{x} \leq x_M$, it holds $\bar{x} \in x^\star$. ■

IV. Proof of Theorem 1

In this section, we present the proof of Theorem 1 together with a few helpful lemmas. First, note that the state values of agents in sets $V_h(k)$ and $V_l(k)$ are discarded by the others. Thus, the actual interaction graph during the execution of Algorithm 1 is never complete nor it is encoded by a doubly stochastic adjacency matrix and therefore Lemma 4 cannot be used straightforwardly. According to Definition 5

Algorithm 1: Resilient Decentralized Constrained Optimization

1: Initialization: $x_i(0) \in X, \forall i \in V$
2: **Repeat**
3: Each agent gathers the state value of all others and builds the next three sets of neighbors:
$V_h(k) = \{F \text{ vertices with highest values}\}$,
$V_l(k) = \{F \text{ vertices with lowest values}\}$,
$V_n(k) = \{\text{Other vertices}\}$.
4: Each agent assigns weights to edges according to: $(q = n - 2F)$
If $i \in V_n(k)$ then $a_{ij}(k) = \begin{cases} \frac{1}{q} & \text{if} \quad j \in V_n(k) \\ 0 & \text{otherwise.} \end{cases}$
Elseif $i \in V_h(k) \cup V_l(k)$, then
$a_{ij}(k) = \begin{cases} \frac{1}{q} & \text{if} \quad j \in \{V_n(k) \setminus i_r^k\} \\ \frac{1}{q} & \text{if} \quad j = i \\ 0 & \text{otherwise.} \end{cases}$
where i_r^k is the vertex with lowest state value in $V_n(k)$ if $i \in V_l(k)$ and the vertex with highest state value in $V_n(k)$ if $i \in V_h(k)$.
5: $v_i(k) = \sum_{j=1}^{n} a_{ij}(k)\, x_j(k)$.
6: $x_i(k+1) = P_X[v_i(k) - c(k)\, d_i(k)]$.
7: **Until stopping criterion is satisfied.**

a complete graph is $2F+1$ robust if $\lfloor \frac{n+1}{2} \rfloor \geq 2F+1$. We notice that each agent discards at most $2F$ of its incoming edges in Algorithm 1. From Lemma 1, $2F+1$ robustness makes us sure that the resulting graph after implementing Algorithm 1 is rooted. In the next lemma, we prove the important result that regular agents converge to consensus after the implementation of Algorithm 1, and the value of consensus is independent of the state of adversarial agents.

Lemma 5: Let Assumptions 1 and 2 hold. If the graph is $2F+1$ robust, all regular agents in the network converge to consensus by executing Algorithm 1,

$$\lim_{k \to \infty} \|x_{i_1}(k) - x_{i_2}(k)\| = 0, \quad \forall i_1, i_2 \in V_r,$$

and the consensus value is independent of the state of adversarial agents.

Proof: First, consider the case that there is no adversarial agent in the ball with $q = n - 2F$ normal state agents. The update rule of regular agents (7) can be written as

$$x_i(k+1) = v_i(k) - c(k)\, d_i(k) + e_i(k), \qquad (11)$$

where $e_i(k) = P_X[v_i(k) - c(k)\, d_i(k)] - (v_i(k) - c(k)\, d_i(k))$. According to Step 5 of Algorithm 1, we can rewrite (11) in the following form

$$x_i(k+1) = \sum\nolimits_{j=1}^{n} a_{ij} x_j(k) - c(k)\, d_i(k) + e_i(k). \quad (12)$$

Then using the transition matrix defined in (5), for all $i \in V$ and $k > s$ we have

$$\begin{aligned} x_i(k+1) &= \textstyle\sum_{j=1}^{n} [\Phi(k,s)]_{i,j} x_j(s) \\ &+ \sum_{r=s}^{k-1} \sum_{j=1}^{n} [\Phi(k, r+1)]_{i,j} (e_j(r) - c(r)\, d_j(r)) \qquad (13) \\ &+ e_i(k) - c(k)\, d_i(k). \end{aligned}$$

According to (7), $x_i(k) \in X$ for all $k \geq 0$ and $\forall i \in V$, then $v_i(k) \in X$ and from Lemma 3 we have

$$\begin{aligned} &\|x_i(k+1) - v_i(k)\|^2 \leq \|c(k)\, d_i(k)\|^2 - \\ &\|x_i(k+1) - (v_i(k) - c(k)\, d_i(k))\|^2 \qquad (14) \\ &= \|c(k)\, d_i(k)\|^2 - \|e_i(k)\|^2. \end{aligned}$$

Considering (10), yields to $-Lc(k) \leq e_i(k) \leq Lc(k)$. Thus, noticing (8), $\lim_{k\to\infty} c(k)\, d_i(k) = 0$ and $\lim_{k\to\infty} e_i(k) = 0$ for all regular agents. From (13) we have

$$\begin{aligned}
&\lim_{k\to\infty} \{x_{i_1}(k+1) - x_{i_2}(k+1)\} = \\
&\lim_{k\to\infty} \sum_{j=1}^{n} \left([\Phi(k,s)]_{i_1,j} - [\Phi(k,s)]_{i_2,j} \right) x_j(s) + \\
&\lim_{k\to\infty} \sum_{r=s}^{k-1} \sum_{j=1}^{n} \left([\Phi(k,r+1)]_{i_1,j} - [\Phi(k,r+1)]_{i_2,j} \right) \dots \\
&(e_j(r) - c(r)\, d_j(r)).
\end{aligned} \tag{15}$$

By $2F+1$ robustness property and Lemma 1, the resulting graph after implementing Algorithm 1 is row stochastic and rooted thus from Lemma 2 we have for all $i \in V_r$, $\lim_{k\to\infty} [\Phi(k,s)]_{i,j} = \varphi_j(s)$, yields

$$\lim_{k\to\infty} \left\{ [\Phi(k,s)]_{i_1,j} - [\Phi(k,s)]_{i_2,j} \right\} = 0. \tag{16}$$

Therefore, the first term of (15) tends to zero. Noticing that $[\Phi(k,s)]_{i,j}$, $e_i(k)$ and $c(k)\, d_i(k)$ are bounded for all i, j, k and s, in the second term, either $r < \infty$ which from Lemma 2, makes $\lim_{k\to\infty} \left\{ [\Phi(k,r+1)]_{i_1,j} - [\Phi(k,r+1)]_{i_2,j} \right\} = 0$ with geometric rate or $r = \infty$ which makes $e_j(r) - c(r)\, d_j(r) = 0$. Thus, the second term also tends to zero (see Lemma 7 in [1]). As result we have

$$\lim_{k\to\infty} \|x_{i_1}(k) - x_{i_2}(k)\| = 0, \quad \forall i_1, i_2 \in V_r. \tag{17}$$

If there are $K \leq F$ adversarial agents in $V_n(k)$, we must have at least K regular agents in $V_h(k)$ and K regular agents in $V_l(k)$. Thus, the state of adversarial agents is a linear combination of two regular agents.

$$x_{a_i}(k) = \sigma_i(k)\, x_{h_i}(k) + (1 - \sigma_i(k))\, x_{l_i}(k). \tag{18}$$

For $i = 1, \dots, K$ and $0 < \sigma_i(k) < 1$ where $x_{h_i}(k)$ and $x_{l_i}(k)$ represent regular agents that belong to $V_h(k)$ and $V_l(k)$, respectively. Thus, $G(k)$ is equivalent to a row stochastic graph in which two filtered regular agents replace each of the remaining adversarial agents in $V_n(k)$. It can be considered that instead of the adversarial agent state, the states of these two regular agents are communicated to its neighbors. Therefore, $G(k)$ is equivalent to a row stochastic graph in which the adversarial agents are filtered. ■

Next, we bring some helpful lemmas for proving our main result.

Lemma 6: Consider that $i_1 \in V_l(k)$ and $i_2 \in V_n(k)$. If i_1 and i_2 use the update rule in Algorithm 1 and $d_{i_1}(k) \geq d_{i_2}(k)$, then $x_{i_1}(k+1) \leq x_{i_2}(k+1)$. Similarly $i_2 \in V_n(k)$, $i_3 \in V_h(k)$ and $d_{i_2}(k) \geq d_{i_3}(k)$ yields to $x_{i_2}(k+1) \leq x_{i_3}(k+1)$.

Proof: By Step 4 in Algorithm 1

$$v_{i_1}(k) - v_{i_2}(k) = \frac{1}{q} \left(x_{i_1}(k) - x_{i_r^k}(k) \right), \tag{19}$$

where $x_{i_r^k}(k) \in V_n(k)$ is the state value of removed agent from $V_n(k)$. Considering the definition of $V_l(k)$ and $V_n(k)$ we have $x_{i_1}(k) - x_{i_r^k}(k) < 0$. Therefore $v_{i_1}(k) < v_{i_2}(k)$. By $d_{i_1}(k) \geq d_{i_2}(k)$ we conclude

$$v_{i_1}(k) - c(k)\, d_{i_1}(k) < v_{i_2}(k) - c(k)\, d_{i_2}(k). \tag{20}$$

Noticing (7), since projection on to the one-dimensional space X is non decreasing function, $x_{i_1}(k+1) \leq x_{i_2}(k+1)$.

With the same argumentation, $x_{i_2}(k+1) \leq x_{i_3}(k+1)$. The desired result is concluded. ■

Lemma 7: Consider the interval X is defined as $l_b \leq x \leq u_b$. If $e_i(k) \neq 0$, $\operatorname{sgn}\{e_i(k)\} = \operatorname{sgn}\{c(k)d_i(k)\}$, where $\operatorname{sgn}\{.\}$ represents the sign function.

Proof: Since $v_i(k) \in X$ we can define $r_1(k), r_2(k) \geq 0$, where $r_1(k) = u_b - v_i(k), \quad r_2(k) = v_i(k) - l_b$.

One can notice that $c(k)d_i(k) = 0 \Rightarrow e_i(k) = 0$, therefore we are only interested in cases where $c(k)d_i(k) \neq 0$. If $c(k)d_i(k) > 0$, $v_i(k) - c(k)\, d_i(k)$ moves toward l_b. If $c(k)d_i(k) \leq r_2(k)$, $e_i(k) = 0$. In case that $c(k)d_i(k) > r_2(k)$ we have

$$e_i(k) = c(k)d_i(k) - r_2(k) > 0. \tag{21}$$

In addition, if $c(k)d_i(k) < 0$, $v_i(k) - c(k)\, d_i(k)$ moves toward u_b. If $-c(k)d_i(k) \leq r_1(k)$, $e_i(k) = 0$. In case that $-c(k)d_i(k) > r_1(k)$ we have

$$e_i(k) = c(k)d_i(k) + r_1(k) < 0. \tag{22}$$

The desired result is achieved. ■

Lemma 8: Under Assumptions 1 and 2, for all regular agents that execute Algorithm 1 there exist a time step T, $T < \infty$ when for all $i_1 \in V_l(T)$, $i_2 \in V_n(T)$ and $i_3 \in V_h(T)$, $d_{i_1}(T) > d_{i_2}(T) > d_{i_3}(T)$, i.e., they are sorted according to their gradients.

Proof:

Consider that at time step s, $d_{i_1}(s) < d_{i_2}(s)$ and $x_{i_1}(s) < x_{i_2}(s)$. If $e_{i_1}(s), e_{i_2}(s) \neq 0$, with the same argument that we used in the proof of Lemma 7, $x_{i_1}(s+1)$ and $x_{i_2}(s+1)$ only can have 3 set of values

$$\begin{cases} x_{i_1}(s+1) = x_{i_2}(s+1) = l_b \\ x_{i_1}(s+1) = x_{i_2}(s+1) = u_b \\ x_{i_1}(s+1) = u_b, \quad x_{i_2}(s+1) = l_b \end{cases}, \tag{23}$$

Thus, $x_{i_2}(s+1) \leq x_{i_1}(s+1)$ and the desired result is achieved.

If $e_{i_1}(s) \neq 0$, $e_{i_2}(s) = 0$ and $d_{i_1}(s) < 0$ then $x_{i_1}(s+1) = u_b$ and also if $e_{i_1}(s) = 0$, $e_{i_2}(s) \neq 0$ and $d_{i_2}(s) > 0$ then $x_{i_2}(s+1) = l_b$ and the desired result is achieved.

As in (18), if there exists any adversarial agent in $V_n(k)$, the respective state can be replaced with two regular agents belonging to $V_h(k)$ and $V_l(k)$. Notice that the adjacency matrix of the equivalent graph is row stochastic. By virtue of Lemma 5, regular agents converge to consensus, therefore for $k \to \infty$ $x_i(k) = v_i(k)$, also by considering (11) and (8) along with (14) we have $v_i(k) = x_i(k+1)$. Thus $x_i(k) = v_i(k) = x_i(k+1)$ and the consensus value of regular agents converges to a constant. So, from Assumption 2, $d_i(k)$, the gradient of $f_i(.)$ at $x_i(k)$, converges to a constant. Assume s is large enough that the order of gradients does not change for $\forall k \geq s$, i.e., it holds $\forall i, j$, if $d_i(s) > d_j(s)$

then $d_i(k) > d_j(k)$, $\forall k \geq s$. In case that $e_{i_1}(k-1) \neq 0$, $e_{i_2}(k-1) = 0$ and $d_{i_1}(k-1) > 0$ then

$$\begin{aligned} &x_{i_2}(k) - x_{i_1}(k) = v_{i_2}(k-1) - v_{i_1}(k-1) \\ &-c(k-1)\left(d_{i_2}(k-1) - d_{i_1}(k-1)\right) - e_{i_1}(k-1). \end{aligned} \tag{24}$$

and from Lemma 7, $e_{i_1}(k-1) > 0$. Similarly if $e_{i_1}(k-1) = 0$, $e_{i_2}(k-1) \neq 0$ and $d_{i_2}(k-1) < 0$ then

$$\begin{aligned} &x_{i_2}(k) - x_{i_1}(k) = v_{i_2}(k-1) - v_{i_1}(k-1) \\ &-c(k-1)\left(d_{i_2}(k-1) - d_{i_1}(k-1)\right) + e_{i_2}(k-1), \end{aligned} \tag{25}$$

and from Lemma 7, $e_{i_2}(k-1) < 0$. In case that both $e_{i_1}(k-1), e_{i_2}(k-1) = 0$,

$$\begin{aligned} &x_{i_2}(k) - x_{i_1}(k) = v_{i_2}(k-1) - v_{i_1}(k-1) \\ &-c(k-1)\left(d_{i_2}(k-1) - d_{i_1}(k-1)\right). \end{aligned} \tag{26}$$

According to Step 4 of Algorithm 1, $v_{i_2}(k-1) - v_{i_1}(k-1) = \frac{1}{q}(x_{i_r^{k-1}}(k-1) - x_{i_1}(k-1))$. Therefore, considering (24), (25) and (26), $\forall k > s$ we have

$$\begin{aligned} &x_{i_2}(k) - x_{i_1}(k) \leq \tfrac{1}{q}\left(x_{i_r^{k-1}}(k-1) - x_{i_1}(k-1)\right) \\ &-c(k-1)\left(d_{i_2}(k-1) - d_{i_1}(k-1)\right). \end{aligned} \tag{27}$$

Noticing that $i_r^{k-1} \in V_n(k-1)$ we have

$$\begin{aligned} &x_{i_r^{k-1}}(k-1) - x_{i_1}(k-1) \leq \tfrac{1}{q}\left(x_{i_r^{k-2}}(k-2) - x_{i_1}(k-2)\right) \\ &-c(k-2)\left(d_{i_r^{k-1}}(k-2) - d_{i_1}(k-2)\right). \end{aligned} \tag{28}$$

From step 6 of Algorithm 1, we find that if $i_r^k \in V_n(k-1)$ or $i_r^k \in V_h(k-1)$, $d_{i_r^k}(k-1) \geq d_{i_2}(k-1)$. Otherwise with similar argumentation we mentioned in the proof of Lemma 6, $x_{i_2}(k) < x_{i_r^k}(k)$, which is a contradiction. Consider that we start from time step s and go forward. If at each time step k, $i_r^k \in V_l(k-1)$, according to Lemma 6, an agent with higher gradient in $V_n(k-1)$ or $V_h(k-1)$ has replaced i_r^k in $V_l(k)$. This replacement is an step toward sorting agents and we reinitialize $s = k$ and go forward again. It must be noted that since the number of agents is bounded and all the replacement are toward sorting agents, we have finite time steps in which $i_r^k \in V_l(k-1)$. Consider time step s in which $i_r^k \in V_n(k-1)$ or $i_r^k \in V_h(k-1)$ for all $k > s$. If we define $D(k) = d_{i_2}(k) - d_{i_1}(k) > 0$, $d_{i_r^{k+1}}(k) - d_{i_1}(k) \geq D(k)$. Let

$$\begin{aligned} D &= \inf_{k \geq s}\left\{d_{i_r^{k+1}}(k) - d_{i_1}(k)\right\} > 0, \\ C &= \sup_{k \geq s}\left\{x_{i_r^k}(k) - x_{i_1}(k)\right\} > 0. \end{aligned} \tag{29}$$

from (28) we can recursively find out that for $k > s$,

$$x_{i_2}(k) - x_{i_i}(k) \leq \frac{C}{q^{(k-s)}} - D\sum_{r=1}^{k-s}\frac{c(k-r)}{q^{r-1}}. \tag{30}$$

We are going to prove that the first term tends to zero with a faster rate, therefore there should exist a time step $T < \infty$ such that $x_{i_2}(T) - x_{i_i}(T) < 0$.

$$\begin{aligned} &\lim_{k\to\infty}\left\{\left(D\sum_{r=1}^{k-s}\tfrac{c(k-r)}{q^{r-1}}\right)/\tfrac{C}{q^{(k-s)}}\right\} = \\ &\tfrac{D}{C}\lim_{k\to\infty}\sum_{r=s}^{k-1} q^{(r+1-s)}c(r) \geq \tfrac{D}{C}\lim_{k\to\infty}\sum_{r=s}^{k-1} c(r) = \infty. \end{aligned} \tag{31}$$

We can write a same argumentation for $d_{i_2}(s) < d_{i_3}(s)$ and $x_{i_2}(s) < x_{i_3}(s)$. The desired result is achieved. ■

Now we are ready to discuss the proof of Theorem 1.

Proof of Theorem 1:

Lemma 6 and Lemma 8 show that there exists a time step T in which the states of the regular agents are sorted according to their gradients for all $k > T$, i.e., the agents with the largest positive gradients have the smallest state value and vice versa. Consider $\Omega_{j_1}, \Omega_{j_2} \in V_\Omega$ that contain $q = n - 2F$ regular agents with the largest and the smallest state values, respectively, for $k > T$. Let us define

$$\overline{M} = \operatorname*{argmin}_{x\in X}\sum_{i\in\Omega_{j_1}} f_i(x), \quad \underline{m} = \operatorname*{argmin}_{x\in X}\sum_{i\in\Omega_{j_2}} f_i(x). \tag{32}$$

The network executing Algorithm 1 which satisfies the assumptions of Lemma 5, achieves consensus, despite malicious agents. To prove Theorem 1, we show that the consensus value achieved, $\bar{x}$, satisfies $\underline{m} \leq \bar{x} \leq \overline{M}$.

Consider the interval X, as defined in Lemma 7. Since $\bar{x} \in X$, If $\overline{M} = u_b$, then $\bar{x} \leq \overline{M}$. Suppose by way of contradiction that $l_b \leq \overline{M} < u_b$, and $\bar{x} = \overline{M} + \epsilon$, $\epsilon > 0$. Let us define

$$M(k) = \textstyle\sum_{i\in\Omega_{j_1}} x_i(k), \quad m(k) = \sum_{i\in\Omega_{j_2}} x_i(k). \tag{33}$$

Since the number of adversaries is bounded by F, and recalling $|\Omega_{j_1}| = q = n - 2F$, there must be at least F regular agents with smaller state values in the network that do not belong to Ω_{j_1}. By the results we obtained in Lemma 6 and Lemma 8, the ordering of the states of regular agents will not change. Therefore, $V_l(k) \cap \Omega_{j_1} = \emptyset$ for all $k > T$. By a similar argument, we have $V_h(k) \cap \Omega_{j_2} = \emptyset$ for all $k > T$. Thus, according to the agents state update procedure (Algorithm 1, step 5) it follows $\frac{1}{q}m(k) \leq v_i(k) \leq \frac{1}{q}M(k)$. Thus considering (11), for all regular agents that execute Algorithm 1 we have

$$x_i(k+1) \leq \frac{1}{q}M(k) - c(k)d_i(k) + e_i(k). \tag{34}$$

Using (34) yields for $M(k+1)$

$$M(k+1) \leq M(k) - \textstyle\sum_{i\in\Omega_{j_1}}(c(k)d_i(k) + e_i(k)). \tag{35}$$

Since we assumed $\bar{x} = \overline{M} + \epsilon$, and noticing that by Lemma 5 all regular agents converge to $\bar{x}$, consider a time step $k_0 \geq T$, in which the following conditions are satisfied: 1) $\overline{M} + \frac{1}{2}\epsilon \leq x_i(k) \leq \overline{M} + 2\epsilon$, for $k \geq k_0$, which leads to $q(\overline{M} + \frac{1}{2}\epsilon) \leq M(k) \leq q(\overline{M} + 2\epsilon)$ and 2) $Lc(k) \leq \frac{1}{2}\epsilon$, for $k \geq k_0$. If these two conditions are satisfied, given the definition of $e_i(k)$ in (11), it follows $e_i(k) \leq 0$, for all $i \in V_r$ and $k \geq k_0$. Therefore, (35) can be rewritten as

$$M(k+1) \leq M(k) - c(k)\textstyle\sum_{i\in\Omega_{j_1}} d_i(k). \tag{36}$$

Starting from k_0 and iterating Z times, leads to

$$M(k_0 + Z) \leq M(k_0) - \sum_{k=k_0}^{Z-1}\left(c(k)\sum_{i\in\Omega_{j_1}} d_i(k)\right). \tag{37}$$

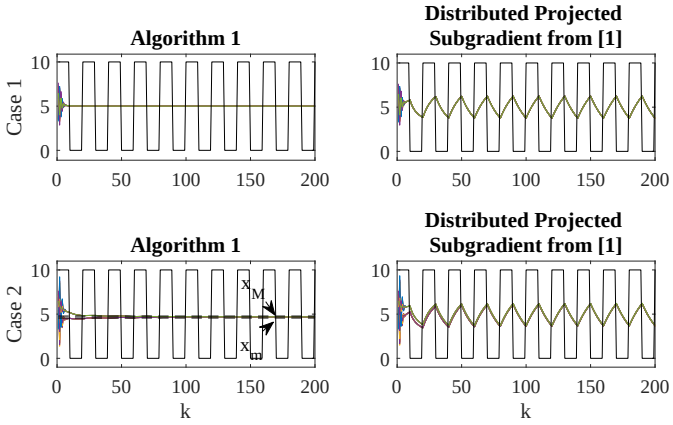


Fig. 1. State of agents in Case 1 and Case 2 by implementing Algorithm 1 (in the left), implementing distributed projected subgradient descent in [1], (in the right), and the state of the adversary depicted in black.

Recall that $l_b \leq \overline{M} < u_b$. According to (32), and the convexity assumption (Assumption 1), $\sum_{i\in\Omega_{j_1}} d_i(k) \geq 0$ if $x_i(k) = \overline{M}$, for all $i \in \Omega_{j_1}$. Since in convex functions, the first derivative is increasing with respect to argument, if for all $i \in \Omega_{j_1}$, $x_i(k) > \overline{M}$, we have $\sum_{i\in\Omega_{j_1}} d_i(k) > 0$. From (8) we notice that $\sum_{k=k_0}^{\infty} c(k) = \infty$. Therefore, according to (37) there exists a large enough Z, where $M(k_0+Z) < q(\overline{M} + \frac{1}{2}\epsilon)$, which is a contradiction, proving that $\bar{x} \leq \overline{M}$. With same argumentation, one can reach $\underline{m} \leq \bar{x}$. ■

V. Numerical example

Our goal is to optimize the scalar function as in [3]

$$G(x) = \sum_{i=1}^{n} \rho_i (x - u_i)^2, \; X = \{x \in \mathbb{R} : 0 \leq x \leq 10\}, \tag{38}$$

where u_i is only known by agent i. The agents are attempting to measure $\hat{x}$ and the agent i measures $u_i = \hat{x} + w_i$ where w_i are jointly Gaussian and zero mean and ρ_i represent the inverse of the variance of w_i. The maximum likelihood estimate is the optimizer of $G(x)$. In this simulation we consider a graph with 20 agents, and choose parameter $F = 1$. We consider that $\rho_i = 1$, for $i = 1, \ldots, n$, $c(k) = 1/(k+1)$, $3 \leq x_i(0) \leq 8$ are uniformly random values and we assume that the adversarial agent broadcasts $x_a(k) \in \{0, 10\}$ like a pulse generator. We simulate this network in 2 cases as follows:

- Case 1: ($2F - redundant$ case) All regular agents with $u_i = 5$;
- Case 2: 11 regular agents with $u_i = 1$, 8 regular agents with $u_i = 9$.

The corresponding x_m and x_M are both 5 in Case 1 ($2F - redundant$). Therefore, as the simulation results show, Algorithm 1 converges to 5 which is the optimal solution of (2). In Case 2, $x^\star = 4.685$, $x_m = 4.56$ and $x_M = 4.77$. As depicted in Fig. 1, the regular agents perfectly converges to $x_m \leq \bar{x} \leq x_M$ whereas the distributed subgradient descent introduced in [1], is affected by adversary in both cases.

VI. Conclusion

In this paper we presented a novel resilient decentralized algorithm for constrained optimization problems in complete graphs. We proved convergence to an approximate solution guaranteed to be bounded by a combination of $n - 2F$ objectives of regular agents. Further works will aim to consider less stringent topological constraints on the communication graph.

References

[1] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[2] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[3] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2015.

[4] M. Kaheni, E. Usai, and M. Franceschelli, "A distributed optimization and control framework for a network of constraint coupled residential besss," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 2021, pp. 2202–2207.

[5] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1063–1076, 2019.

[6] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[7] N. Ravi, A. Scaglione, and A. Nedić, "A case of distributed optimization in adversarial environment," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5252–5256.

[8] L. Su and N. H. Vaidya, "Byzantine-resilient multiagent optimization," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2227–2233, 2021.

[9] K. Kuwarancharoen, L. Xin, and S. Sundaram, "Byzantine-resilient distributed optimization of multi-dimensional functions," in *2020 American Control Conference (ACC)*, 2020, pp. 4399–4404.

[10] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 90–104, 2012.

[11] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 766–781, 2013.

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

[13] N. Gupta and N. H. Vaidya, "Fault-tolerance in distributed optimization: The case of redundancy," in *Proceedings of the 39th Symposium on Principles of Distributed Computing*, 2020, pp. 365–374.

[14] N. Gupta, T. T. Doan, and N. H. Vaidya, "Byzantine fault-tolerance in decentralized optimization under 2f-redundancy," in *2021 American Control Conference (ACC)*, 2021, pp. 3632–3637.

[15] J. So, B. Güler, and A. S. Avestimehr, "Byzantine-resilient secure federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2168–2181, 2021.

[16] M. Cao, A. S. Morse, and B. D. O. Anderson, "Reaching a consensus in a dynamically changing environment: A graphical approach," *SIAM Journal on Control and Optimization*, vol. 47, no. 2, pp. 575–600, 2008.

[17] N. H. Vaidya, "Security and privacy for distributed optimization & distributed machine learning," in *PODC'21: Proceedings of the ACM Symposium on Principles of Distributed Computing*, 2021, p. 573.